Transformation of Domain-specific Models as Foundation for Context-Awareness in Complex Systems

Michael Cebulla

Technische Universität Berlin, Fakultät für Elektrotechnik und Informatik,
Institut für Softwaretechnik und theoretische Informatik,
e-mail:mce@cs.tu-berlin.de

Abstract. In this paper we propose a modeling framework for the integration of domain-specific knowledge into complex systems. This framework especially supports the robustness of behavior in autonomic systems. In order to provide meaningful and adequate behavior in the presence of dynamic (or adverse) environments systems must possess the ability to *assess* their current state on the basis of *knowledge* about themselves and their environments. We consider a system's state as represented by an array of domain-specific models (each describing a specific *aspect* of the system) and support the reasoning about behavior using transformation rules (based on multiset-rewriting). In order to get even closer to the semantics of domain-specific terminologies we rely on fuzzy description logics. Thus we are able to support robust automatic reasoning about behavior w.r.t. to different aspects of environments.

1 Introduction

Our approach relies on the conviction that the behavior of complex systems in dynamic environments has to rely on an *understanding* of their current situation. This can only be achieved when we enhance the systems capabilities to represent and reason about situations using abstract concepts which have a richer semantic content than concepts from traditional programming languages. In our approach we claim that the systems have to be able to represent the actual situation as domain-specific model thus using similar concepts, abstractions and reasoning mechanisms as human domain-experts. Thus, complex applications which are deeply embedded into medical processes (e.g. supervisioning a patient during a medical operation) have to be able to *interpret* the data in an adequate way as well as sensor networks have to *know* the subject of an observation in order to react to requests from the environment.

By the integration of domain-specific models into the architecture of complex systems we support a bundle of important requirements. One major requirement for this kind of systems is related toward their capability to automatically *adapt* to changing and unexpected environments. Thus they have to provide meaningful functionality also in situations which could not be *anticipated* by their developers. Especially the unavailability of expected services in pervasive environments may cause the necessity to search for alternative services or to invoke a reasonable fallback behavior. Consequently these systems belong to a new type of *systems* which have to be *context aware* and to possess the ability to *autonomously* control their behavior according to environmental conditions. This bundle of features is currently discussed under the topic *autonomous computing* [7]. In this presentation we claim that the automatic assessment of systems' behavior can well be supported by integrated reasoning about domain-specific models.

We claim that the role of formal methods and modeling techniques has considerably changed in the light of this evolution in the field of systems engineering. Traditionally formal methods were used to predict the systems' behavior during the design phase in order to support claims concerning their behavioral properties. In the future however modeling techniques will support the systems' ability regarding the assessment of their own behavior. For this sake systems have to internally represent their environment, to anticipate future developments and to initiate compensatory measures. As we claim the use of domain-specific models is a viable approach with respect to the increasing of situational understanding.

In this paper we concentrate on the knowledge-based assessment of the systems' situational behavior. We claim that this capability of self-evaluation is a precondition for advanced features of autonomous behavior. We propose to use description logics [1] as a light-weight formalism for domain-specific modeling and the support of automated reasoning. From our point of view such a light-weight formalism has to meet three requirements which may be unfamiliar from a traditional viewpoint.

Intelligibility. Formal specifications have to support domain specific high level concepts which enable an adequate view on the systems environment. We claim that techniques from knowledge representations and ontological reasoning can be employed in order to incorporate relevant domain knowledge in the systems. Thus, ontological modeling provides an instrument for a *seamless integration* of domain knowledge into the systems' architecture.

Uncertainty and Incompleteness. Adverse environments are characterized by unexpected changes *(uncertainty)* and by a high complexity which makes an exact description impossible or inefficient. Incomplete specifications make it possible to handle this situational vagueness in a robust way. As we will see the issues of vagueness and uncertainty are treated by the introduction of fuzzy logics [8] and modal logics [19] into terminological reasoning.

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. For this sake we heavily rely in our approach on the notion of *model transformation.* As we will see later we chose multiset rewriting as a foundation of transformation.

In this paper we present an approach for the integration of domain-specific models into complex systems which satisfies these requirements. We claim that the resulting knowledge-enabled systems show enhanced features in terms of context-awareness and adaptive behavior.

First we give an outline of the general architecture (cf. Section 2) which we are using for the integration of domain models. Then we briefly discuss the basic concepts of fuzzy description logics (cf. Section 3) which can be considered as the meta-model in our framework. Finally we describe how we can employ model transformation (based on multiset rewriting) in order to reason about complex behavior in dynamic environments (cf. Section 4).

2 General Architecture

Multi-Agent Systems. We base our approach on knowledge-based modeling of multi-agent systems [5]. It is well-known that multi-agent systems support behavioral adaptivity and local decision making and thus enable meaningful behavior in highly distributed systems. The resulting problem of *coordination* of local behaviors however can be approached in terms of knowledge-based modeling. In order to so we provide an architecture which enhances the availability of knowledge for the agents. As we already argued, we use domain-specific models (i.e. ontologies) for the description of the relevant modeling aspects. From an architectural perspective the relevant information is gathered in a *blackboard*-component [14] (or *tuple space*[4]) (cf. Figure 1). One characteristic feature of our approach consists in the distinction of multiple *knowledge aspects* (cf. [11] for a similar approach) represented by different domain-specific models which rely on the same metamodel. Thus knowledge about complex systems is represented as an array of related sub-models (represented by different knowledge bases) which we call *aspects.*

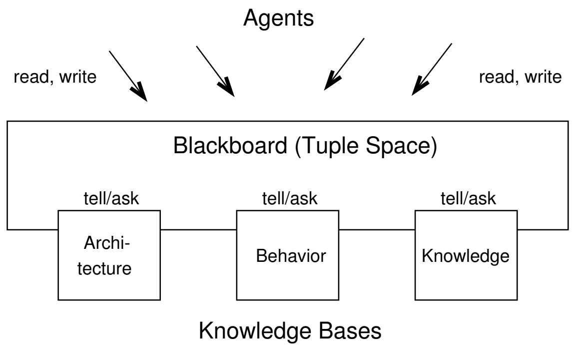


Fig. 1. Global Architecture

Aspect-related Domain-Models. Using description logics as metamodel we define terminologies as domain-specific models related to specific knowledge aspects. The separation between different aspects allows us to keep the metamodel simple and to reuse the same procedures for automatic reasoning (based on fuzzy subsumption). This kind of reuse heavily relies on semantic correspondences between description logics, dynamic logics and modal logics (first described by [13]). In the architecture the domain-specific models are represented as knowledge bases in our architecture which can be modified from outside during runtime using *tell-*, *ask-* and *remove-*routines which are provided by KB-interfaces [10].

This architecture directly supports automated domain-specific analysis and consistency checks. Since we organize domain-specific knowledge in knowledge bases we can apply reasoning services which are supplied by description logic [1]. Especially the reasoning about (fuzzy) subsumption supports the assessment of the current state w.r.t. to global requirements. Note that the use of description logics directly supports reasoning about incomplete specifications (via the open world assumption) and the integration of implicit information (via reasoning about subsumption).

Fuzziness. In general we use fuzzy extensions of description logics in order to provide an adequate representation of vague knowledge [16]. This vagueness enhances the effectivity of knowledge processing as well as the robustness of systems' behavior. Moreover in many cases a system's actual state has to be evaluated w.r.t. conflicting requirements. This means that it is impossible in many situations to satisfy all requirements completely at the same time. This type of problems which is known as fuzzy optimization problems [20] can be modeled using fuzzy description logics. Especially the automatic evaluation of a system's state can be supported by a fuzzy comparison of the actual state with the desired state. For an example cf. Section 3.

Applying the Chemical Metaphor. Following [3] we use the metaphor of a *chemical solution* for the domain-specific representation of a situation. A solution contains *molecules* which represent *terms* which are taken from the aspect-related domain-models (terminologies). For example they may represent knowledge about systemic agents, their behavior or their expectations. When these molecules meet certain criteria they can react according to *reaction rules.* Since in our model the terms are embedded in multisets the semantics of reaction rules consists in *multiset rewriting.* 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.

Systems Behavior: Transformation of Models. We use the rule-based transformation of domain-specific models in order to describe complex systems behavior. The behavior of complex systems depends on many influences which determine the characteristics of the successor state in a given situation. In our approach we conceive these influences as *aspects* and manage the knowledge related to them in different domain specific models. This approach enables us to describe global behavior as coordination process between different knowledge bases. For this sake we define a construct which we call *task* which is considered as a self-contained and meaningful entity of global systems behavior. Tasks are represented by transformation rules.

Definition 1 (Task (simplified)). *We define a task as a tuple* $T = \langle precon, post \rangle$, *where precon is a set of preconditions and post a set of post-conditions.*

From a technical viewpoint a task can be described as transformation of domain-specific models. Consequently the conditions in set *precon* have to hold in order to enable the application of a task while the conditions from set *post* are introduced into the knowledge bases in order to represent the state after the accomplishment of the task. Our concept of task is very similar to the planning operators in STRIPS [12] or to *service descriptions* [2].

Reasoning about Complex Behavior. There are two basic styles of reasoning about complex behavior which are supported by this kind of modeling.

Executability. An action is *executable,* if its preconditions hold (i.e. if they are satisfied by the interpretation of the knowledge bases $\mathcal{I} \models \mathsf{pre}$).

Projection. An action leads to the desired consequences (described by its post-conditions) if its post-conditions hold in the transformed interpretation $\mathcal{I}'$.

Precon	collapse-observed	Y	Y
	expertise	low	high
Post	do-alarm	Y	Y
	pulse-control	low	high
	stable-blood-circulation	Y	Y

Fig. 2. Task *Reanimation*

In Figure 2 we give an example for a simplified task specification (from the medical domain) using a domain-specific description. A small subset for the international guidelines for cardiopulmonary reanimation [18] is described using a tabular notation. We use a simple tabular notation (known as AND-OR-table) [9] in order to support an easy specification of systems behavior by domain experts. In the upper part we describe the situation that has to be given in order to make the application of the rule possible while in the lower part the resulting situation is described. Note that the semantic counterpart to situations are CHAM-style solutions which contain domain-specific terms as *molecules.*

In our example the task's goal consists in the achievement of a stable blood circulation. According to the guidelines we distinguish two situations: in the first the active agent has no professional training while in the second he is trained for handling such occurrences. For the sake of example in the first case the patient's pulse should not be tested (because of the limited exactness of this test). Note that we use the fuzzy attribute *expertise* to reason about the capabilities of the active agent.

Note that we supply a tabular notation which is suitable for the needs of domain-experts while at the same time we have a well-defined semantics for rule application which relies on multiset transformation. Consequently such rules can be applied in the course of automatic reasoning. They thus represented an constitutive part of knowledge which is available in the system.

3 Fuzzy Description Logics

In order to keep this presentation self-contained we give a brief review of fuzzy description logics in this section.

Following [15] we describe a fuzzy extension of description logics. We introduce semantic uncertainty by introducing multi-valued semantics into description logics. Consequently we have to introduce fuzzy sets instead of the crisp sets used in the traditional semantics (cf. [1]). 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 now 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* $C^{\mathcal{I}} : \Delta^{\mathcal{I}} \rightarrow [0, 1]$*;*

3. a role R into a membership function $R^{\mathcal{I}} : \Delta^{\mathcal{I}} \times \Delta^{\mathcal{I}} \rightarrow [0, 1]$.

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 $\mathcal{I}$ [15].

The interpretation function $\cdot^{\mathcal{I}}$ has to satisfy the following equations:

$$\begin{aligned}
\top^{\mathcal{I}}(d) &= 1 \\
\bot^{\mathcal{I}}(d) &= 0 \\
(C \sqcap D)^{\mathcal{I}}(d) &= min(C^{\mathcal{I}}(d), D^{\mathcal{I}}(d)) \\
(C \sqcup D)^{\mathcal{I}}(d) &= max(C^{\mathcal{I}}(d), D^{\mathcal{I}}(d)) \\
(\neg C)^{\mathcal{I}}(d) &= 1 - C^{\mathcal{I}}(d) \\
(\exists R.C)^{\mathcal{I}}(d) &= sup_{d' \in \Delta^{\mathcal{I}}}\{min(R^{\mathcal{I}}(d.d'), C^{\mathcal{I}}(d')\} \\
(\forall R.C)^{\mathcal{I}}(d) &= inf_{d' \in \Delta^{\mathcal{I}}}\{max(1 - R^{\mathcal{I}}(d.d'), C^{\mathcal{I}}(d')\}
\end{aligned}$$

In this article we silently introduce fuzzy numeric restrictions as well as predicates on fuzzy concrete domains (which are very similar to linguistic variables and support the integration of linguistic hedges [8]). We also heavily rely on the concept of *fuzzy subsumption* which we introduce by example.

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. In the following example we are interested in the degree to which a patient's *o2-saturation* can be considered as *normal* during a medical operation.

$$\begin{aligned}
\mathsf{normal} &\doteq \mathsf{situation} \sqcap \exists\ \mathsf{o2\text{-}saturation.very(High)} \\
\mathsf{current\text{-}situation} &\doteq \mathsf{situation} \sqcap \exists\ \mathsf{o2\text{-}saturation.}{=_7}
\end{aligned}$$

On this background we can reason about the following statement:

$$KB \mid\!\approx \mathsf{current\text{-}situation} \sqsubseteq \mathsf{normal}$$

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

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

As as solution we obtain a support of .33.

4 Context-Aware Behavior

Briefly considering another example we can demonstrate what kind of contribution may be expected from our approach with respect to context-awareness in complex

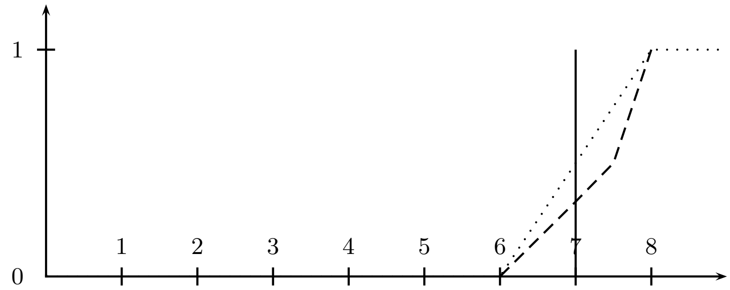
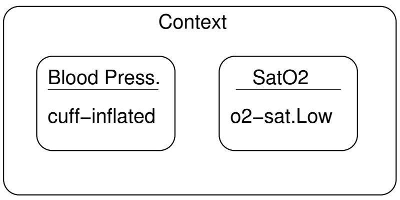


Fig. 3. Very High

systems. We show for example that a knowledge-based integration of medical sensor technology can be achieved on the basis of integrated domain-specific modeling. While there have been huge progresses concerning the development of sensors for the measurement of vital data the integration of these devices is still incomplete [17]. As a consequence a large number of nuisance alarms are bothering the medical staff and affect the system's safety. A typical case for an undetected interdependence between sensors resulting in false alarms is represented by the situation where an alarm related to oxygen saturation occurs when the blood pressure cuff is inflated.

On the background of our argument we require that in this type of situation the O_2-sensor should *know* whether the cuff is inflated. A meaningful behavior consists in recognizing situations where the issuing of alarm is inadequate.



Precon	02-saturation	low	low
	K cuff-inflated	N	Y
Post	do-alarm	Y	N

Fig. 4. Context-Aware Behavior of a Sensor Component

In Figure 4 we specify both variants of behavior. Note that the *knowledge* of the sensor component concerning its environment plays a specific role. When the component *knows* that the cuff is inflated the alarm is suppressed. Note that an agent's knowledge (denoted by operator **K**) is integrated as an aspect model in our framework.

5 Conclusions

In this paper we described a framework for the knowledge-based support of context-awareness in complex systems which relies on the integration of domain-models representing relevant information. Especially we support high-level descriptions of systems transformation on the basis of domain-specific modeling. In contrast to [6] we do not use architectural models but ontological knowledge representation.

By doing this we support a *seamless knowledge management,* i.e. enable the direct integration of relevant expert knowledge.

6 Acknowledgements

We are indebted to the anonymous reviewers for pointing out some weaknesses in an earlier version of this paper.

References

1. Franz Baader, Diego Calvanese, Deborah McGuinness, Daniele Nardi, and Peter Patel-Schneider, editors. *The Description Logic Handbook*. Cambridge University Press, Cambridge, U.K., 2003.
2. Franz Baader, Carsten Lutz, Maja Milicić, Ulrike Sattler, and Frank Wolter. Integrating description logics and action formalisms for reasoning about web services. Technical report, Technische Universität Dresden, Dresden, 2005.
3. Jean-Pierre Banâtre and Daniel Le Métayer. Programming by multiset transformation. *Communications of the ACM*, 36(1):98–111, 1993.
4. Nadia Busi, Paolo Ciancarini, Roberto Gorrieri, and Gianluigi Zavattaro. Coordination models: A guided tour. In Andrea Omicini, Franco Zambonelli, Matthias Klusch, and Robert Tolksdorf, editors, *Coordination of Internet Agents. Models, Technologies, and Applications.* Springer-Verlag, Berlin, 2001.
5. Ronald Fagin, Joseph Y. Halpern, Yoram Moses, and Moshe Y. Vardi. *Reasoning about Knowledge.* The MIT Press, Cambridge, Mass., 1996.
6. David Garlan and Bradley Schmerl. Model-based adaptation for self-healing systems. In J. Allen, R. Fikes, and E. Sandwall, editors, *Proc. of the First Workshop on Self-healing systems*, New York, 2003. ACM Press.
7. Jeffrey O. Kephart and David M. Chess. The vision of autonomic computing. *IEEE Computer*, 36(1):41–50, 2003.
8. George J. Klir and Bo Yuan. *Fuzzy Sets and Fuzzy Logic. Theory and Applications.* Prentice Hall, Upper Saddle River, N.J., 1995.
9. Nancy Leveson, Mats Heimdahl, Holly Hildreth, and John Reese. Requirements engineering for process-control systems. *IEEE Transactions on Software Engineering*, 20(9), 1994.
10. Hector J. Levesque. Foundations of a functional approach to knowledge representation. *Artificial Intelligence*, 23, 1984.
11. Peter Pepper, Michael Cebulla, Klaus Didrich, and Wolfgang Grieskamp. From program languages to software languages. *The Journal of Systems and Software*, 60, 2002.
12. R.E.Fikes and N.J.Nilsson. Strips: A new approach to the application oftheorem proving to problem solving. In J.Allen, J.Hendler, and A.Tate, editors, *Readings in Planning.* Kaufmann, San Mateo, Ca., 1990.
13. Klaus Schild. A correspondence theory for terminological logics: Preliminary report. Technical Report KIT-Report 91, Technische Universität Berlin, Franklinstr. 28/29, Berlin, Germany, 1991.
14. Mary Shaw and David Garlan. *Software Architecture. Perspectives on an emerging discipline.* Prentice Hall, Upper Saddle River, N.J., 1996.
15. Umberto Straccia. Reasoning with fuzzy description logic. *Journal of Artificial Intelligence Research*, 14:137–166, 2001.
16. Umberto Straccia. Towards a fuzzy description logic for the semantic web. Technical report, ISTI-CNR, Pisa, 2004.
17. Richard C. Watt, Eugene S. Maslana, and Kenneth C. Mylrea. Alarms and anesthesia. challenges in the design of intelligent systems for patient monitoring. *IEEE Engineering in Medicine and Biology*, 12(4):34 – 41, 1993.
18. V. Wenzel, W.G. Voelckel, A.C. Krismer, V.D. Mayr, H.-U. Strohmenger, M.A. Baubin, H.Wagner-Berger, A. Stallinger, and K.H. Lindner. Die neuen internationalen richtlinien zur kardiopulmonalen reanimation. *Der Anaesthesist*, 50:342–357, 2001.
19. Michael John Wooldridge. *The Logical Modelling of Computational Multi-Agent Systems.* PhD thesis, Department of Computation. University of Manchester, 1992.
20. Hans-Jürgen Zimmermann. *Operations Research. Methoden und Modelle.* Vieweg, Braunschweig, 1987.