From Situation Modelling to a Distributed Rule-Based Platform for Situation Awareness: An Ontological Framework for Disaster Management Applications

João Luiz Rebelo Moreira

Services, Cyber-security and Safety group (SCS), University of Twente, Netherlands
j.luizrebelomoreira@utwente.nl

Abstract. Situation-aware (SA) applications are particularly useful for disaster management. The complex nature of emergency scenarios presents challenges to the development of collaborative and distributed SA solutions. These challenges concern the whole lifecycle, from specification to implementation phases, such as how to model the reaction behavior of a detected situation and how to provide an interoperable situation notification service. In addition, treating unforeseen situations, i.e. situations not modeled at design-time, is an open topic. In this PhD research we intend to deal with these issues by proposing a framework for SA application development, which is applicable in different emergency scenarios and covers the conceptual, specification and implementation aspects, through an ontological model-driven methodology. The conceptual requirements can be addressed by adopting an appropriate foundational ontology. The specification aspects include the design of the context, the situation types of interest and the decision to be performed by the application when a situation is detected. For supporting the implementation of the SA application, we intend to improve the interoperability and performance of an existing rule-based situation notification platform, which is based on service oriented architecture and complex event processing. A separate module takes advantage of predictive analysis techniques to detect unforeseen situations. We plan to evaluate our framework with SA application prototypes and proofs-of-concept.

Keywords: Situation awareness, situation modeling, ontology, complex events processing, emergency management, disaster, distributed rule-based systems.

1 Introduction

The main requirement of situation-aware (SA) applications is the ability to perceive the environmental elements (the context): space, time and other properties of objects, such as a person's temperature or a river's throughput. These characteristics correspond to a situation's composition, which is a clearly recognized pattern objects having properties and standing in relations to each other [1]. SA applications are applied in a variety of domains, and they are particularly useful for emergency and disaster management. Emergency situations are complex due to their dynamic nature, which involve a series of events and a set of actors, such as victims, rescue teams and

adfa, p. 1, 2011.

officials. A difficulty in an emergency scenario is to detect situations and be able to quickly react to them. SA solutions build on several disciplines, such as conceptual modeling, human-computer inter-action, information fusion, data mining and high performance computing [4, 13, 14, 15, 20, 25]. Some SA tools rely on ontological approaches for specification and implementation [2]. Regarding the emergency domain, enterprises are applying complex event processing (CEP) and data stream management solutions, for processing high rates of information flow and extending their traditional databases [13]. These approaches aim at improving real-time decision making. Moreover, they provide mechanisms for predictive analysis to recognize and reason about situations that could not be foreseen during the SA application's design-time [27, 29].

The modeling of emergency situations and their actions requires further exploration [4, 5, 8, 9, 24]. When integrating business processes of heterogeneous and distributed SA applications we have to face interoperability issues [14, 19, 20, 22]. Processing and managing interrelated data as events, that characterizes situations, bring performance and scalability problems [21, 22, 27]. In addition, the detection of unforeseen situations from diverse information sources is a subject that needs to be investigated in more detail [13, 27]. These problems are interrelated and SA applications' development requires a holistic approach to overcome them.

This PhD research aims to deal with the aforementioned issues through a well-founded framework. Our approach is divided into three parts and a methodology (Figure 1). In the conceptual part, situations and their relations to events and structural aspects (the context) are addressed. In the specification part, modeling languages address: (i) context; (ii) conditional patterns that define situation compositions brought by events and; (iii) actions to be fired when the situation occurs. In the implementation part, interoperability of SA applications is treated by a distributed environment based on the Service-Oriented Architecture (SOA) paradigm. Complex Event Processing (CEP) engines are used to enhance the situation's management of SA tools and support the detection of unforeseen situations. A methodology for the lifecycle of SA application development serves as guidelines.

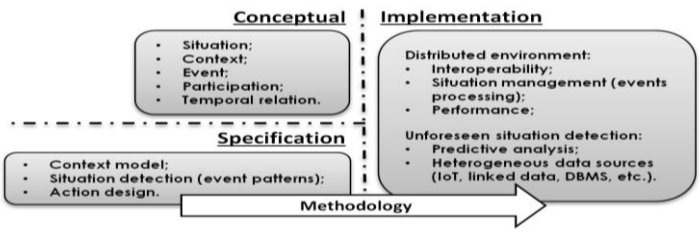

Fig. 1. Aspects in developing situation-aware applications

This paper is organized as: Section 2 presents the problem analysis related to the PhD research goals. Section 3 describes some existing solutions and their appropriateness to solve the problems. Section 4 introduces our approach proposal. Section 5 describes how we intend to validate the framework. Section 6 lists the expected contributions, identifies some possible limitations and presents the work's current status.

2 Problems and Research Goals

The dynamic nature of emergency situations presents a series of challenges because of its complexity [3]. Emergency situations involve environments, risks, hazards and multiple events occurrences that all unfold in time and space. These aspects are related to parties that have to interact, including people (first responder groups, victims), governmental officials, non-governmental organizations and technological resources. To handle the emergency response phase in emergency management, decision makers must perform operational decisions to efficiently manage human and computational resources. In the last years a number of SA applications have been developed to deal with the ever-changing nature of situations in emergency scenarios and real-time decision making, while managing emergency situations, which are dynamically shaped by (combinations of) events. For example, SA tools that support logistics for first responders of disasters have particular characteristics that undergo constant changes, such as the calculation of escape routes of a city being affected by a tsunami, the available paths and how damaged they are. They must consider information from different sources, such as SMS messages from involved people, satellite images and sensors in the field. This leads to challenges regarding the characterization of situations at design time and the interoperability among SA solutions at runtime, in terms of collaboration and information exchange.

The existing mechanisms for SA (e.g. context-aware applications), usually consider situations that have been foreseen and modeled at design time. Unforeseen situations are, therefore, not considered by the application implementation, but they need to be addressed. In disaster management, the unpredictability is inherent to the emergency context and can affect the emergency plans, being vital to the success of decision making. For instance, consider an SA application that supports a rescue mission by locating the doctor of the rescue team nearest to a victim that needs care. This application behavior takes place because this situation was thought and modeled a priori, during the application design time. However, if another victim is a doctor with adequate conditions to help on other victim, but she is not part of the rescue team, and she is closest to the victim, then she may be the best option for treating the victim. If this situation type has not been foreseen at design-time, then the SA application would not be able to suggest it at runtime.

Therefore, with respect to the problems mentioned, the goals of this PhD research can be summarized as:

- Modelling emergency situations and their response for developing SA applications;
- Supporting distributed SA applications for disaster management, considering characteristics such as interoperability, performance, scalability and reliability;
- Developing a mechanism to discover and react to unforeseen emergency situations based on large amounts of heterogeneous data;
- Adapting a development methodology for creating or modifying existing SA applications.

3 Background

Situation awareness (SA) is the ability to recognize a situation. The notion of situation has been studied for decades in Philosophy [1] and can be defined as limited parts of the perceived reality, i.e. a set of patterns (the state of affairs) of the observed world [15]. “A situation is a particular configuration of a part of reality which can be understood as a whole.” [11]. The concept of SA is used in several domains, e.g. aviation and maritime navigation (piloting), traffic control, power grid operation and military command. In Computer Science, SA approaches are typically used to support disaster management (DM) processes, which are based on prevention, mitigation and preparedness (before), response (during) and recovery (after). During the first phase, objectives are established; risks are assessed, prevented and mitigated; plans are made; teams and emergency equipment are prepared. The emergency response focuses on rescue, relief and salvage; immediate damage assessment and the protection of damaged heritage. During the recovery phase, the damage assessment is detailed; the restoration, repair and re-habitation are made. DM is an urgent societal need causing huge investments in R&D projects (e.g. H2020). SA solutions rely on multidisciplinary approaches (from conceptual modeling to cloud computing) and are implemented as different types of situation assessment tools, such as device-based (e.g. mobile), location and dead-reckoning and unmanned aerial vehicles (UAV); and decision support systems (DSS). SA solutions can be applied in all DM phases.

In the context of this work, the common high-level requirements for SA applications are (from [20]): (i) compatibility with existing DM tools for smooth integration, in terms of interoperability; (ii) providing a holistic overview of the situations and being able to synthesize available data; (iii) dynamicity in making data available; (iv) efficient organization and coordination of goods and personnel. We describe some solutions proposed in the literature to realize these requirements by each part, considering our goals.

3.1 Conceptual

Some approaches rely on SA ontologies as reference models [8, 25], which are independent of the specific domain. These ontologies are called core ontologies [15], or upper ontologies [2] ontology, and provides a precise definition of the structural knowledge in a field independently of any specific domain application [23]. From a core ontology it is possible to derive a set of ontology design patterns as modeling solutions for recurrent ontology development problems [7, 26]. A core ontology is based on a foundational ontology (e.g. DOLCE [23] or UFO [10]), which is a high-level system of categories grounded on philosophical logic, linguistics and cognitive psychology. A foundational ontology represents the most basic and general concepts and relations that make up the world, and it is known in the Conceptual Modeling community as a top-level ontology [10]. This ontology describes the principles of identity, rigidity, and dependency, using axioms in a first-order logic as a formal ontology. The boundaries between a foundational and a core ontology, as well as between a core and a domain ontology, are quite fuzzy and represent an open topic for research. DOLCE adopts the Descriptions and Situations (DnS) pattern, providing an ontological formalization of a contextual situation, i.e. events and objects observed in a concrete situation that satisfy a specific description.

UFO was conceived from DOLCE and other foundational ontologies. It brings a set of well-founded representations of *perdurants* (coined as UFO-B [11]), i.e. things that occur in time (e.g. event's mereology, participation and temporal relations). It is related to a full-set of *endurant* representations, i.e. structural aspects (e.g. kind, role, mediation and mereological relation types), coined as UFO-A [10]. Initially, in UFO, a **Situation** [4] was suggested as a specialization of the **Universal** meta-concept, disjoint and complete with **Substantial** and **Moment**, being composed by other **Universals**. A **Context Situation** was defined as a specialization of **Situation** that has one or more **Entity** and one or more **Context**. Five specializations of a **Context Situation** were proposed, namely: **Relational**, **Intrinsic**, **Formal Relation**, **Combined** and **Situation of Situation** [4]. In addition, the temporal aspects of situations were explored in [11], where begin and end time points of a **Situation** are obtained from the two possible causality relations with **Events**: **brings-about** and **triggers**. Moreover, a **Situation** can **activate** an object **Disposition**, which is manifested by an atomic event that was triggered by the **Situation**. However, these definitions have not been harmonized and axiomatized in UFO, forming a gap in the conceptualization of situations.

3.2 Specification

In conceptual modeling, an ontological language follows the representation of the concepts from a foundational ontology [10]. This type of language considers structural relations as primitive constructs, satisfying meaning postulates and giving semantic expressiveness to models. The specific design patterns for one ontological language can be derived from core ontologies representing a specific field [7]. The ontological language differs from epistemological and logical languages because it restricts the interpretation of the designed model. However, it increases the complexity of the ontology and tends to be inappropriate for automatic reasoning. Therefore, a methodology – as a theoretic and consistent ontological engineering process – is necessary. This methodology drives the development and mapping of the reference (conceptual) model, represented by an ontological language, to lightweight ontologies, represented by an epistemological language (e.g. OWL). This methodology is analogous to software engineering, where the conceptual model (reference ontology) is built in the analysis phase to establish common sense, whilst lightweight ontologies are built during construction phase to realize computational requirements. Ontological Model-Driven Engineering (MDE) is a field that aims at automating this methodology by realizing the transformations from the domain ontology to software constructs.

OntoUML [10] is an ontological language (UML extension) with syntax rules to enforce the ontological assumptions of UFO. It is used for representing a domain in a reference model [10], and is being applied in several domains, such as emergency plans generation [9]. In addition, it is supported by the Menthor tool [16], which addresses some issues in ontological MDE, e.g. model verification and validation through anti-patterns and Alloy analyzer [12], constraints for temporal rules in OCL, mappings for UML [28] and code generation for OWL and SWRL.

An MDE approach to SA applications was proposed in [5] where the Situation Modelling Language (SML) was introduced for: (i) defining situation types at design time and detecting instances of these situation types at runtime; (ii) defining a situation type as a set of constraints related to its participants' properties and relations; (iii) considering temporal parts of situations; and (iv) combining (compositing) situation types, allowing their reuse. SML uses OntoUML to represent the context model and to enhance the ontological MDE with the formal validation method for situation assessment [24]. A set of mappings to the Alloy logical language was added to the Menthor tool, providing a mechanism for SA application designers to improve their conceptual models through visual simulation. Nevertheless, the specification of the behavior of the SA application when a situation is detected, i.e. the actions that should be performed as reaction to a situation, remains relatively unexplored [4].

3.3 Implementation

Whilst SML plays the role of situation specification, the Drools rule-based platform [6] was chosen for the support the realization of situations' detection in [5]. The idea of this approach comes from a prior work [4]. To specify context-aware reactive rules, based on the event-control-action (ECA) pattern, it introduced the domain-specific language ECA-DL, which can be deployed at runtime due to the rule-based nature of the core engine. Rules are similar to *if-then* statements, where the *if* part is called the left hand side (LHS), defining a set of patterns that, when satisfied, fires the actions defined in the *then* part, i.e. the right hand side (RHS). Drools implements the RETE pattern matching algorithm, which efficiently detects the patterns in the LHS of a rule and stores them as facts in its working memory (WM). In this way, the detected situations can be remembered from the past pattern matching tests. Drools also provides a CEP engine (Fusion) to select the interesting events (and their relations) from an event set and infer new data. An event is a record that occurs in a point in time causing a change of state in the application domain. Fusion supports the main requirements of CEP platforms, e.g. (i) event detection, correlation, aggregation and composition; (ii) processing of events' streams; (iii) temporal constraints for temporal relations (Allen´s operators); and (iv) sliding windows of events. One can compare Drools Fusion to ESPER [27] (open source CEP platform), which uses the Event Processing Language (EPL) as the specification of complex events patterns. EPL resembles languages such as SQL, SPARQL and ECA-DL.

SCENE [21] is an extension of Drools Fusion that natively supports rule-based SA. It deals with the definition of situation compositions through Java classes, inheriting the situation type pre-defined class, and Allen's operators. It extends the temporal reasoning of Drools Fusion by recording currently (active) situations. This is fundamental for situation lifecycle management, i.e. the management of situation's state changes (activated and deactivated). Moreover, SCENE benefits from the truth maintenance system of Drools, which is responsible for ensuring logical integrity of facts in the WM. To facilitate the distribution of these capabilities to SA applications following the SOA paradigm, SCENE was extended to include a situation notification service (SiNoS) [22]. SiNoS realizes the requirements of: (i) providing situation type detection as a service; (ii) distributing services according to situation types; (iii) managing the lifecycle of distributed situations; (iv) decoupling situation providers from consumers; (v) sharing instances of the same situation type among multiple providers; (vi) computing in high performance in means of speed, efficiency, resource consumption, throughput and response time; (vii) mirroring the WM of providers and consumers that are implemented with SCENE; and (viii) providing platform independence for situation consumers that are not implemented with SCENE.

SiNoS supports those requirements by implementing common functionalities of message-oriented middleware (MOM) brokers, such as using publish/subscribe model to support event-driven processing and queuing messages through channels management. Despite the platform independence requirement (in terms of SCENE in subscribers), SiNoS underlying technology is based on Java RMI. This choice was made to achieve the WM mirroring requirement and to facilitate the analysis of the platform performance. A drawback is that providers and consumers have to be implemented in Java, limiting the interoperability of the platform. In addition, it caused a series of problems that are usually addressed by MOM brokers, with regards to: (i) reliability, since the ability to perform messaging management under stated conditions for a period of time can bring processing overhead and affect performance; and (ii) scalability, since the ability to deal with heavy overloads of request publications can bring resources management issues.

3.4 Related work

MOM brokers provide an abstraction layer for programming by trying to hide typical issues of low-level codification through: (i) multiple communication protocols; (ii) message serialization; (iii) abstraction of network physical attributes; (iv) transparent cooperation of heterogeneous systems in terms of their platforms (e.g. operating system, programming language); (v) automatic message buffering and delivery; (vi) load balancing and high scalability for optimal use of resources, dynamically routing and multiplexing when data volume increases (sharing message queues) [19]. Among the most popular communication protocols, SOAP is an open standard MOM specification based on XML that brings performance issues for extreme lightweight messaging transport, which is required for telemetry data exchange. Architectures such as REST and web sockets are being applied to deal with this issue, focusing on machine-to-machine connectivity, as in the Internet-of-Things (IoT) [14]. They support web services communication for business-to-business (B2B) integration. Usually, services are used and composed in BPM suites, which provide mechanisms to specify business processes with Business Process Management Notation (BPMN), for orchestration and B2B choreography. A BPM suite brings the same benefits as MDE: the application software is (semi) automatically built from process specifications.

An enterprise service bus (ESB) aims at enabling the implementation, deployment and management of SOA solutions by supporting a number of MOM engines [19]. In addition, ESB platforms usually address the dynamicity and reduce the cost of resource allocation in distributed environments through cloud computing techniques. This type of infrastructure becomes more necessary whenever large amounts of data that are produced and consumed by a number of actors, including SA applications, the IoT environment and linked datasets (*triplestores*), which are valuable sources of information. Data warehousing approaches for extracting, transforming and loading data are evolving to deal with the requirements of volume, variety and velocity (the 3V's of the "big data era"), termed as real-time ETL. This functionality has commonalities with the CEP feature of fast processing of data streaming for (near) real-time decision making. Furthermore, real-time ETL and CEP approaches try to support predictive analysis through data mining and machine learning mechanisms, by recognizing patterns and reasoning about them [29].

4 Proposed Approach

We structured our framework in terms of conceptual, specification and implementation (realization) parts and a methodology. The high-level requirements for each part are:

- **Conceptual**: conceptualizing *perdurants*, e.g. situation, event, participation;
- **Specification**: modelling situation types as conditional patterns, their associated structural context and the actions to be fired when the situation is detected;
- **Implementation**: processing and managing situations' lifecycles in a distributed environment to support SA applications;
- **Methodology**: engineering to create or adapt SA applications to our platform.

Figure 2 illustrates the general architecture of the framework, its components and their relations. To address the conceptual part of our approach we propose the adoption of UFO because it copes with perdurants. For specification, we propose the adoption of OntoUML for the structural representation of the context, because it is a mature ontological language and is supported by an appropriated modelling tool. For situation modeling, we purpose the use of SML for designing the conditional patterns of the behavior of the situation participants. SML also supports situation compositions with temporal relations among them, resembling complex events design. To represent the RHS of a decision, i.e. the proper actions to be performed when a situation occurs, we propose the use of BPMN as process specification language. BPMN brings a set of benefits, such as its high expressivity and its widespread adoption in BPM suites for SOA specification and realization.

To address the distribution environment for SA applications we intend to use the SiNoS general architecture as starting point. Then, to realize the requirement of platform independence, we intend to port SiNoS to more widely accepted interoperability standards (e.g. SOAP and REST), and support the integration of applications developed in different languages and technologies. A MOM broker, such as an ESB, can address this need by serving as a complete message backbone infrastructure, mediating service providers and consumers, alleviating interoperability issues based on the idea of configuring applications integration instead of coding. To allow transparent performance evaluation, the broker code must be available for debugging and testing. Thus, we will investigate the open source brokers that: (i) support Drools; (ii) support elasticity in the cloud; (iii) support lightweight publish/subscribe messaging where network bandwidth has high priority (e.g. sensors communication for machine-to-machine connectivity); (iv) are currently available and up-to-date; and (v) are widely accepted in the community. Examples are JBoss ESB, Mule, OpenESB, Petals ESB, MQTT, Apache ServiceMix, Apache Kafka and Apache Spark.

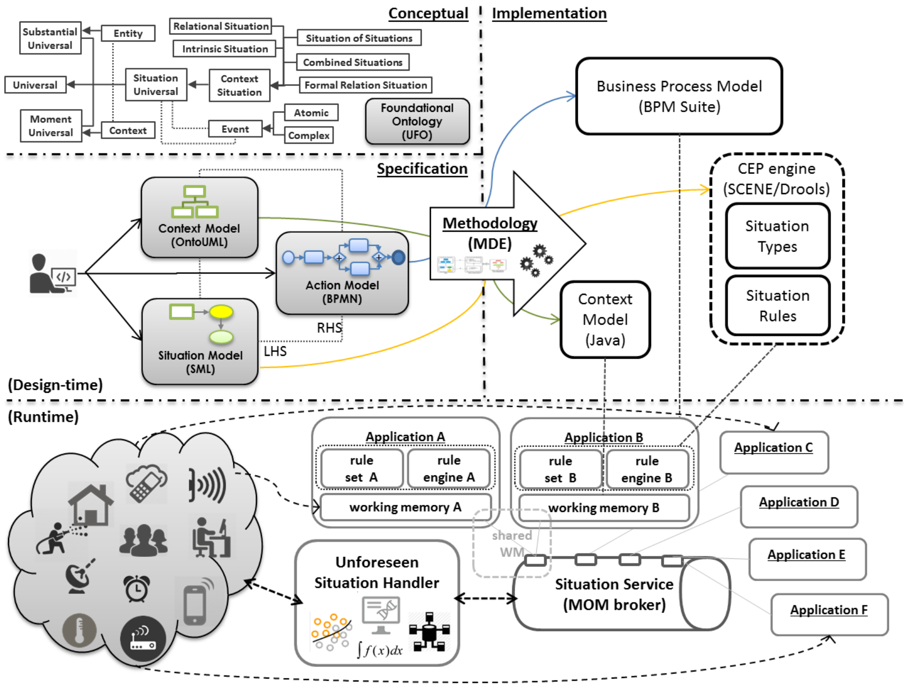

Fig. 2. Framework for the development of situation-aware applications

To deal with unforeseen situation detection, methods for predictive analysis will be considered. The idea is to conceive a module to learn patterns from different data sources and suggest them as unforeseen situations. To enhance the efficacy of these suggestions, measures should be used, such as the impact of the unforeseen situations in the specific domain of the application, considering the most common effects of their occurrences. We intend to design a historical situation data warehouse to correlate data originated from the detected situations in the platform as well as data from available relational databases, linked data datasets and the IoT environment. Then, these discovered situations must be somehow presented to the SA applications designers, so that they can choose if they want to take them into account at design time. In this way, designers can include situations already specified by other applications. Ideally, the unforeseen situations discovery mechanism could also try to infer the relevant situations and adapt the involved applications at runtime, characterizing them as self-adaptive systems. However, we need to analyze the effort needed to incorporate this and its impact in our research planning.

In the methodology, we intend to describe the processes to create and/or change existing SA applications so that they benefit from our platform (for both publishers and subscribers). These guidelines will consider ontological MDE to derive the implementation (semi) automatically from the specification.

5 Application and Evaluation

We intend to research and compare other approaches that cover SA application development for emergency and disaster management, arguing their advantages and disadvantages. For example, languages for representing situations (and their related events), such as the ECA-DL, BPMN rules, visual EPL and profiles of UML (e.g. event and state) will be compared with the ones we have chosen. For the evaluation of the implementation, other engines for event processing (e.g. ESPER) will be compared with our choices in terms of interoperability, scalability, performance and reliability. Moreover, when our platform evolves, we shall compare it with its former versions. For instance, the adoption of a MOM broker to realize the technological independence requirement will be compared with the first version of SiNoS by using equivalent experimentation. We will apply our framework to the development of new (examples or use cases) and/or the maintenance of existing SA applications for disaster management, which will play the role of publishers and subscribers on the platform.

As an entry point, we intend to use an emergency core ontology to underpin each SA application domain ontology. An ontology built in prior work [9] defines general properties of emergency events and, therefore, can be used as a basis for this core ontology. In addition, other ontologies should be investigated and may be integrated through ontology alignment. Interesting situations for the SA applications will be designed, as well as their respective actions (as business processes), which can even benefit from our platform in their service compositions. Then, ontological MDE will be applied for the implementation (or specific changes) of each application. Furthermore, this methodology will prescribe how to deploy the applications in our infrastructure. We plan to evaluate this process by comparing it to existing solutions (e.g. [5]) in terms of completeness of the transformation rules and their correctness with respect to what was intended in the models. At last, the unforeseen situation handler module will be executed for the discovery of new patterns that were not considered during the SA application design time. To evaluate its efficacy, we plan to measure how the detected situations can enhance the application goals, for instance, by interviewing SA application users.

6 Expected Contributions

The main intended contribution of this PhD research is a framework for the development of SA applications for disaster management. More specific expected contributions are: (i) a solution applicable in different disaster scenarios, considering conceptual, specification and implementation aspects of SA applications and a MDE process; (ii) a well-founded approach for a situation's design and its actions, considering the modeling of the context as an ontology; (iii) a distributed environment for a situation's lifecycle management and process; (iv) a mechanism for unforeseen situations detection. Limitations are expected due to the range of problems that may arise during the development of the framework. Addressing all of these specific problems may be too much work for a single PhD project. However, the issues revealed in this work may motivate other (PhD) projects. Examples of expected limitations are: (i) the treatment of detailed security and privacy characteristics when sharing situations from different data sources; and (ii) the particular configurations for elasticity in the cloud. As current work, we are analyzing the MOM broker options to realize the requirement of platform independence from the situation notification service, as well as studying related work on the conceptual, specification and implementation aspects.

References

1. Barwise, J.; Perry, J. (1980). The Situation Underground. Stanford Works Papers in Semantics, v. 1, p. 1-55.
2. Baumgartner, N., Retschitzegger W. (2006). A survey of upper ontologies for situation awareness. 4th International Conference on Knowledge Sharing and Collaborative Engineering (IASTED). St. Thomas, USA.
3. Coskun, E., Ozceylan, D. (2011). Complexity in Emergency Management and Disaster Response Information Systems (EMDRIS). 8th International Conference on Information Systems for Crisis Response Management (ISCRAM).
4. Costa, P. D. (2007). Architectural Support for Context-Aware Applications: From Context Models to Services. PhD thesis. University of Twente, Netherlands.
5. 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). Beijin, China.
6. Drools – www.drools.org – accessed at 2015-4-4.
7. Falbo, R. A., Barcellos, M. P., Nardi, J. C., Guizzardi, G. (2013). Organizing Ontology Design Patterns as Ontology Pattern Languages. The Semantic Web: Semantics and Big Data International Conference.
8. Fan, Z., Zlatanova, S. (2011). Exploring ontologies for semantic interoperability of data in emergency response. Journal of Applied Geomatics.
9. 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). Kristiansand, Norway.
10. Guizzardi, G. (2005). Ontological Foundations for Structural Conceptual Models. PhD thesis, University of Twente, The Netherlands.
11. Guizzardi, G., Wagner, G., Falbo, R. De A., Guizzardi, R. S. S. and Almeida, J. P. A. (2013). Towards Ontological Foundations for the Conceptual Modeling of Events. 32nd International Conference on Conceptual Modeling (ER). Hong Kong, China.

12. Guizzardi, G., (2014). Ontological Patterns, Anti-Patterns and Pattern Languages for Next-Generation Conceptual Modeling. 33rd International on Conceptual Modeling Conference (ER). Atlanta, USA.
13. Hasan, S., Curry, E., Banduk, M., Riain, S. O. (2011). Toward Situation Awareness for the Semantic Sensor Web: Complex Event Processing with Dynamic Linked Data Enrichment. 10^{th} International Semantic Web Conference (ISWC) – Workshop on Semantic Sensor Networks (SSN).
14. Ibbotson, J. Gibson, C., Wright, J., Waggett, P., Zerfos, P., Szymanski, B. K., Thornley, D. J. (2010). Sensors as a Service Oriented Architecture: Middleware for Sensor Networks. 6th International Conference on International Conference on Intelligent Environments (IE).
15. Matheus, C.J., Kokar, M.M., Baclawski, K. (2003). A Core Ontology for Situation Awareness. 6th International Conference on Information Fusion.
16. Menthor – http://www.menthor.net – accessed at 2015-4-4.
17. Moreira, J. L. R., Cordeiro, K. F., Campos, M. L. M., Borges, M.. (2014). OntoWarehousing – Multidimensional Design Supported by a Foundational Ontology: A Temporal Perspective. 16th International Conference on Data Warehousing and Knowledge Discovery (DaWaK/DEXA). Munich, Germany.
18. Moreira, J. L. R., Cordeiro, K. F., Campos, M. L. M., Borges, M.. (2015). Hybrid Multidimensional Design for Heterogeneous Data Supported by Ontological Analysis: an Application Case in the Brazilian Electric System Operation. 18th International Conference on Extending Database Technology (EDBT), workshop Energy Data Management (EnDM).
19. Papazoglou, M.P. (2012). Web Services & SOA: Principles and Technology. Pearson – Prentice Hall, January 2012, 856 pages.
20. Pavkovic, B., Berbakov, L. (2014). Situation Awareness and Decision Support Tools for Response Phase of Emergency Management: A Short Survey. International Conference on Database and Expert Systems Applications (DEXA) - Workshop on Information Systems for Situation Awareness and Situation Management (ISSASiM).
21. 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). San Diego, USA. Best paper.
22. 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). San Antonio, USA.
23. Scherp, A., Saathoff, C., Franz, T., Staab, S. (2011). Designing core ontologies. Journal of Applied Ontology, 177-221.
24. 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). Florida, USA.
25. Valiente, M. C., Machín, R., Barriocanal, E.G., Sicilia, M.A. (2011). An Ontology-Based Integrated Approach to Situation Awareness for High-Level Information Fusion in C4ISR. 5th Ontology, Models, Conceptualization and Epistemology in Social, Artificial and Natural Systems Workshop (ONTOSE). London, UK. Lecture Notes in Business Information Processing Volume.
26. Verdonck, M. (2014). Providing guidance for conceptual modelling using core ontologies. PhD symposium in 33rd International on Conceptual Modeling Conference (ER).

27. Margara, A., Cugola, G., Tamburrelli, G. (2014). Learning From the Past: Automated Rule Generation for Complex Event Processing. 8th ACM International Conference on Distributed Event-Based Systems (DEBS).
28. Carraretto, R. and Almeida, J. P. A. (2012). Separating Ontological and Informational Concerns: Towards a Two-Level Model-Driven Approach. 16th IEEE International Enterprise Distributed Object Computing Conference Workshops (EDOCW), pp. 29-37.
29. Fülöp, L. J., Beszédes, Á., Tóth, G., Demeter, H., Vidács, L., Farkas, L. (2012). Predictive Complex Event Processing: A Conceptual Framework for Combining Complex Event Processing and Predictive Analytics. 5th Balkan Conference in Informatics (BCI).