Developing Situation-Aware Applications for Disaster Management with a Distributed Rule-Based Platform

João Moreira[1], Luís Ferreira Pires[1], Marten van Sinderen[1], Patrícia Dockhorn Costa[2]

[1] Services, Cyber-security and Safety group (SCS), University of Twente, Netherlands.
{j.luizrebelomoreira,l.ferreirapires,m.j.vansinderen}@utwente.nl

[2] Federal University of Espírito Santo (UFES), Brazil. pdcosta@inf.ufes.br

Abstract. In order to enhance interoperability and productivity in the development of situation-aware applications for disaster management, proper mechanisms and guidelines are required. They must address the lack of semantics in modelling emergency situations. In addition, the ever-changing and unpredictable nature of disaster scenarios present challenges for information processing and collaboration. This paper proposes a framework that combines the following elements: (i) a foundational ontology for temporal conceptualization; (ii) well-founded specifications of structural and behavioral models; (iii) a CEP engine based on a distributed rule-based platform for situation management; (iv) a model-driven approach. We illustrate the operation of the framework with a scenario for monitoring tuberculosis epidemy.

1 Introduction

Natural and man-made disasters are characterized by widespread destruction and distress, typically leading to material loss and human causalities. The disaster management process addresses an urgent social need and is being considered by a number of R&D projects, e.g. within FP7 and H2020 EU calls [10]. Situation-aware (SA) applications try to detect situations and react to them, being particularly useful to support disaster management. Examples are applications for meteorological forecasting, traffic control, victims' dead reckoning and power plants operation. The complexity and dynamic nature of emergency situations present a series of challenges, involving parties' interaction, including people (first responder groups and victims), environments, risks, hazards and multiple events. Those relations rely on complex rules, which are difficult to characterize during the specification of SA applications. Moreover, operational decisions must efficiently manage human and computational resources in short time, making the design of real-time decision making a big challenge. The information collaboration among SA applications has great importance and needs to be addressed. Furthermore, the unpredictability is inherent to disasters and considering all eventualities and possible situations during design time is unfeasible.

Our goal is to deal with the aforementioned issues by offering the necessary guidelines and components as a framework. The approach is divided in three interrelated parts guided by a model-driven approach: (i) a conceptual, to support the definition of real world constructs for modelling languages; (ii) a specification, to use the appropriate modelling languages; (iii) a implementation, to realize the specification. At the conceptual level, temporal and structural aspects are addressed by extending a foundational ontology. The specification level considers graphical modeling languages for context, situation and reaction. At the implementation level, the interoperability of SA applications is handled by a distributed environment. Complex event processing (CEP) and predictive analysis techniques are used to enhance situation management and support the detection of unforeseen situations. We illustrate our framework with a scenario in which a tuberculosis epidemy needs to be detected.

2 Development of Situation-aware Applications

Our framework is depicted in Figure 1 and is aligned with Endsley's situation awareness theory in dynamic decision making [1]. The starting point is the conceptual part based on the Unified Foundational Ontology (UFO) [4], which is a top-level ontology and provides a precise definition of temporal aspects, e.g. situations, events, temporal relations and participations [2]. An initial adaptation of UFO to consider the notion of situation types and context situations was introduced in [3].

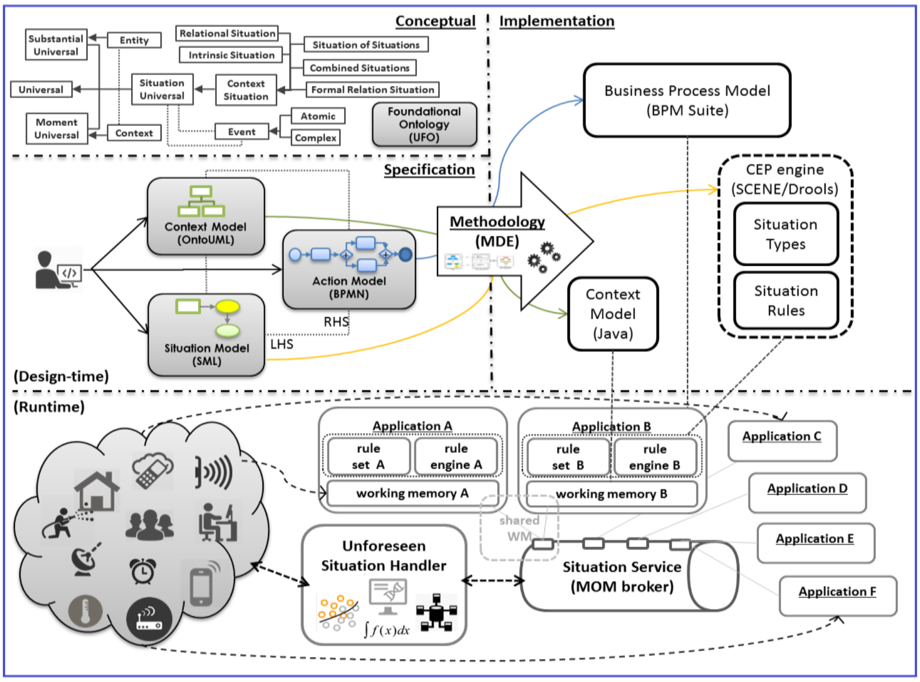

Fig. 1. Framework for SA application development.

At the specification part we adopted OntoUML [4], which is an ontological language having syntax rules to enforce the ontological assumptions from UFO. It is used for representing core and domain ontologies through a reference model. OntoUML is evolving since its introduction in [12] and a number of works address the relations to standards such as OWL, OCL, SWRL and UML [13], which are parts of the Ontology Definition Metamodel [14], the current OMG standard for ontological MDE. However, OntoUML lacks to support the definition of specific situation patterns. The Situation Modeling Language (SML) [6] can be used for this purpose, since it supports the design of SA applications and complements the specification of situations in OntoUML [7]. For modelling reactive behaviors we propose the use of BPMN. Based on those specification components, a MDE approach is applied:

- A cyclical semantic enhancement process is performed to increase the context and situation models quality. It involves the SA application designer and the domain experts. It is a formal lightweight verification and validation process to assess the model, taking advantage of the ontological model-driven framework introduced in [4]. This framework provides a set of capabilities through the Menthor modeling tool [11], such as the detection of anti-patterns and visual validation supported by Alloy logic language [7]. In addition, this tool considers the automatic transformations from OntoUML to standards (e.g. UML, OWL and SWRL).
- A (semi) automatic code generation process taking as input the specifications (in OntoUML, SML and BPMN). The context model is transformed in Java classes. The situation model is mapped to SCENE [8], a CEP engine that extends Drools Fusion for situation management. For the realization of the behavioral models, a BPM suite is adopted, where business process automation can be performed.

An infrastructure for distributed situation management based on SCENE provides a situation notification service bus (SiNoS) [9]. This enables the interoperation of SA applications and the information collaboration among them. At last, during runtime, a module to infer situations not specified (unforeseen situations) gathers information from SiNoS and other data sources (e.g. sensors and linked data), and tries to predict situations not modeled during design-time, suggesting them to the user.

3 Applying the framework for tuberculosis epidemics control

Suppose that a SA application is required to detect tuberculosis (TB) epidemics in a certain *geographical region*, e.g. a *city*. The system can use information from *hospitals* within the *city*, which provide relevant real-time data about their *patient treatments*, such as their body *temperature* and exams' results. At first, the context and situation models must be specified. The former is provided by the extension of the OntoEmerge core ontology [5], shown in Figure 2a. Notice that the concepts of the ontology is stereotyped with UFO categories, e.g. a *treatment* as a **relator** between the *patient* (a *person* **role**) and the *hospital* (an *installation* **subkind**). As described in [8], the detection of a *person* having TB disease depends on a combination of TB symptom situations of this person during a TB infection, illustrated in Figure 2b. A *person* with TB symptom is characterized by a past situation of high fever (*temperature* higher than 38.5° C) followed by a common fever situation (*temperature* higher than 37° C) after one hour and before two days. The TB infection situation occurs when the patient has a positive result for the Interferon-Gamma Release Assays (*IGRA*) exam.

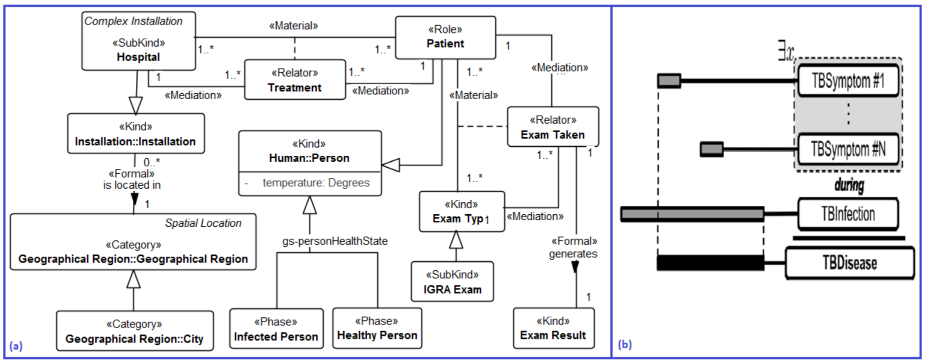

Fig. 2. (a) Healthcare context model. (b) Timeline for TB disease situation detection [8].

Once a TB infection is detected, the *healthy* phase of a *person* changes to *infected*. As described in [7], a possible contagion situation of the infected *patient* (e.g. John) being treated in a *hospital*, overlapping with another *person* (e.g. Maria) being treated in the same *hospital* and becoming *infected*. Figure 3a illustrates the SML model that describes this situation, considering its composition by other situations. A TB epidemy situation in a city can be characterized by the occurrence of a set of TB possible contagions among nearby *hospitals*, where the number of contagions are greater than the expected by the National Health Institute. Once the tuberculosis epidemy situation is described with SML, the reaction should be defined. A simple reaction is to warn the involved parties, i.e. the *hospitals* involved in the epidemic, as well as the civil defense and the National Health Institute, illustrated in Figure 3b. A more complex reaction can be designed by taking advantage of the expressiveness in BPMN. The assessment of the possible contagion situation is fully described in [7].

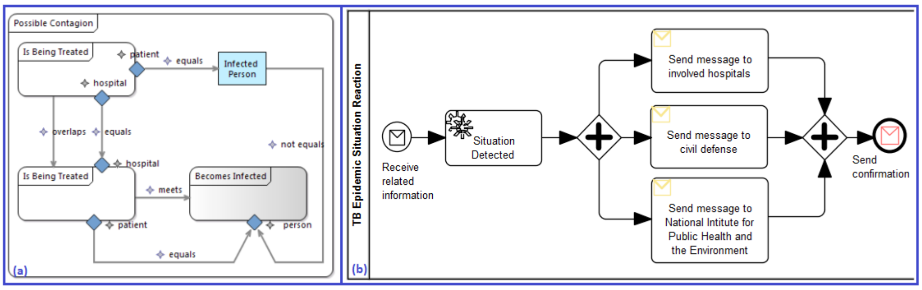

Fig. 3. (a) Possible contagion situation in SML [7]. (b) Reaction model in BPMN.

For the SA application implementation, Java classes are generated from the concepts in the context model, such as *patient* and its attributes, as well as its *treatment* relation to *hospital*. Situation types (e.g. TB disease) are also represented as Java classes within SCENE environment, and situation rules are mapped to the adapted Drools domain specific language (DRL) for situations management, as illustrated in Figure 4. The warning reaction model is attached to the TB epidemy situation. In the BPM suite, the reaction is designed and each task is an activity in which an SMS message is sent.

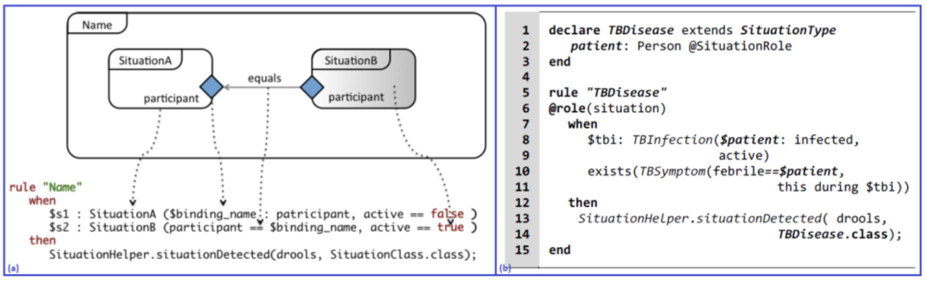

Fig. 4. (a) Mapping rule from SML to DRL. (b) TB disease situation as a SCENE rule.

4 Conclusion

This paper proposes a framework for the development of SA applications to cope with disaster management. We handle the specification and implementation phases at design-time by a model-driven approach. The former is grounded in UFO and considers structural and behavioral modelling through the graphical languages: OntoUML, SML and BPMN. The later includes SCENE – a CEP engine based on a rule-based platform – and SiNoS (Situation Notification Service) – a publish-subscribe distributed environment. A BPM suite complements the behavioral realization and a predictive analysis module infers new situations of interest. The framework is exemplified through a tuberculosis epidemy control scenario.

The use of the framework points to a productivity enhancement in developing SA applications for disaster management. Furthermore, by improving the semantics in modeling and adopting a model-driven approach, the implementation becomes less error prone. Although we exemplified the framework for tuberculosis control, it can be easily applied in different domains under the disaster management field, based on OntoEmerge core ontology.

At the current stage some manual effort is required for the code generation process from the context (OntoUML) and situation (SML) models to the CEP engine (SCENE), since the mapping rules from [6] needs to be revised and updated. In addition, the translation of these modelling languages to the current reaction rule language standard, the RuleML [15], represents an open topic. Current and future work includes: (i) the integration of the specification components; (ii) the adoption of quantifiers in SML; (iii) the comparison of SML with other rule languages, such as BPMN rules and visual EPL; (iv) the automation of the model-driven process; (v) the formalization of the UFO extension; (vi) the comparison of SiNoS with other approaches, such as ESPER and Apache Spark; (vii) the extension of OntoEmerge for other disaster situations; (viii) the application of the framework in a real case; (ix) mapping transformations from SML to RuleML.

Acknowledgements

Our acknowledgments to CAPES PhD scholarship (process BEX 1046/14-4) and Services, Cybersecurity and Safety (SCS) group, University of Twente.

References

1. M.R. Endsley. (1995). Toward a theory of situation awareness in dynamic systems. Human Factors and Ergonomics Society, vol. 37, no. 1.
2. Guizzardi, G., Wagner, G., Falbo, R. D. A., Guizzardi, R., & Almeida, J. P. (2013). Towards ontological foundations for the conceptual modeling of events. 32nd International Conference on Conceptual Modeling (ER), 8217 LNCS, 327–341.
3. Costa, P.D. (2007). Architectural Support for Context-Aware Applications: From Context Models to Services Platforms. Ph.D. Thesis, University of Twente.
4. Guizzardi, G. (2014). Ontological Patterns, Anti-Patterns and Pattern Languages for Next-Generation Conceptual Modeling. Keynote on 33rd International on Conceptual Modeling Conference (ER).
5. Ferreira, M. I., Moreira, J. L. R., Campos, M. L. M., Sales, T. P., Braga, B. F. B., Cordeiro, K. F., Borges, M. (2015). OntoEmergePlan: variability of emergency plans supported by a domain ontology. 12th International Conference on Information Systems for Crisis Response and Management (ISCRAM).
6. Costa, P. D., Mielke, I. T., Pereira, I., Almeida, J. P. (2012). A Model-Driven Approach to Situations: Situation Modeling and Rule-Based Situation Detection. 16th International Enterprise Distributed Object Computing Conference (EDOC).
7. Sobral, V. M., Almeida, J. P., Costa, P. D. (2015) Assessing Situation Models with a Lightweight Formal Method. 5th International Multi-Disciplinary Conference on Cognitive Methods in Situation Awareness and Decision Support (CogSIMA).
8. Pereira, I., Costa, P. D., Almeida, J. P. (2013). A Rule Based Platform for Situation Management. 3rd International Multi-Disciplinary Conference on Cognitive Methods in Situation Awareness and Decision Support (CogSIMA). Best paper.
9. Raymundo, C. R., Costa, P. D., Almeida, J. P., Pereira, I. (2014). An Infrastructure for Distributed Rule-Based Situation Management. 4th International Multi-Disciplinary Conference on Cognitive Methods in Situation Awareness and Decision Support (CogSIMA).
10. http://ec.europa.eu/programmes/horizon2020. Accessed on June, 2015.
11. http://www.menthor.net. Accessed on June, 2015.
12. Guizzardi, G. (2005). Ontological foundations for structural conceptual models. Ph.D. Thesis, University of Twente.
13. Carraretto, R.; Almeida, J. P. A. (2012). Separating Ontological and Information Modeling Concerns Towards a Two-Level Model-Driven Approach. International Workshop on Models and Model-Driven Methods for Service Engineering (3M4SE).
14. http://www.omg.org/spec/ODM/1.1/. Accessed on June, 2015.
15. Paschke, A., Boley, H., Zhao, Z., Teymourian, K., & Athan, T. (2012). Reaction RuleML 1.0: Standardized Semantic Reaction Rules. 6th International Symposium on Rules: Research Based and Industry Focused (RuleML).