Ontology-Driven Conceptual Modeling for Early Warning Systems: Redesigning the Situation Modeling Language

João L. R. Moreira[1], Luís Ferreira Pires[1], Marten van Sinderen[1] and Patricia Dockhorn Costa[2]
[1]*Faculty of Electrical Engineering, Mathematics and Computer Science, University of Twente, Enschede, The Netherlands*
[2]*Computer Science Department, Federal University of Espírito Santo (UFES), Espírito Santo, Brazil*

Keywords: Early Warning System, Public Health Surveillance, Situation Modelling Language, Events Processing.

Abstract: An early warning system (EWS) is an integrated system that supports the detection, monitoring and alerting of emergency situations. A possible application of an EWS is in epidemiological surveillance, to detect infectious disease outbreaks in geographical areas. In this scenario, a challenge in the development and integration of applications on top of EWS is to achieve common understanding between epidemiologists and software developers, allowing the specification of rules resulted from epidemiological studies. To address this challenge this paper describes an ontology-based model-driven engineering (MDE) framework that relies on the Situation Modelling Language (SML), a knowledge specification technique for situation identification. Some requirements are realized by revisiting SML, which resulted in a complete redesign of its semantics, abstract and concrete syntaxes. The initial validation shows that our framework can accelerate the generation of high quality situation-aware applications, being suitable for other application scenarios.

1 INTRODUCTION

In public health epidemiological surveillance an early warning system (EWS) is a system of systems that supports the detection, monitoring, decision making, alerting and responding to outbreaks and epidemics (Lai et al., 2015). These capabilities are addressed by EWS components, e.g. sensors` system for collecting data and a complex event processing (CEP) for situation detection.

A challenge for software designers in the development of applications that identify situations in this domain is to clearly specify the rules over the data, which are usually identified as results of epidemiological studies. These rules reflect temporal, causal and existential relationships. Moreover, such components must realize a set of non-functional requirements, e.g. changing rules at runtime and adequate interoperability to integrate with healthcare and emergency systems. Model-driven engineering (MDE) is a common approach to realize these needs (Boubeta-Puig et al., 2015, Bui Thi Mai et al., 2015, Sobral et al., 2015, Martínez-García et al., 2015, De Nicola et al., 2012, Lim Choi Keung et al., 2015). In this paper we describe the advances of our MDE-based development strategy to create and integrate high quality applications on top of EWS. We stress the redesign of the Situation Modelling Language (SML), a graphical language that supports the representation of real-world situations, allowing developers to build systems that react upon these situations. Here we describe how SML can evolve to fulfil the requirements imposed by the epidemiological context.

Ontology-driven modelling techniques are extremely suitable for context modelling (Guizzardi et al., 2015). SML redesign was grounded in foundational ontologies, which provide interpretation for theories of Barwisean situation semantics and Endsley`s situation-awareness, discussed in (Moreira et al., 2015). Although the SML redesign was motivated by requirements from epidemiology, the resulting language is not restricted to this domain, but can still be used in other domains whenever it is necessary to describe situations.

Section 2 summarizes the EWS framework we developed for the improvement of semantics in situational awareness. It includes some of the public health surveillance requirements, the specification strategy, technologies and the methodology used. Section 3 presents our SML redesign in terms of its semantics and syntax. Section 4 presents some related work. Section 5 presents our final remarks.

L. R. Moreira J., Ferreira Pires L., van Sinderen M. and Dockhorn Costa P.
Ontology-Driven Conceptual Modeling for Early Warning Systems: Redesigning the Situation Modeling Language.
DOI: 10.5220/0006208904670477
In *Proceedings of the 5th International Conference on Model-Driven Engineering and Software Development (MODELSWARD 2017)*, pages 467-477
ISBN: 978-989-758-210-3

2 EWS FRAMEWORK

The current United Nations agenda for disaster risk reduction (Sendai) establishes the high-level requirement of integrating global warning and response systems (Bello-Orgaz et al., 2015).

2.1 Requirements

The data sources characterize if the epidemiological surveillance system is event-based (reports, rumours and unstructured data) or indicator-based (traditional practices with official and structured reports). In both cases these systems implement a set of rules over the data to detect initial outbreaks, having epidemiological studies as the foundation to identify health outcomes such as infectious agents, symptoms, risk factors and transmission behaviour. In this context, measures of frequency and association play an important role to describe how likely a person can be infected in accordance to a specific population and to identify cause-effect factors. The measures of frequency are descriptive multivariate functions (risk, odds, rate and prevalence), which consider variables such as the number of new and total cases, the number of individuals at risk and all individuals in a population, the number of individuals with and without the health outcome and the total person-time at risk. Measures of association aim at comparing the association between a specific exposure and a health outcome. An epidemiological study about the risk of Zika outbreaks in Europe is described in (Rocklöv et al., 2016), which uses information about Aedes mosquito populations, climate (temperature and precipitation), peak flow of air travellers and areas for mosquito-borne transmission of Zika.

Once epidemiologists draw results from these studies, they report their analyses through natural language description of the complex rules over the context elements required to detect the outbreak situations. These rules must be implemented by software developers responsible for the development and integration of EWS for epidemiological surveillance, which implies that the rules should be understandable for them. The main functional requirement here is to improve the common understanding of such rules and the involved context, i.e. to achieve successful communication among epidemiologists and software modellers by decreasing the semantic gap among them.

An EWS must provide temporal reasoning over the events being monitored (Lai et al., 2015), which relies on temporal relations to correlate events. For example, the situation of a possible contagion of Zika is characterized by the situation of high risk of Zika infection overlapping the situation of a person diagnosed with Zika within the same geographical area. “Overlapping” is a temporal relation between the two situations, usually referred as an operator from Allen’s calculus for temporal reasoning.

In addition, an EWS must implement temporal existential rules, i.e. rules that only match events occurring in a specific time window (ranges of time units). For example, the case of influenza-like illness (ILI) is defined by WHO as “an acute respiratory infection with: measured fever ≥ 38 C° and cough; with onset within the last 10 days”. Therefore, ILI is characterized by the existence of fever and coughing within the past 10 days. Sliding time windows allows the definition of existential rules for EWS (Liu et al., 2015). An EWS for public health surveillance also needs to allow the definition of descriptive multivariate and aggregation functions, for example to describe average epidemic levels and sums in populations (Marsh et al., 2016).

An EWS requires a dynamic and adaptive approach to be able to change the applications at runtime (Al-Khudhairy et al., 2012). Moreover, an EWS needs to have high interoperability to receive upstream data from e-Health sensor platforms in accordance to electronic health records standards, e.g. HL7; and to send downstream information for medical staff, e.g. EDXL (Wächter and Usländer, 2014). Finally, the specification of the EWS components must be supported by a modelling tool widely adopted in the healthcare community (Martínez-García et al., 2015).

2.2 Architecture

To address the requirements listed above in the development of an EWS for epidemiological surveillance, we proposed an ontology-based MDE framework in (Moreira et al., 2015). Figure 1 (top) shows the specification and implementation phases for the development of components of an integrated EWS in our framework. Figure 1 (bottom) also shows the integrated EWS at runtime. This framework guides the implementation of components (applications) with different roles in an EWS, e.g. an application that monitors properties of patients, as body temperature and other symptoms of Zika, and an application that monitors posts about Aedes mosquitos in social media. As usual in MDE, the implementation is (partially) generated by MDE transformations from the specification models. Here the implementations make use of technologies such as rule-based systems, complex event processing (CEP) and business process management (BPM).

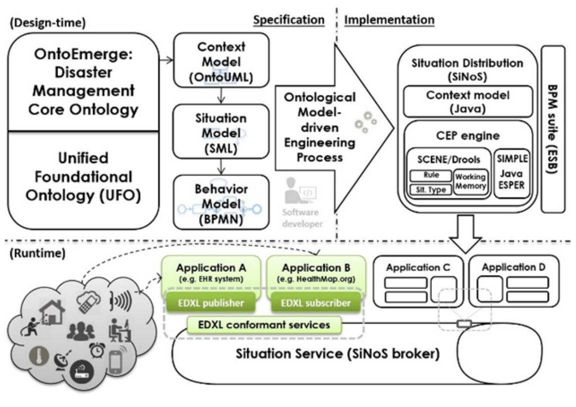

Figure 1: MDE framework for integrated EWS.

Our MDE-based framework has been designed to support the needs of software designers to describe situations, the response actions and to facilitate the generation of CEP code and services compliant to interoperability standards and ontologies. CEP is a technology for detecting event patterns in real time, usually applied for situation inference (Boubeta-Puig et al., 2015). CEP is supported in our framework by the SCENE platform, which extends Drools Fusion, to implement situation-aware applications as rule-based systems. It adds the notion of situation management to the Drools Rule Language through metadata annotations, applied for epidemiological surveillance (Costa et al., 2016). The Situation Notification Service platform (SiNoS) distributes the processing of situation management of SCENE, consisting of a service broker and a message-oriented middleware (MOM) following the publish-subscribe pattern. Business processes can be triggered through an enterprise service bus (ESB) whenever a situation instance is activated. For example, emergency plans specify evacuation actions that can be implemented as data flows within an ESB and called as a subscriber of a service exposed by SiNoS. These services follow appropriate standards according to the type of information. For example, an instance of a situation identified in SCENE is transformed to EDXL data structure, as described in (Moreira et al., 2016).

2.3 Specification Strategy

We consider both structural and behavioural models as suggested in (Brambilla et al., 2012). We propose the use of a high expressive modelling language that includes clear real-world semantic distinctions to model the observed contextual elements and the situation patterns to be identified. We adopted the ontology-driven conceptual modelling approach (Guizzardi et al., 2015), where structural modelling is grounded in a foundational ontology. In particular, we use OntoUML, an extension of UML (a well-founded ontological language), based on the Unified Foundational Ontology (UFO). OntoUML can solve classical problems of structural languages, e.g. ambiguity and low expressivity of UML associations (Guizzardi et al., 2015). OntoUML was chosen to address the expressiveness requirement to specify the context of EWS. Our core ontology, coined OntoEmerge, was designed with OntoUML and includes basic elements for the specification of contextual elements, such as health units, patients, risks, installations and plans. These elements can be mapped from the context model specification onto implementation code as Java classes.

SCENE addresses the requirements of implementing temporal reasoning, existential rules and descriptive multivariate and aggregation functions, as well as the dynamic adaptive behaviour to change rules at runtime, due to the rule-based nature of Drools. The realization of epidemiology requirements with SCENE (Costa et al., 2016) showed that these capabilities could be useful to support the definition of the SML. SML is a graphical language for modelling complex rules, having MDE transformations for SCENE (Costa et al., 2012). In SML a Situation Type (ST) is the representation of patterns of contextual elements in time. A ST allows the characterization of rules among these contextual elements and their changes caused by events. SML relies on the theory of the three levels of situation awareness: "(i) the perception of the elements in the environment within a volume of time and space, (ii) the comprehension of their meaning and (iii) the projection of their status in the near future" (Wickens, 2008).

Originally, the semantic formalization of SML considered a mapping from the syntactic elements of the language to a logic framework (frame-based models). This framework was chosen because frame-based models deal with the temporal aspects discussed above, having frames and their properties as a modelling primitive. Two essential (disjoint, complete) categories of STs were defined: (i) *Simple ST* is interpreted as an open sentence formula which includes free variables (e.g. parameters), the situation itself and the variables representing the contextual elements (characterized by predicates); and (ii) *Complex ST* has the same characteristics of a simple situation but it considers new predicates for each situation that is part of the composition of the complex (whole) situation, i.e. it allows the use of situation participant.

There are two ways to compose a *complex ST* with situation participants (references to other STs), i.e. relate the participants within the whole ST. (1) the use of *equals* relation between entity (or relator) participants of the ST referred by the situation participant, and (2) the use of temporal (Allen) relations between situation participants.

One important aspect of SML is the management of the *situation lifecycle*, i.e. the creation (activation) and the deactivation of an instance of a ST. One or more events in the context characterize the creation of an instance of a ST. For example, the event of increase over 37 degrees of a person`s body temperature triggers the ST of fever, i.e. characterizes the creation of an instance of fever ST. This instance is said to be active (or current) while the constraints hold; and is said to be deactivated (past) when the constraints are not satisfied anymore. A fever ST instance holds in time while the body temperature of a person is greater than 37 degrees, but it is deactivated when the temperature measured over the time stops to satisfy the constraint. An instance is never reactivated, e.g. if a fever instance is deactivated because the temperature measured changed from 38 to 37 and right after the temperature measured is 38, a new instance of fever is created, rather than reactivating the deactivated instance. The temporal relations between STs depend on the status of each ST participant (current or past). This is .reflected in the temporal relations accepted by SCENE.

The framework also includes the design of the reaction to a ST with BPMN, i.e. the processes to be executed when a situation is detected (activated), as emergency plans. The designer can also specify template-based messages to be sent during the process execution. Once the context model is described with OntoUML, the situations with SML and the behaviour with BPMN, the designer can make use of the cyclic verification and validation approach introduced in (Sobral et al., 2015). This approach allows syntax verification of the context model and visual validation through simulation (also called situation assessment). This approach uses the Alloy logic language to automatically generate possible instances of the STs and their participants as object diagrams that the user may inspect to find whether the model represents intended or unintended state of affairs. This is supported by MDE transformations from OntoUML/SML to Alloy. The software designer uses this validation process repeatedly, until sufficient confidence in the models is achieved. Then, SCENE code can be automatically generated from the SML model.

Here we emphasize the fundamental role of SML in our framework. Although the initial version of SML (1.0) covered temporal reasoning and existential rules, it did not possess the necessary expressiveness. SML 1.0 did not consider foundational aspects, as the explicit distinction of context, situation and event, as well as their association (including causal) relationships. These distinctions are necessary to properly formalize the measures of association and causality. (Sobral et al., 2015) discusses the verification and validation of SML with OntoUML, but SML 1.0 still fails to represent, for example, the mutability and cardinality of participants, relationships and self-reference of STs, functions and aggregation functions of collections. These elements are also required for the definition of the measures of frequency within epidemiological studies. These limitations have inspired our SML redesign.

3 SML REDESIGN

The SML redesign is discussed in terms of semantics (meaning of the elements) and its metamodel, as suggested in (Brambilla et al., 2012).

3.1 Foundations (Semantics)

In colloquial language, the term *situation* is often used with the same meaning as *context*. The term *situation* is also often used as a synonymous of *event*, especially in the CEP community. Experience with SML 1.0 showed that this overloaded and ambiguous use of these terms brings difficulties in understanding the language. In this work we explicitly distinguish among these concepts. As foundations to support the choices made here we studied the *Barewisean* situation semantics theory, its extension for the GFO ontology (the *Situoid* theory), the *perdurantism* theory behind UFO and the *Endsley`s* SA theory for human factors (Moreira et al., 2015). Formal ontology and human factors communities discuss whether the *situation* concept is an *endurant* or a *perdurant*. Following *endurantism*, a *situation* is a “snapshot” of the observed world, a static object. In contrast, following *perdurantism*, a situation has a temporal and spatial extent. In particular, the *situoid* theory (Herre and Heller, 2005) deals with this issue by including the concept of *situoid*, a *perdurant* element that is the composition of snapshots (*situations*) in time. S*ituoid* is a combination of infinitesimal slices, where each slice is a *situation* framed by its initial and final timestamps, i.e. a situation is a projection of a *situoid* on time boundaries. A *Situoid* is composed by situations that satisfy the configuration of a set of rules among events changing the contextual elements.

Definition (Contextual Element): In this work we refer to a structural entity (an *endurant*) that participates in a context as a *contextual element* to avoid the overloaded use of the term *entity*, having similar meaning to an *infon*. A contextual element can be perceived as a complete concept even if we were able to freeze time. A context is composed of one or more contextual elements.

Definition (Context): *Context* is "what can be said about an entity in its environment, i.e. context does not exist by itself. The context of an entity may have many constituents, called context conditions. Examples of context conditions of a person are the person's location, mental state, and activity. Together, these context conditions form the entity's context." (Costa, 2007). Context refers to the data elements being perceived from the observed world, i.e. the first level of the SA theory (Wickens, 2008). A context can be represented using a structural modelling language, i.e. context model is a conceptual model of context. Analogous to the ontological categories of *moment* in UFO (intrinsic and relational), we define two main categories of context: *intrinsic context* and *relational context*. An *intrinsic context* belongs to the essential nature of a single entity, not depending on relationships with other contextual elements, e.g. the body temperature of a person. On the contrary, a *relational context* depends on the relation between different entities.

Definition (Event): An *event* (or *perdurant*) is an occurrence. An event can be atomic or complex. An *atomic event* occurs in a moment in time, does not last, i.e. its begin and end time points are the same. A *complex event* is an individual composed of other event instances, i.e. it accumulates temporal parts, extending in time (Guizzardi et al., 2013). An event is responsible for the transition between states of a contextual element. Examples of events are: an increase of a patient's body temperature, the arrival of an ambulance, a recommendation of hospitalization and a bite of a mosquito. Events are responsible for the lifecycle of a relationship (or *relator* in OntoUML), i.e. a relationship exists according to the occurrence of events. For example, the event of a health professional giving first aid to a patient can initialize an instance of a treatment relationship between the patient and a health unit. This instance depends on other events to keep their existence, such as the patient taking medicine.

Definition (Situation): "A situation is a special configuration which can be comprehended as a whole and satisfies certain conditions of unity imposed by certain universals, relations and categories associated with the situation" (Herre and Heller, 2005). Comprehension refers to the second level of the SA theory, i.e. the understanding on how the data elements perceived (contextual elements) relate to each other in a way that it can be recognized as a whole. Therefore, comprehension is the reasoning capability of the phenomena observer – in SML context the software designer and domain experts. Universals refer to the homonymous category in UFO. A situation is a part of the reality that can be comprehended as a whole, has duration and can be past (we use the term *deactivated*) or current (*actived*). Therefore, a situation is an individual that is composed by other individuals, including the states of contextual elements and other situations, constrained by formal and temporal rules. For example, in the situation of "John being infected with Zika", the contextual elements are John, Zika and (possibly) an Aedes mosquito. The event that triggers this situation is the bite of the mosquito on John skin, followed by the event of the virus entering in John's blood flow.

We classify *simple* and *complex situation* as fundamental categories, avoiding the use of the term *type* to reflect the abstraction of a domain-related situation, the initial idea of ST. In order to address this abstraction of types of events, types of situations and their causal relations, we employed the multi-level theory embedded in UFO-MLT (Carvalho et al., 2015). We say that *situation* and *event* are individuals, while *situation type* and *event type* are first-order types. Thus, an *event* is *instance of* (*iof*) an *event type*, while a *situation iof* a *situation type*. "A mosquito biting a person" is an instance of an *event type*. "A mosquito biting John's skin in a specific moment" is an instance of an *event* and, by transitivity, is *iof* that *event type*. Similarly, the *situation type* of "possible contagion of Zika within a geographical area" is a *powertype* of the situation of "a contagion of Zika in Brazil in 2016". With this distinction, we can abstract causal relationships among types of *event* and *situation*, e.g. the *event type* of "a mosquito biting a person" can *cause* the *situation type* of "possible contagion of Zika".

When developing a situation-aware application that relies on data events we need to differentiate an event observed in the real world and its representation as a digital data. For example, the increase of a body temperature of a patient is an event observed in real world, while the specific measurement collected by a thermal sensor attached to the patient is the representation of this event. Therefore, a *situation creation trigger event* is a *simple event* (observed in real word) that evidences a *situation creation event* that activates a situation in the software system. The same approach is used for the *situation deactivation trigger event*, an event perceived from real world (e.g. body temperature decreases to 37) that evidences the *situation deactivation event* (the digital data collected by the sensor and sent to the software), which is responsible for deactivating the situation instance (e.g. fever) in the software system.

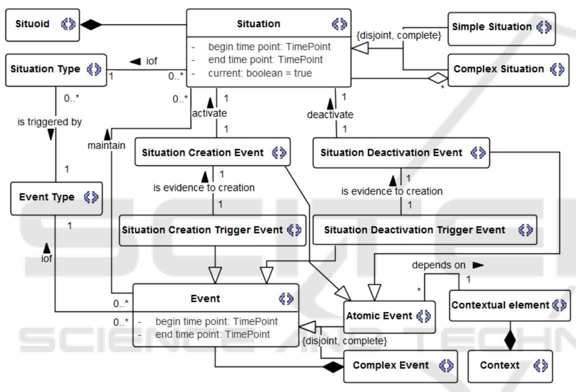

Figure 2: Situations x Events.

Figure 2 illustrates the metamodel with these most fundamental concepts, considering the properties of an *event* and the derived properties of a *situation*. An *event* has a *begin* and *end time points*. The *end time point* of an event is equal to the *begin start time point* of the situation triggered by this event. For example, the situation of "Zika transmission in one person" is triggered by the Aeedes mosquito bite (a *situation creation trigger event*), therefore the *begin time point* of this situation is equal to the *end time point* of the event. The temporal properties of the *event* are also responsible for the *current* property of a *situation*. When an *event triggers* a *situation*, the *situation* instance is created and the *current* property is set to true as default. When a *situation deactivation event* occurs, the *current* property of the situation is set to false and its *end time point* is set to the same value of the *end time point* of the *event*, i.e. the *situation* instance is deactivated and made immutable. This enables to design a *situation* composed of deactivated situations (historical data), such as "intermittent fever" situation, which is a situation of repeatable deactivated "fever" situations within a time period.

An *atomic event* depends on a *contextual element*, i.e. a structural object plays the role of a *participant* of an *event*. For example, a person participates in the event of "body temperature changing" while the "body temperature changing" existentially depends on a person. This definition is aligned with UFO-B in terms of the existential dependency between objects and events (Guizzardi et al., 2013). Similar to the properties of situation derived from *event*, the participants of a *situation* are derived from the contextual elements participating in the events related to the situation (the creation and deactivation trigger events). This property is transitive for composite situations, i.e. participants of a composite situation are derived from the participants of the situations within the composite situation.

3.2 SML Metamodel

The abstract syntax of our SML redesign is described here in terms of its metamodel by discussing the improvements necessary to address the requirements of the epidemiology scenario.

OntoUML improves the expressiveness of the context models. For example, OntoUML allows the dynamic classification of entities through modality types, one of the main benefits of an ontological language (Guizzardi et al., 2015). This is achieved by the clear distinction of rigidity: the instance must instantiate a class while it exists, and non-rigid concepts: the instance can instantiate a class contingently. Table 1 summarizes the mappings between the SML and OntoUML metamodels. In OntoUML a *Quality* is related to a *substantial*, while a *Property* is an attribute of a *substantial* (as a class property). Intrinsic context in the former SML was used to represent both *qualities*, i.e. properties that can be directly evaluated (e.g. temperature and location), and *modes*, i.e. properties that cannot be directly evaluated (measured) in terms of a single space (e.g. a person's headache). Since OntoUML provides this distinction, we absorb it by mapping the *AttributeReference* (SML) to *Quality* or *Property* (OntoUML), and by introducing *ModeReference* (SML), mapping to *Mode* of OntoUML. In addition, a relational context in SML is derived from the notion of *Relator* in OntoUML, which is a moment representing objectifications of relational properties.

SituationTypeComplex specializes *SituationType* and allows the use of *SituationParticipant* in its diagram, which is a reference for a *SituationType* and has properties *isCurrent*, *beginTime* and *endTime*. The property *isCurrent* reflects the *TemporalKind* enumerator, characterizing whether the *SituationType* is present or past. In addition, *SituationParticipant* has the ability of exposing each of the *EntityParticipant* that composes the referenced *SituationType*. A *SituationTypeComplex* also allows the use of each specialization of *AllenLink* (e.g. before, during, overlaps) to relate two different *SituationParticipant*. A *SituationTypeSimple* also specializes *SituationType*, but does not allow the use of *SituationParticipant* neither *AllenLink*.

Table 1: Mappings from OntoUML to SML metaclasses.

SML	OntoUML
ContextModel	Model
AttributeReference	Quality/Property
ModeReference	Mode
QualityLiteral	ReferenceStructure/DataType
TypeLiteral	SubstantialClass
EntityParticipant	SubstantialClass
RelatorParticipant	Relator
ContextFormalLink	FormalAssociation
CharacterizationLink	Characterization
MediationLink	Mediation

A *SituatioTypeElement* can be a *ReferableElement*, a Literal or a *SituationTypeAssociation*. A *ReferableElement* can be a *ModeReference*, an *AttributeReference*, a *Function* or a *Participant*. *ModeReference* and *AttributeReference* are derived from OntoUML (mode and quality, respectively), while *Function* represents a relation between elements not defined in the context model for some reason (e.g. scope limitation). A *Function* can be a calculus or derivation function, i.e. a user-defined operation, between one or more *AttributeReference* and *QualityLiteral*, which play the role of parameters of the function. A *Participant* can be a *SituationParticipant*, an *EntityParticipant* or a *RelatorParticipant*. A *Literal* can be a *QualityLiteral* or a *TypeLiteral*, derived from OntoUML. A *QualityLiteral* can be used only when related through an *OrderedComparativeLink* or *EqualsLink*. A *TypeLiteral* can be used only when related through an *InstanceOf* to an *EntityParticipant*.

Mutability was addressed in SML 1.0 to specify whether exists (or not) the occurrence of a *SituationParticipant* element in a ST with the *exists* logical quantifier, representing the "at least one" instance rule. In SML 2.0 we extended this capability to *EntityParticipant* and *RelatorParticipant* so it is possible to specify if an entity (or relator) exists in a ST. This is defined by the meta-attribute *immutable* in the metaclass *Participant*. Mutability is defined apart from multiplicity (through cardinality) because the goal of cardinality is to allow the specification of minimum and maximum numbers of instances. *Participant* cardinality is addressed through the meta-properties *max* and *min* ([1..1] as default) in the metaclass *Participant*. With mutability it is possible to specify an ILI situation for example, i.e. the model of the rule "if exists fever and coughing within the past 10 days". With cardinality it is possible to specify an intermittent fever. SML also allows the specification of descriptive multivariate functions, i.e. user-defined operations, through the *Function* metaclass. Examples of these functions are sum, square, division or even more complex functions, as multivariate polynomial integrals and derivations. Figure 3 illustrates the main elements of the SML 2.0 discussed here.

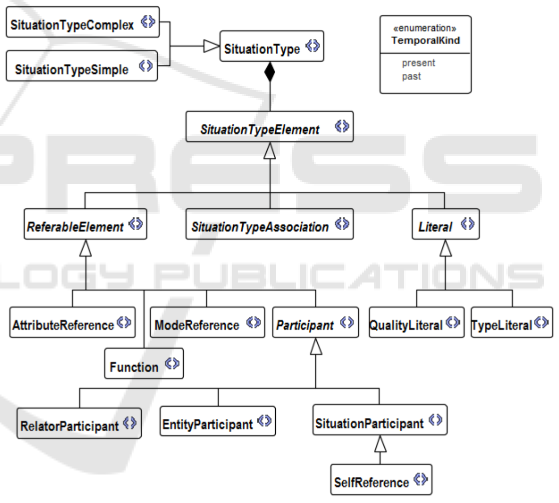

Figure 3: Main elements of SML 2.0 metamodel.

SituationTypeAssociation represents all possible relationships in a SML model. To create a relationship in MDG technology a stereotype needs to extend a metaclass named of *Association*, having the properties of the relation described as attributes of this *metaclass*. For example, the style of the line is described by the attribute *_lineStyle*, source and target multiplicities are set by *_SourceMultiplicity* and *_TargetMultiplicity*, respectively. Direction, reflexivity, symmetry, cyclicity and transitivity are defined through attributes too.

In OntoUML a *Quality*, *Property* and *Mode* are dependent on a relationship to a substantial (e.g. a *Kind* or a *Role*). This substantial is mapped to SML as an *EntityParticipant* (or a *RelatorParticipant*) and an *AttributeLink* relationship is created to link them. Besides *AttributeLink*, SML presents other two relationships that are derived from OntoUML: (i) *CharacterizationLink*, which reflects the characterization relationship between a mode and an entity (links a *ModeReference* to an *EntityParticipant*); and (ii) *MediationLink*, which reflects the mediation relationship between a relator and an entity (links a *RelatorParticipant* to an *EntityParticipant*).

SML 2.0 also allows the creation of comparative relationships through (i) domain-specific formal (as OntoUML formal relationship), (ii) primitive (equals, greater than, less than) and (iii) Allen (before, overlaps, meets, etc.) relationships. This is addressed by, respectively: (i) the *ContextFormalLink*, (ii) the *EqualsLink* and the specializations of the *OrderedComparativeLink* and (iii) the specializations of the *AllenLink*. An example of using a primitive relationship to specify the rule of the fever ST is: person (*EntityParticipant*) body temperature (*AttributeReference*) greater than (*OrderedComparativeLinkGreaterThan*) 37 degrees (*QualityLiteral*). The main difference to SML 1.0 is that each Allen relator was introduced as a metaclass specialization of *AllenLink*, improving the expressiveness of SML.

Regarding the composition of STs, an issue of SML 1.0 was to limit the relation of *EntityParticipant* (or *RelatorParticipant*) within a *SituationParticipant* to *EntityParticipant* (or *RelatorParticipant*) within the complex ST only with the *equals* relationship, which had the side-effect that the *EntityParticipant* (or *RelatorParticipant*) should be bounded by the temporal space in which the complex ST occurs. To address this limitation, SML 2.0 allows exposing not only the *EntityParticipant* (or *RelatorParticipant*) of the *SituationParticipant*, but also the attributes (*AttributeReference*) of these participants. Therefore, in addition to *equals* relation from a participant of a ST, the new metamodel allows to use of *AttributeLinks* of a participant to be exposed in a *SituationParticipant*.

Dynamic classification was partially covered by SML 1.0, by using past situations to indicate whether a situation participant was an instance of some type in the past. This is a limited solution because it is not possible to address the cases in which a participant is no longer an instance of this same type in the present. For example, a ST of "healed patient" is composed by a past situation of "has any ongoing treatment", which exposes the reference to the patient instance, used to link to the person entity (through a comparative relation *equal*). This model allows an instance of a person that is not being treated anymore but can still be a patient, which is an unwanted instance. To address this issue, we introduced the *instance of* relationship in SML, indicating if contextual element is *instance of* a type or not. In the example scenario, the negation of this relation can be applied to the person entity, characterizing that in the current situation the person is not an instance of patient anymore. This approach also solves the problem of representing "past specialization", a common issue in conceptual modelling.

Another issue of SML 1.0 regards that specific formal relations were needed to specify that a past situation (of the same ST) occurred at some point in the past. For example, in (Costa et al., 2012) the formal relation "within the past" had to be created in the example of AccountUnderObservation ST, embedding the semantics of the Allen relation *before*, which is unwanted since it is considered a workaround. To address this problem, the *SelfParticipant* was added as a specialization of *SituationParticipant* with *temporality* set to *current*, allowing the designer to use *AllenLinks* to relate a ST reference to the ST being composed (the "self", which is always the current situation).

Regarding the use of *Function*, the function parameters (input(s)) are represented by the *FunctionParameterLink*, which is a relationship from an *AttributeReference* or a *Literal* to the *Function*. The *FunctionResultLink* is the relationship used to specify the output(s) of a *Function*. *OrderedComparativeLinks* and *EqualsLink* can be used from a *Function* when the *Function* has only one output and the designer wants to specify a comparative relation with this output. For example, the ST "High Body Mass Index (BMI)" uses the "square" Function with a person`s height as input, producing the squared height (as output), which is used as input, along with the person`s weight, for the function "division". The greater than relationship is used from the output of "division" to the literal "25" (if it is greater than 25, it characterizes a high BMI).

3.3 Validation and Discussion

An initial validation to measure whether SML 2.0 satisfies the requirements listed in section 2.2 was performed by using the example scenario of Zika outbreak. Figure 4 illustrates the "possible Zika contagion" ST (ST1). On top, the "High risk Zika infection" ST is based on (Rocklöv et al., 2016), relating three STs in the same geographical region (within 50 km): (ST2) "adequate conditions for Aedes proliferation", dependent on temperature, precipitation and hydrology of the location; (ST3) "Aedes mosquitos reported in the past year"; and (ST4) "Person travelling from infected area". On bottom, "Person diagnosed with Zika" (ST5) is activated if there is a positive result of specific exams for a person. The temporal relation characterizes that ST1 *overlaps* ST5, while the location of ST1 must be near the location of the infected person from ST5.

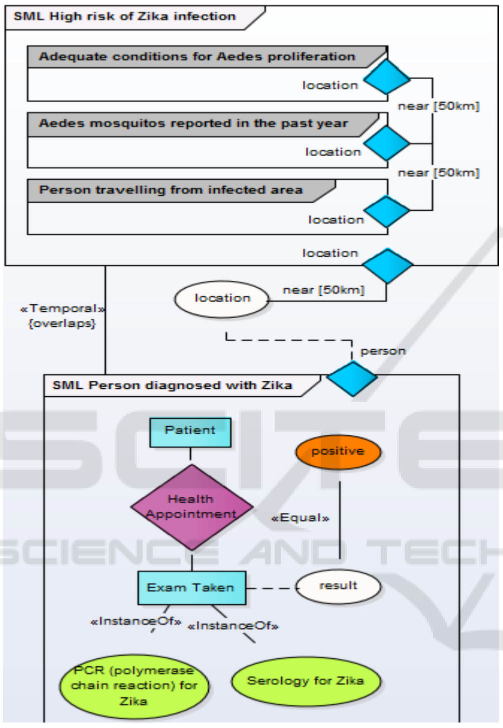
SML High risk of Zika infection
Adequate conditions for Aedes proliferation
location
near [50km]
Aedes mosquitos reported in the past year
location
near [50km]
Person travelling from infected area
location
location
location
near [50km]
«Temporal»
{overlaps}
person
SML Person diagnosed with Zika
Patient
positive
Health Appointment
«Equal»
Exam Taken
result
«InstanceOf»
«InstanceOf»
PCR (polymerase chain reaction) for Zika
Serology for Zika

Figure 4: "Possible Zika contagion" designed with SML.

SML graphical notation was designed according to the easiness of assimilation and comprehension of the chosen symbols (Moody, 2009). Visual cognitive principles were used in this evaluation, such as: (i) semiotic clarity: correspondence between a semantic constructor and a graphical symbol achieved by the one-to-one mapping from the concrete syntax classes to the graphical elements; (ii) perceptual discriminability: the easiness and precision in which graphical symbols can be different to each other is achieved by distinct textures, forms, colours and brightness of the graphical elements of SML. For example, a *SituationParticipant* is a diagram reference of a ST, when having the tagged value *isCurrent* set to true the label background is greyed, otherwise white; (iii) visual expressivity: the number of visual variables among the elements is addressed by distributing the use of these variables among the graphical vocabulary to optimize the cognitive load required for interpretation. The symbols in SML are differentiated by two variables at least. For example, *ModeReference* and *TypeLiteral* are distinct by colour and shape; (iv) graphic economy: the control of the number of distinct graphical symbols is addressed by a balanced approach between adding new symbols or only differentiate them by minor details, such as textual differentiation. Although using text is not considered a good practice for the perceptual discriminability, this choice was made for the existential composition of situations, to avoid the increase of the language complexity. When the *immutable* attribute of the specializations of *Participant* are set to false, the existential symbol (∃) is added before the name of the Participant. Negation (∃!) is also possible with *negationMutable*.

The concrete syntax of SML indicates that the improvement of common understanding of rules can be achieved by the higher expressivity of the language supported by the graphical symbols. Temporal reasoning specification is addressed by Allen`s operators among STs. Temporal existential rule is addressed by mutability in SML (the existential symbol). It is clear that, although we used SCENE technology to support the SML redesign, SML 2.0 is independent of any implementation technologies and is independent of a specific domain. SML`s semantic and syntax address the epidemiology and CEP requirements without referring exactly to them, therefore being general.

4 RELATED WORK

There is a number of MDE approaches in healthcare (Martínez-García et al., 2015), but only few address epidemiological studies. Kendrick language is a DSL that supports epidemiological modelling, through mathematical models for deterministic, stochastic and individual-based simulation (Bui Thi Mai et al., 2015). It differs from ours because it is a DSL that supports epidemiologists on the creation of their studies, while our approach emphasizes on using the findings of epidemiological studies to EWS for real-time epidemiological surveillance. The TRANSFoRm (FP7 project) Query Workbench (Lim Choi Keung et al., 2015) is a MDE platform enabling search of patient`s electronic health records from distributed clinical data repositories. Equivalent to Kendrick, it supports epidemiologic studies, but enabling health data extraction from disparate sources. It is founded on the Clinical Data Integration Model, a core ontology grounded on the Basic Formal Ontology (BFO). Another ontology-based approach for epidemiological data integration was described in (Pesquita et al., 2014) (Epiwork FP7) with the Epidemiology Ontology (EPO). EPO describes epidemiology-specific terms and their relations, supported by the OBO Foundry guidelines, grounded on BFO and based on other sources, as the Infectious Disease Ontology, SNOMED-CT and Unified Medical Language System.

Developing interoperable context-aware applications for e-Health is explored in (Cardoso de Moraes, 2014), a survey of the most common standard-based approaches for healthcare ubiquitous computing, e.g. HL7, Multilevel Healthcare Information Modelling, EN13606 and openEHR. Although none of these standards are specific for epidemiology, they support the communication of upstream data to EWS. In the context of MDE to support disaster risk reduction, the approach of (De Nicola et al., 2012) introduces the Crisis and Emergency Modelling Language (CEML), a graphical language to describe emergency scenarios with critical infrastructures. It includes modelling constructs as security, rescue and expert teams, emergency vehicles and equipment. It extends SysML and uses OCL. CEML does not address temporal reasoning and comparative relations. MEdit4CEP (Boubeta-Puig et al., 2015) addresses this gap through Model4CEP language by abstracting CEP constructs. It introduces a graphical notation with simple and complex events, pattern timers and operators, data windows and aggregation operators. SML has a higher abstraction level (for domain experts) than Model4CEP (for software programmers); therefore, they are a complementary approach. Another complementary approach is RuleML, a XML standard used for forward and backward rules description, supporting the exchange of rules among CEP platforms.

5 CONCLUSIONS

In this paper we described our MDE framework for the development of situation-aware applications that compose public health surveillance early warning systems (EWS). In particular, we described the evolution of the detection specification element, the Situation Modelling Language (SML), for modelling rules resulted from epidemiological studies, improving common understanding among stakeholders. According to (Brambilla et al., 2012), "semantics is often neglected in the definition or in the usage of the language. (…) It doesn't make any sense to define a language without fully specifying the conceptual elements that constitute it and their detailed meaning". Therefore, we gave emphasis to the description of SML semantics and the consequences on the abstract and concrete syntaxes, grounded on the theories of situation-awareness and situation logics and a foundational ontology (UFO).

The current version of SML was initially validated by designing examples of situation types based on epidemiological studies of Zika, Influenza and Tuberculosis. As result of this validation, the main requirement to clearly specify the knowledge generated by epidemiological studies for software development was addressed, due to SML`s higher expressivity and visual cognitive principles. The main limitations identified are the lack of aggregation functions and Boolean operators for composing comparative relations. Future work comprises addressing these limitations and an extended validation of SML in outbreak scenarios in Europe, considering the transformations from SML to SCENE and to EDXL. Future work includes (i) formalization of the SML elements in UFO with first order logic, a common approach in formal ontology community; (ii) formal representation of causation relationship between situation and event, based on an epidemiological causality model, as the Bradford Hill Criteria; (iii) transformations between SML and Model4CEP; (iv) use of RuleML to share the situation identification rules; (v) integration with existing scientific workflow engines.

Although SML was redesigned to realize the requirements of epidemiological surveillance EWS, it can be used by non-IT experts for other types of applications that require situation identification modelling. SML is general enough to specify the detection of other types of emergencies, as floods, landslides and wildfire, as well as logistics monitoring and crime detection. We believe that our framework is suitable for the development of applications to support the aforementioned domains.

ACKNOWLEDGEMENTS

Our thanks to CAPES (process BEX 1046/14-4).

REFERENCES

Al-Khudhairy, D., Axhausen, K., et al. 2012. Towards integrative risk management and more resilient societies. *The European Physical Journal Special Topics.*

Bello-Orgaz, G., Hernandez-Castro, J., et al. 2015. A Survey of Social Web Mining Applications for Disease Outbreak Detection. *Intelligent Distributed Computing VIII.*

Boubeta-Puig, J., Ortiz, G., et al. 2015. MEdit4CEP: A model-driven solution for real-time decision making in SOA 2.0. *Knowledge-Based Systems.*

Brambilla, M., Cabot, J., et al. 2012. *Model-Driven Software Engineering in Practice*, Morgan \& Claypool Publishers.

Bui Thi Mai, A., Stinckwich, S., et al. KENDRICK: A Domain Specific Language and platform for mathematical epidemiological modelling. *In:* Research, Innovation, and Vision for the Future.

Cardoso De Moraes, J. L. 2014. *Methodological support to develop interoperable applications for pervasive healthcare.* PhD.

Carvalho, V. A., Almeida, J. P. A., et al. 2015. Extending the Foundations of Ontology-Based Conceptual Modeling with a Multi-level Theory. *International Conference Conceptual Modeling.*

Costa, P. D. 2007. *Architectural support for context-aware applications: from context models to services platforms.* PhD.

Costa, P. D., Almeida, J. P. A., et al. 2016. Rule-Based Support for Situation Management. *Fusion Methodologies in Crisis Management: Higher Level Fusion and Decision Making.*

Costa, P. D., Mielke, I. T., et al. A Model-Driven Approach to Situations: Situation Modeling and Rule-Based Situation Detection. *In:* Enterprise Distributed Object Computing Conference.

De Nicola, A., Tofani, A., et al. 2012. An MDA-based Approach to Crisis and Emergency Management Modeling. *International Journal On Advances in Intelligent Systems.*

Guizzardi, G., Wagner, G., et al. 2015. Towards ontological foundations for conceptual modeling: The unified foundational ontology (UFO) story. *Journal of applied ontology.*

Guizzardi, G., Wagner, G., et al. Towards Ontological Foundations for the Conceptual Modeling of Events. *In:* International Conference Conceptual Modeling.

Herre, H. & Heller, B. 2005. Ontology of time and situoids in medical conceptual modeling. *Proceedings of the 10th conference on Artificial Intelligence in Medicine.*

Lai, P.-C., Chow, C. B., et al. 2015. An early warning system for detecting H1N1 disease outbreak – a spatio-temporal approach. *International Journal of Geographical Information Science.*

Lim Choi Keung, S. N., Khan, O., et al. 2015. A query tool enabling clinicians and researchers to explore patient cohorts. *International Conference on Informatics, Management, and Technology in Healthcare.*

Liu, S., Smith, K., et al. 2015. A multivariate based event detection method and performance comparison with two baseline methods. *Water Research.*

Marsh, K., Ijzerman, M., et al. 2016. Multiple Criteria Decision Analysis for Health Care Decision Making—Emerging Good Practices. *Value in Health.*

Martínez-García, A., García-García, J. A., et al. 2015. Working with the HL7 metamodel in a Model Driven Engineering context. *Journal of Biomedical Informatics.*

Moody, D. 2009. The "Physics" of Notations: Toward a Scientific Basis for Constructing Visual Notations in Software Engineering. *IEEE Transactions on Software Engineering.*

Moreira, J. L. R., Ferreira Pires, L., et al. 2015. Towards ontology-driven situation-aware disaster management. *Journal of applied ontology.*

Moreira, J. L. R., Ferreira Pires, L., et al. 2016. Improving semantic interoperability of big data for epidemiological surveillance. *I-ESA, BDI4E workshop.*

Pesquita, C., Ferreira, J. D., et al. 2014. The epidemiology ontology: an ontology for the semantic annotation of epidemiological resources. *Journal of Biomedical Semantics.*

Rocklöv, J., Quam, M. B., et al. 2016. Assessing Seasonal Risks for the Introduction and Mosquito-borne Spread of Zika Virus in Europe. *EBioMedicine.*

Sobral, V. M., Almeida, J. P. A., et al. Assessing situation models with a lightweight formal method. *In:* 2015 IEEE International Multi-Disciplinary Conference on Cognitive Methods in Situation Awareness and Decision.

Wächter, J. & Usländer, T. 2014. The Role of Information and Communication Technology in the Development of Early Warning Systems for Geological Disasters: The Tsunami Show Case. *Early Warning for Geological Disasters: Scientific Methods and Current Practice.*

Wickens, C. 2008. Situation awareness: Review of Mica Endsley's 1995 articles on situation awareness theory and measurement. *The Journal of the Human Factors.*