Modeling Collaboration for Crisis and Emergency Management

Antonio De Nicola, Alberto Tofani, Giordano Vicoli, Maria Luisa Villani
ENEA
Roma, Italy
{antonio.denicola, alberto.tofani, giordano.vicoli, marialuisa.villani}@enea.it

Abstract—Managing crisis and emergency requires a deep knowledge of the related scenario. Simulation and analysis tools are considered as a promising mean to reach such understanding. Precondition to these types of tools is the availability of a graphical modeling language allowing domain experts to build formally grounded models. To reach this goal, in this paper, we propose the CEML language and the related meta-model to describe structural aspects of crisis and emergency scenarios. The meta-model consists of a set of modeling constructs, a set of domain relationships, and a set of modeling rules. Finally, we propose a preliminary set of collaboration design patterns to model interaction and communication exchange arising among emergency services providers and citizens to solve the crisis.

Keywords - Conceptual Modeling; Collaborative Networks; Critical Infrastructures; UML Profiles.

I. INTRODUCTION

Recent natural disasters (e.g., earthquakes, floods, fires) and technical faults (e.g., power outages) and their impact on critical infrastructures (CIs) and population have caused a growing attention on how to manage crisis and emergency. In this context, CI services may not work or could not guarantee an acceptable level of service. Since dependencies among CI services are often unpredictable, they could generate further unexpected faults in the CI network. Communications channels could be unavailable to teams needing to collaborate to solve the crisis. Furthermore, beneficiaries of CIs, not provided with the needed resources, can act in uncontrolled mode, hindering the work of operators who are trying to restore CI services.

To cope with such complexity and mitigate such effects, a promising approach is to simulate these scenarios. Simulation allows creating a portfolio of virtual crisis and emergency management experiences to be used, for instance, for training institutional operators with the responsibility of solving the crisis.

A precondition to build effective simulation tools is the availability of a modeling language and a modeling methodology allowing domain experts to build formally grounded models that can be converted into simulation models. The MDA (Model-Driven-Architecture) [1] approach can help us to this aim as it provides methods and tools that can be used by domain experts, i.e., institutional operators with a deep knowledge of crisis and emergency scenarios and with not necessarily high-level IT skills. The first required feature of such language is the domain adequacy, i.e., how the language is suitable to represent the addressed domain [2]. This is achieved by providing experts with modeling constructs and relationships better reflecting their knowledge about the domain. In the CI domain, it is required to allow modeling of collaboration and interaction among CI services, population, institutional operators and stakeholders operating in crisis and emergency scenarios. Then the language has to permit modeling of both structural and behavioral aspects. It has to be formally grounded to allow models to be processed as source code of appropriate simulation programs. It has to be based on widely accepted existing standards to support model interoperability. Finally, it has to be supported by a graphical notation to allow intuitive and user-friendly modeling.

In this paper we propose CEML (Crisis and Emergency Modeling Language), an abstract level language to model crisis and emergency management scenarios. In particular, we describe the related CEML meta-model, consisting of a set of modeling constructs, a set of relationships, a set of modeling rules, and its formalization using SysML [3] and OCL [4]. For sake of space, we focus mainly on presenting how CEML supports structural modeling of a crisis and emergency scenario. Modeling of behavioral aspects will be treated in a future paper.

Then we propose a modeling methodology tailored to model collaboration needed in crisis and emergency scenarios. This methodology is based on Collaboration Design Patterns (CDP)s. A design pattern is a reusable solution to a recurrent modeling problem [5]. In particular, collaboration design patterns model interaction and communication exchange arising during the crisis. Again for sake of space, here we propose just two CDPs: clustered service and heterogeneous networking.

The rest of the paper is organized as follows. Section 2 presents related work in the area. Section 3 describes the meta-model for crisis and emergency scenarios and its formalization. Section 4 proposes a preliminary set of collaboration design patterns for crisis scenarios. Section 5 describes an example concerning emergency management after earthquake events and showing the usability of the proposed modeling framework. Finally, Section 6 presents conclusions and future works.

II. RELATED WORK

Nowadays there is an increasing interest on crisis and emergency management modeling and simulation. The aim is to propose effective modeling and simulation approaches
to analyze crisis scenarios, and to test crisis and/or disaster management procedures.

The main concepts and definitions related to critical infrastructures (CI) are presented in [6]. An interesting approach to describe various aspects of CI is the ontological approach. In [7], for instance, five meta-models are proposed to characterize various aspects of an infrastructure network, such as managerial, structural and organizational aspects. These meta-models are defined as a UML profile with the aim to completely describe the critical infrastructures domain and their interdependencies. Instead, here we concentrate on the problem of graphically building structural models of crisis management scenarios, also involving humans for simulation purposes.

Ontologies to describe either emergency plans or disaster affecting critical infrastructures are presented in [8], [9], [10], and [11].

All these works, which we have considered as a starting point for our research, are complementary to our result, as they provide means to semantically enrich simulation models realized with our language.

Finally, in [12] and [13], SysML is proposed as “standard” meta-model for high level discrete event simulation models to be mapped to Arena and DEVS programs. Indeed, this is proposed to easy the access to simulation technology to non ICT experts and to allow exchange of simulation models between tools.

In addition to what presented by others in the same field, we propose a set of CDPs to support analysts and crisis management experts in modeling crisis scenarios.

III. A META-MODEL FOR CRISIS AND EMERGENCY SCENARIOS

In this section we present the CEML meta-model, to guide modeler in representing the structural aspects of a crisis and emergency scenario. A meta-model is a design framework describing the basic model elements, the relationships between them, and their semantics. Furthermore it defines rules for their use [14].

A. Modeling Constructs

Abstract Service. It represents the active entity processing either a resource entity or a message entity or a connectivity entity. It can be either a service (e.g., power house, information service, electrical power grid) or a human service (e.g., fire brigades) or a communication service (e.g., telecommunications provider).

Behavior. It represents an operational feature of either a service or a human service or a communication service or a user entity. This allows to complete the structural model with behavioral specifications.

External event. It represents the active entity (e.g., failure, earthquake) affecting the operational status of either a service entity or a human service entity or a communication service entity or affecting the wellness of a user entity.

User. It represents the entity using or consuming a resource entity (e.g., hospital). It is characterized by a wellness level.

Message. It represents information content exchanged in a communication.

Resource. It represents the passive entity processed by either a service entity or a human service entity. It can be input to either another service entity or a communication service entity or a human service entity or a user entity. It can contribute significantly to user’s wellness level.

Connectivity. It represents, from a physical perspective, the output of a communication service entity.

B. Relationships

Resource Flow. It represents resource passing through ports from a service or human service entity to either a user or a service or a human service or a communication service entity.

Connectivity Flow. It represents, from a physical perspective, the communication channel provision (through ports) from a communication service entity to either a service or a human service or a user or another communication service entity.

Message Flow. It represents, from a logical perspective, the exchange of information content through ports between two of the following entities: service, human service, and user (e.g., between two services, between a service and a user).

Abstract Port. It represents the abstract entity linking either an abstract service entity or an user entity to either one or more connectivity flow entities, or one or more message flow entities, or one or more resource flow entities. It can be either a message port or a a communication port or a resource port.

Communication Port. It represents the abstract entity linking either a communication service or a human service or a service or a user entity to one or more connectivity flow entities.

Message Port. It represents the abstract entity linking either a service or a human service or a user entity to one or more message flow entities.

Resource Port. It represents the abstract entity linking either a service or a human service or a communication service or a user entity to one or more resource flow entities.

Communication Port Group. It represents the abstract entity grouping one communication port entity and one or more message port entities and belonging to either a service or a human service or a user entity.

Impact. It represents how an external event entity affects one or more of the following entities: service, communication service, human service, and user.

C. Modeling Rules

C1. An element can be categorized only as a modeling construct or as a relationship.

C2. A service element has 0..n incoming resource port elements, 1..n outgoing resource port elements, and 0..n connection port group elements.

C3. A human service element has 0..n incoming resource port elements, 1..n outgoing resource port elements, and 0..n connection port group elements.
C4. A communication service element has 0..n incoming resource port elements and 1..n outcoming communication port elements.

C5. The service element, the human service element, and the communication service element are specializations of the abstract service element.

C6. The message port element, the communication port element, and the resource port element are specializations of the abstract port element.

C7. Every abstract service element is characterized by 0..n behavior elements.

C8. Every abstract service element is affected by 0..n external event elements by means of the impact element.

C9. A user element has 0..n incoming resource port elements and 0..n connection port group elements.

C10. A user element is affected by 0..n external event elements by means of the impact element.

C11. A message flow element is linked to 1..n message elements and holds between two message port elements belonging to two connection port group elements.

C12. A resource flow element is linked to 1..n resource elements and holds between 2 resource port elements. The resource flow element is directed from a resource port element belonging either to a service or human service element and to a resource port belonging either to an abstract service element or to an user element.

C13. A connectivity element is directed from a communication port element, belonging to a communication service element, to a message port element, belonging to a connection port group.

D. Meta-model formalization

In order to equip the language with a sort of formal grounding, so that smart editors could be defined with validation facilities, we identified SysML [3] a standard language sponsored by OMG (Object Management Group), as a good candidate. SysML comes as a profile of UML 2.0, that is, extends the UML meta-model with constructs to enable “system” other than “software” modeling and provides some new diagram types. Therefore, SysML inherits all the advantages of UML: the multi-views representation of a system model; the simplicity of the notation, which is addressed to stakeholders with different levels of technical knowledge; the xml schema for tools interoperability (XMI); and finally the “semi-formal” specification, which has been better clarified starting from version 2.0, that allows model-driven development to take place. Our meta-model is an application of SysML profile tailored to critical infrastructures modeling and, as such, it is a domain-specialization of a subset of SysML. We do this by creating a new profile following the stereotype extension mechanism specified by UML.

Specifically, we consider the components of the Internal Block Diagram of SysML, which is based on the Block entity. According to the OMG specification, blocks “are modular units of a system description, which define a collection of features to describe a system or other elements of interest. These may include both structural and behavioral features, such as properties and operations, to represent the state of the system and behavior that the system may exhibit”.

Figure 1 shows the relationship of the User and AbstractService constructs of our meta-model with the Block entity of SysML. They can have a behavior specified and can be connected with other blocks through ports. However, differently from services, a User does not provide functions/resources to other model elements. Note that the User construct in our meta-model cannot be mapped to the UML (or SysML) Actor meta-class as we intend the User be inside the model (and not part of the environment).

Flow ports are introduced in SysML as a specialization of UML ports “to specify the input and output items that may flow between a block and its environment”. Flow ports are generally typed with respect to the item that can flow (in, out, or inout). In our meta-model we decided to introduce three port types as shown in Figure 2.
that can flow through an atomic port (e.g., water, power) in SysML is specified by the \textit{FlowProperty} stereotype, which can be simply a label. In our case, we want to distinguish between: \textit{message}, \textit{connectivity}, and \textit{resource}, which we define as a specialization of \textit{FlowProperty}. Instead, non-atomic \textit{Flow Ports} in SysML are defined through a FlowSpecification object, which is a collection of FlowProperty objects, each referring to a single item. In SysML, items flow through \textit{Connectors}, used to link blocks. For graphical convenience only, we defined a SysML connector specialization for \textit{message flow} to represent it as a dashed arrow line (see Table II below).

As we want to design analysis scenarios for crisis management, we need to represent the events that may happen and what services/users they may affect. Here we want to represent just the type of the \textit{external event}, such as earthquake, flood, and so on, and its “affecting” relationship to one or more scenario entities. Therefore, we intend the event being an abstract element outside the model (part of the environment) but influencing it, and so this definition specializes that of the \textit{Actor} in UML.

Finally, each kind of service or user element, being a UML Class, might be modeled internally through a \textit{Behavior} object, which is the link to one or more behavioral descriptions of the scenario that we will treat as future work.

The following tables include the list of all the constructs and relationships of the proposed meta-model, with the corresponding formal notation describing the extension from the SysML profile and UML references, and the graphical symbol we associated to them to be used in our diagrams.

### TABLE I. MODELING CONSTRUCTS

<table>
<thead>
<tr>
<th>Modeling Constructs</th>
<th>SysML Specification</th>
<th>Graphical Notation</th>
</tr>
</thead>
<tbody>
<tr>
<td>Abstract Service</td>
<td>SysML::Blocks::Block::AbstractService</td>
<td>NA</td>
</tr>
<tr>
<td>Service</td>
<td>SysML::Blocks::Block::AbstractService::Service</td>
<td></td>
</tr>
<tr>
<td>Human Service</td>
<td>SysML::Blocks::Block::AbstractService::HumanService</td>
<td></td>
</tr>
<tr>
<td>Communication Service</td>
<td>SysML::Blocks::Block::AbstractService::CommunicationService</td>
<td></td>
</tr>
<tr>
<td>Behavior</td>
<td>UML::CommonBehaviors::BasicBehaviors::Behavior</td>
<td>NA</td>
</tr>
<tr>
<td>External Event</td>
<td>SysML::Actor::ExternalEvent</td>
<td></td>
</tr>
<tr>
<td>User</td>
<td>SysML::Blocks::Block::User</td>
<td></td>
</tr>
<tr>
<td>Message</td>
<td>SysML::Property::FlowProperty::Message</td>
<td></td>
</tr>
<tr>
<td>Resource</td>
<td>SysML::Property::FlowProperty::Resource</td>
<td></td>
</tr>
<tr>
<td>Connectivity</td>
<td>SysML::Property::FlowProperty::Connectivity</td>
<td>NA</td>
</tr>
</tbody>
</table>

### TABLE II. RELATIONSHIPS

<table>
<thead>
<tr>
<th>Relationships</th>
<th>Definition …</th>
<th>Graphical Notation</th>
</tr>
</thead>
<tbody>
<tr>
<td>Resource Flow</td>
<td>SysML::Ports&amp;Flows::ItemFlow::ResourceFlow</td>
<td><img src="image" alt="ResourceFlow" /></td>
</tr>
<tr>
<td>Connectivity Flow</td>
<td>SysML::Ports&amp;Flows::ItemFlow::Connectivity</td>
<td><img src="image" alt="ConnectivityFlow" /></td>
</tr>
<tr>
<td>Message Flow</td>
<td>SysML::Ports&amp;Flows::ItemFlow::MessageFlow</td>
<td><img src="image" alt="MessageFlow" /></td>
</tr>
<tr>
<td>Abstract Port</td>
<td>SysML::Ports&amp;Flows::FlowPort</td>
<td></td>
</tr>
<tr>
<td>Connection Port Group</td>
<td>SysML::Blocks::Block::ConnectionPortGroup</td>
<td></td>
</tr>
<tr>
<td>Message Port</td>
<td>SysML::Ports&amp;Flows::FlowPort::MessagePort</td>
<td></td>
</tr>
<tr>
<td>Communication Port</td>
<td>SysML::Ports&amp;Flows::FlowPort::CommunicationPort</td>
<td><img src="image" alt="CommunicationPort" /></td>
</tr>
<tr>
<td>Resource Port</td>
<td>SysML::Ports&amp;Flows::FlowPort::ResourcePort</td>
<td><img src="image" alt="ResourcePort" /></td>
</tr>
<tr>
<td>Impact</td>
<td>SysML::Association::AffectingImpact</td>
<td><img src="image" alt="Impact" /></td>
</tr>
</tbody>
</table>

In a UML profile, “well-formedness” rules, such as the constraint listed in sub-section C, can be encoded in OCL, which is a declarative formal language to express properties of UML models. An OCL rule is defined within a context, that is, the element to which some Boolean expression, specified by the rule, should apply. For sake of space, we give here only one example of OCL implementation of the constraints of our meta-model.

#### C4. A communication service element has 0..n incoming resource port elements and 1..n outcoming communication port elements.

**Context** SysML::Blocks::Block

\begin{verbatim}
selfoclisTypeOf(CommunicationService) implies
    (self.attributes->select(oclisTypeOf(MessagePort))->size()=0) and
    (self.attributes->select(oclisTypeOf(CommunicationPort).direction='"out"')->size()>0)
    and (self.attributes->select(oclisTypeOf(CommunicationPort).direction='"in"')->size()=0)
    and (self.attributes->select(oclisTypeOf(ResourcePort).direction='"out"')->size()=0)
    and (self.attributes->select(oclisTypeOf(ResourcePort).direction='"in"')->size()=0)
    and (self.attributes->select(oclisTypeOf(CommunicationPort).direction='"in"')->size()=0)
\end{verbatim}

![ResourceFlow](image)

![ConnectivityFlow](image)

![MessageFlow](image)

![ResourcePort](image)

![CommunicationPort](image)
IV. CRITICAL INFRASTRUCTURES COLLABORATION DESIGN PATTERNS

Design patterns are proving to be one of the most promising methodological tools to support building of models and, more in general, ICT artifacts like software programs. Currently, there are several proposals of design patterns in different fields, e.g., UML design patterns for software engineering [15], workflow patterns for business process management [16], and ontology design patterns for ontology building [17]. Here we propose to use a particular type of design pattern, the collaboration design pattern devoted to facilitate modeling of interaction and communication exchange arising among emergency services providers and citizens to solve the crisis. In particular, a CDP allows to represent a chunk of the reality where collaboration is performed. Using this approach, modelers can create a repository of CDPs to be reused to describe similar scenarios. As stated in the introduction, here we propose only two CDPs that are described in the following.

CDP1. Clustered Service

Figure 3 shows the clustered service CDP devoted to model collaboration arising among different services working together to either provide or produce or transport a resource. In particular, the objective of this CDP is to model exchange of resources and information. Furthermore, this CDP models the physical connection provided by a communication service and allowing information exchange.

CDP2. Heterogeneous Networking

Figure 4 presents the heterogeneous networking CDP modeling a network of different communication services, guaranteeing the physical connection between two services.

V. EMERGENCY SCENARIO EXAMPLE

The objective of this section is to demonstrate usability and flexibility of the proposed modeling framework by describing an actual emergency scenario after an earthquake [18]. In particular, we focus on the main services, resources and users related to the Italian Civil Protection (ICP) emergency management protocol. For the sake of brevity, we omit some details as our aim is to demonstrate the usability and flexibility of the proposed modeling framework. A detailed description of the scenario is available in [18]. After an earthquake event, the ICP is able to have a global picture of the impact of this event by using sensor networks, simulation tools, and specific expert team reports. The Mixed Operative Center (COM in Figure 5) is established near the areas mostly damaged by the earthquake. In this example, the COM plays the role of final user. Then there are the Emergency Services, the Emergency Call Service, and the Lifeline networks. The Emergency Services represent all actors involved in the emergency management protocol. We describe the details about this service using the clustered service CDP (Figure 6). The Emergency Call Service represents the network of emergency call centers devoted to receive feedbacks from user in order to assess how well ICP is facing the emergency. The Lifeline Networks element models the infrastructure networks (e.g., electrical distribution and telecommunication network, gas and water pipelines, water treatment systems) of the damaged area. Evaluation of the lifeline performances is one of the most important tasks during an emergency to allow rescue teams to properly and safely operate during an emergency. The networks and their dependencies can be further specified using an appropriate clustered service CDP. The Telco Network communication service models the connectivity services and resources operating in the area.
Using the clustered service CDP, it is possible to refine the definition of the Emergency Services to model the coordination messages that are exchanged among the major actors during emergency management (Figure 6). The decisional board is represented by the National Civil Protection Service (SNPC). The coordination messages aim to gather information about available resources at a national, regional, provincial, local level. The Direction and Command on site (DiComaC) service is in charge of resources distribution and operations management. All decisions rely on the information about the lifeline performance provided by the Lifeline Owners service. The Figure 6 shows also the output resources of the Emergency Services to the COM.

Currently, CEML supports modeling structural aspects of a scenario. We are going to extend the language and the related meta-model to behavioral aspects in a future work. In particular, we intend to use ECA (Event Condition Action) rules [19] and the expressive power of a domain ontology to allow advanced reasoning. Finally, we are developing a simulation tool to permit these models to be simulated.

REFERENCES