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

Challenges

Developing Situation-Aware (SA) applications to support the disaster management process, considering:

- How to characterize situations and response actions
- How to manage and collaborate the detected situations among SA applications
- How to handle situations not specified at design-time

Our solution

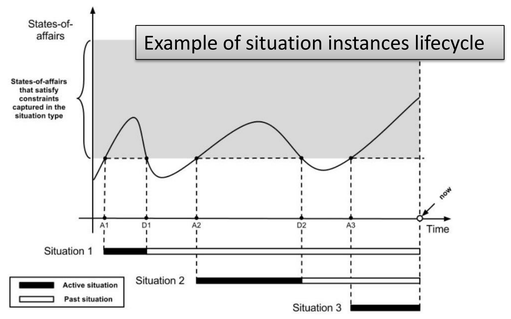
A framework for SA applications development, based on:

- A foundational ontology for temporal conceptualization
- Well-founded structural and behavioral specifications
- A CEP engine as a distributed rule-based platform
- A model-driven approach
- An unforeseen situation module to suggest new situations

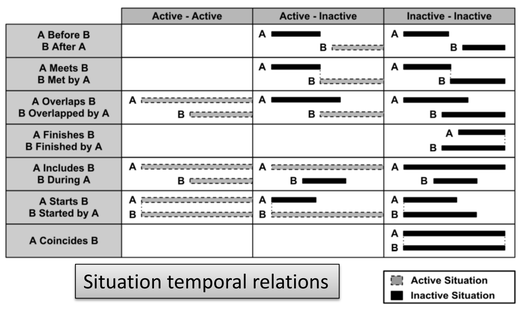
Specification

Foundation Ontology (UFO)

Context Model (OntoUML)



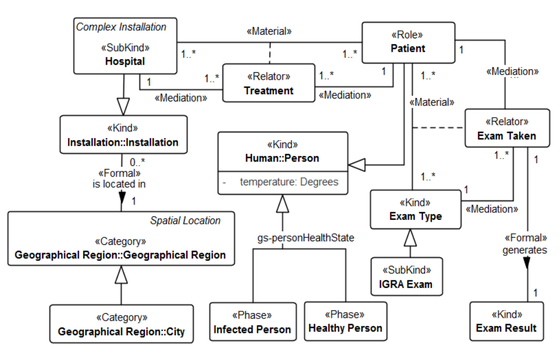
Situation Model (SML)



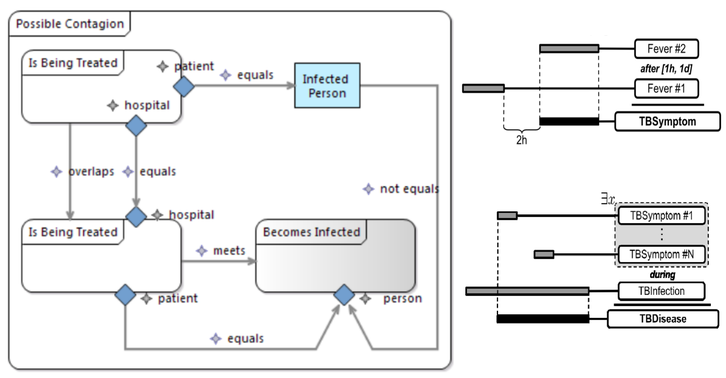
Situation temporal relations

Action Model (BPMN)

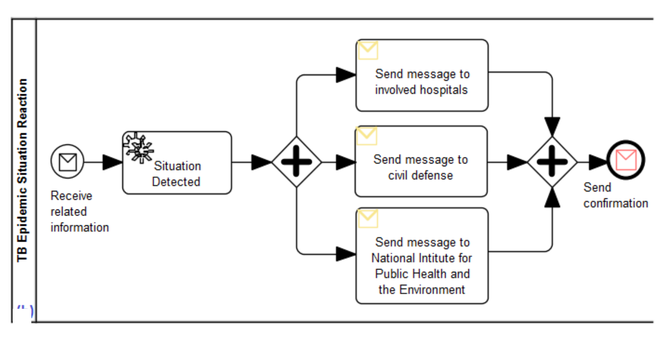
Example scenario: SA application for tuberculosis (TB) epidemy detection



Context (OntoUML): Patient, Hospital, Treatment, Exam Taken, etc.



Situation (SML): Possible Contagion, Being Treated, TB Infection, Fever, etc.

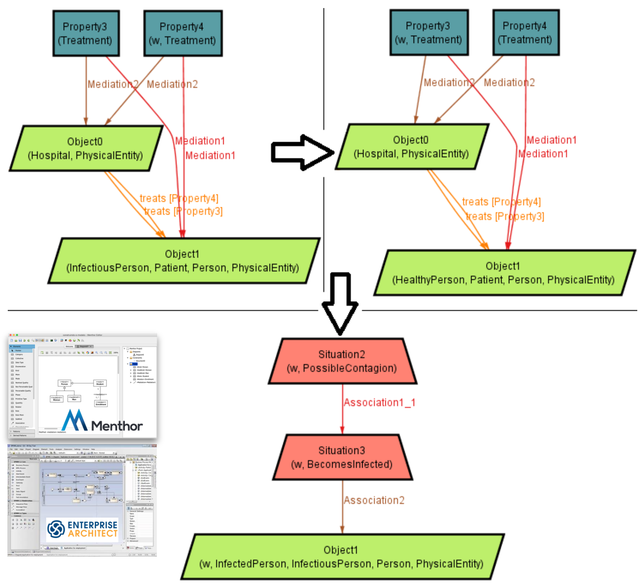


Action (BPMN): Warn hospitals and authorities (e.g. Civil defense).

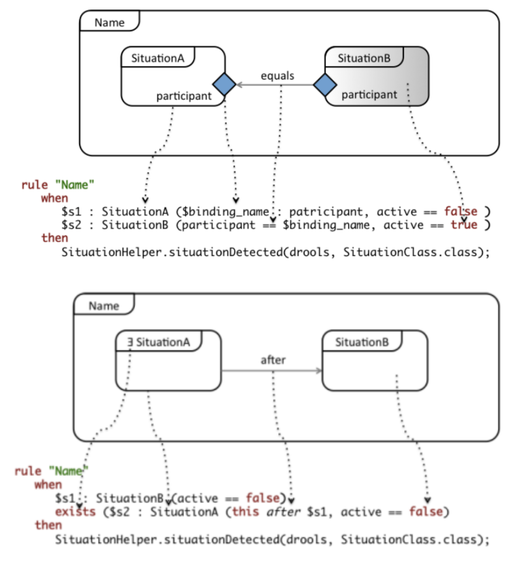
Model Driven Engineering (MDE)

1

A **Verification and Validation**: model assesment to increase its semantics.



B **Code generation**: from models to implementation.



Implementation

2

CEP engine (SCENE/Drools)

Situation Types

Situation Rules

```
1  declare TBDisease extends SituationType
2     patient: Person @SituationRole
3  end
4
5  rule "TBDisease"
6  @role(situation)
7     when
8        $tbi: TBInfection($patient: infected,
9                          active)
10       exists(TBSymptom(febrile==$patient,
11                        this during $tbi))
12    then
13     SituationHelper.situationDetected( drools,
14                        TBDisease.class);
15 end
```

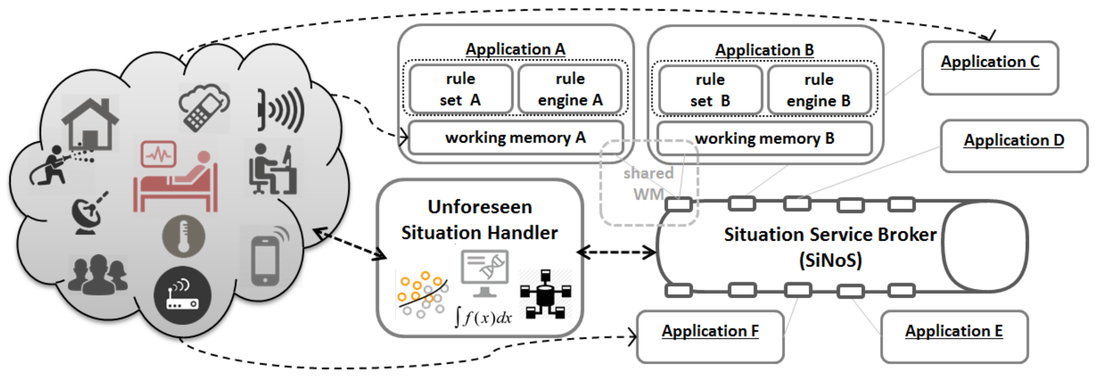
Context Model (Java)

Business Process Model (BPM Suite)

(design-time)

3

(runtime)



Intended contributions

Social benefits: effective and efficient use of available resources aiming at less casualties and damage.

Scientific goals of the framework include:

- Specification of situation and decision making
- Architecture for distributed SA applications
- Discovery mechanism of unforeseen situations

Preliminary results

- Design of the architecture components of the framework
- Extension of a well-founded emergency ontology (OntoEmerge)
- Example case in tuberculosis epidemic scenario

Scientific background

- Temporal conceptualization with the Unified Foundational Ontology (UFO/OntoUML)
- Model-driven engineering (MDE): from Situation Modeling Language (SML) to a rule-based system
- Complex event processing (CEP) mechanism for distributed situation lifecycle management

UNIVERSITEIT TWENTE.