The International Emergency Management Society
2015 Annual Conference, 30th September - 2nd October 2015, Rome, Italy

Towards the Next Generation Training System for Crisis Management

Jingquan Xie[1] and Erik Vullings[2] and Marianthi Theocharidou[3]

[1] *Fraunhofer IAIS, Sankt Augustin, Germany*

[2] *TNO, the Netherlands*

[3] *European Commission, Joint Research Centre, IPSC, Ispra, Italy*

Email: jingquan.xie@iais.fraunhofer.de, erik.vullings@tno.nl, marianthi.theocharidou@jrc.ec.europa.eu

ABSTRACT :

Training plays a central role to improve the skills and responsiveness of crisis managers during critical situations. Well-trained crisis managers can handle crisis more thoroughly and effectively, especially when critical infrastructures and cascading effects need to be considered. Current training approaches are mainly based on real world exercises, which are in general difficult and time-consuming to organise and to perform. Modern technologies like computer modelling simulation and real-time communication over Internet provide a promising alternative to perform training in a more efficient and cost-effective way. This is particularly important for cross-border training where people from different countries need to participate. In this paper we propose a novel system architecture for building such next generation training systems for crisis management. This architecture embraces federated modelling and simulation technology to provide system dynamics of real world infrastructures like electrical networks, telecommunication networks and flooding developments, etc. Real-time communication is integrated into this architecture to enable collaborations for distributed training. Under simulated crisis situations, both spatial and temporal aspects of critical infrastructures elements can be handled and presented in a standard and consistent manner. Mitigation actions issued by crisis managers are performed in a simulated environment with complete rollback support (what-if analysis). Furthermore, impacts and consequences of cascading effects caused by the actions can be analysed according to different criteria in a transparent way.

KEYWORDS:

Crisis management training, What-if analysis, Complex Event Processing, Modelling Simulation and Analysis, Collaborative training, Critical Infrastructure Protection

1. Introduction

Crisis management is a challenging task in modern society. As more critical infrastructures (CI) like power grid, telecommunication networks, etc. with sophisticated dependencies are deployed, it becomes more difficult for crisis managers to make proper decisions under crisis situations when cascading effects need to be considered. The situation is even worse if the crisis management needs to be done in a cross-border fashion where different countries have their own rules and regulations. Training plays a central role in crisis management: it is crucial to improve the skills of crisis managers during the preparedness phase, in which only pre-designed crisis scenarios are used. Well-trained crisis managers can handle the situations, like making an evacuation decision under real crisis, more effectively and efficiently - especially in the cross-border case where crisis managers from different countries need to collaborate with each other. Currently, to provide more realistic training experiences in practice, a commonly used approach is to organise real world exercises [Stolk et al., 2001]. However, due to organisational overhead and high costs, these real world training activities often take place at a rather low frequency - typically once or twice a year [Robert and Lajtha, 2002]. For cross-border scenarios involving multiple countries, the frequency is even lower. Furthermore, it usually takes a long time to analyse the training results based on real-world training activities to assess the effectiveness of the training. For that reason, crisis managers and other participants of the real-world training cannot get the feedback immediately, which can have a significant impact on the effectiveness of their skill improvements.

To overcome the drawbacks of the traditional approach, a novel training system architecture is proposed in this paper. It aims to facilitate the development of modern crisis management training systems. This architecture is designed by utilising two major emerging technologies: 1) computer-based modelling simulation and analysis (MS&A), federated simulation in particular and 2) Web-based real-time communication (WebRTC). Federated simulation is adopted to provide sophisticated system dynamics of CI at runtime by taking the comprehensive inter-dependencies into consideration. Multiple domain-specific simulators can be integrated into the architecture to provide both holistic and detailed dynamic views of the whole system under simulated crisis situations. Web-based real-time communication provides a solid technological foundation for the development of a collaborative platform enabling both efficient and effective information exchange among different training participants. The final training protocols are managed in a central database and can be analysed simultaneously as the training progresses. It is used to provide timely feedback to all training participants. One major advantage of this architecture is its agnostics to crisis scenarios and hazards, i.e. systems developed based on this architecture can be applied to different scenarios involving different CI. The only component that may need to be modified is the simulation adapters.

This paper is organised as follows: Section **Errore. L'origine riferimento non è stata trovata.** gives an overview of the training systems based on federated simulations. Section **Errore. L'origine riferimento non è stata trovata.** elaborates the application of a scenario-driven approach focusing on scenario description and integration. Section **Errore. L'origine riferimento non è stata trovata.** illustrates the real-time collaborative platform followed by the discussion of related works of computer-based crisis management training systems in Section **Errore. L'origine riferimento non è stata trovata.**. Finally, Section **Errore. L'origine riferimento non è stata trovata.** concludes this paper and provides insights of possible future directions.

2. Support Training with Federated Simulations

Federated simulation [Rome et al., 2014] is a technology to provide sophisticated dynamics of a complex system of systems by coupling multiple domain-specific simulators. One emerging application area of federated simulation is the analysis of interdependencies of critical infrastructures [IRRIIS, 2006] and the cascading effects caused by natural and human-made hazards [DIESIS, 2008]. Training systems for crisis management require the capability to provide comprehensive system dynamics under simulated crisis situations.

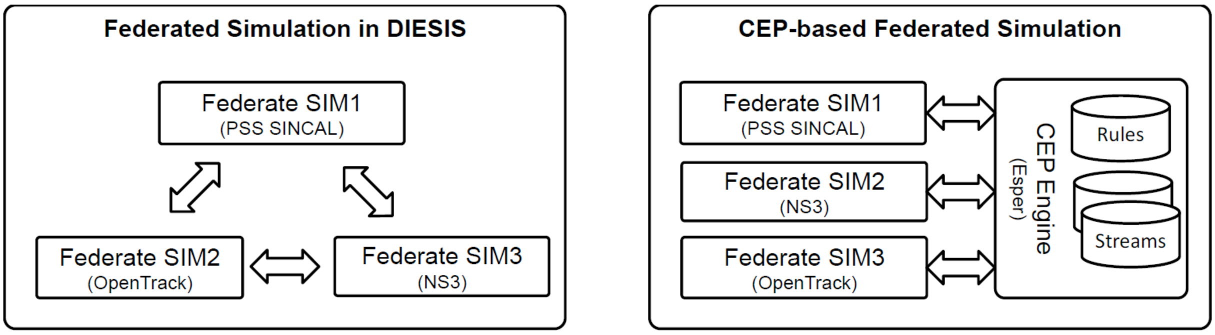
2.1. *Overview of Federated Simulation*

Among other approaches, dedicated computer-based simulators provide the most comprehensive way to provide fine-grained details about the physical world. Various computer-based simulators - commercial and open source - exist for different domains like load flow calculation of electrical networks [SimTec, 2015], telecommunication networks [NS-3 Consortium, 2015], and railway systems [Nash and Huerlimann, 2004], etc. However they are independently developed to simulate the dynamics of a specific domain instead of targeting a larger system of systems, which would enable the simulation of (inter-)dependent CI.

Federated simulation emerges as a solution to solve the problem. The core concept is to laterally couple multiple domain-specific simulators to provide a holistic view of complex system dynamics involving multiple domains. This is exactly the case for modern crisis management involving CI with sophisticated (inter-)dependencies. Several standards have been proposed during the last decades to provide a more unified way to integrate simulators. The most prominent one is High-Level Architecture HLA [IEEE, 2010], which arose from the military applications. The standard however does not gain its popularity in industry due to the over-complication of dealing with federates [Kuhl et al., 2000]. For the purpose of analysing the (inter-)dependencies of critical infrastructures, the DIESIS project [DIESIS, 2008] proposed another simplified approach to couple various simulators in a lateral fashion. In the following subsections, the adoption of federated simulation into crisis management training systems is based on a modified version of the DIESIS approach.

2.2. *Federation with Complex Event Processing*

The federation approach in DIESIS provides a solid foundation for integrating simulators on both functional and technical levels (see Figure 1). One of the primary drawbacks however is its in-flexibility to handle different scenarios: most of the logics on the federation level in DIESIS are hard-coded. That means, in order to work with new scenarios with different storylines and (inter-)dependencies, most of the federation codes need to be modified, tested and re-built. This is not acceptable for training systems that normally need to deal with multiple training scenarios. To solve this problem, the concept of Complex Event Processing (CEP) [Luckham, 2001] is employed to provide a declarative fashion to design the federation logics, see Figure **Errore. L'origine riferimento non è stata trovata.**.



(a) Federation simulation with de-centralised communication.

(b) Federation simulation based on Complex Event Processing.

Figure 1: The side-by-side comparison of de-centralised and CEP-based federated simulation approach. Declarative rules in CEP engines models the federation logics.

CEP aims to provide a systematic approach to derive new information by correlating events and detecting patterns in large-scale event clouds. After more than a decade research, several CEP engines [EsperTech, 2012, Adi and Etzion, 2004] have been developed. In the federation simulation environment, simulators need to exchange information and synchronise with each other. These data payloads of these operations and behaviours can be modelled as events. For instance, simulator `SIM1` wants to incrementally send the simulation result to simulator `SIM2`. In the original DIESIS design, `SIM1` communicates with `SIM2` directly in a de-centralised fashion, see Figure 1. A block of hardcoded logics in `SIM2` handles the simulation results and further update its own model. This is a reasonable solution for small-scale federated simulation environment with a handful of federates. For large-scale federated simulation environments however, this design is less flexible. To improve this, a component with one dedicated CEP engine is introduced, see Figure **Errore. L'origine riferimento non è stata trovata.**. There are two major improvements in this new design:

- **Decoupled system** - federates do not need to know each other directly. They communicate with each other through the CEP engine. Decoupled systems are less fragile and can be extended by adding new functionalities with less effort.
- **Declarative rules** - the federation logics are not hardcoded in the federates any more. Instead they are implemented as declarative rules in the CEP engine. This brings a huge amount of flexibility into the federation system like dynamic changing of the federation logics, hot integration of additional federates, etc.

The declarative event processing rules play a central role in the new design. It is called *declarative* because the event patterns can be specified in a declarative fashion. For instance, one typical operation in CI-based federated simulation is to determine the state of a given CI [Nieuwenhuijs et al., 2008]. One solution is to use a Finite State Machine (FSM) based on the incoming events, then it is necessary to maintain the states locally in a high-level imperative language. Another solution is to specify the pattern of incoming events declaratively. Based on the event patterns in the stream, the state of CI can be derived accordingly. The pseudo code of the rule is illustrated below:

```
ON ci_under_water FOLLOWED BY ci_under_water
IF INTERVAL IS OVER 3 HOURS AND ci.state == `normal'
THEN ci.state = `stressed'
```

It basically denotes that if any CI with state `normal` has been flooded for over three hours, change the state to `stressed`. It is not necessary to imperatively provide the algorithm to detect two events `ci_under_water` in the event stream to match the specified pattern. By using this kind of declarative approach, the readability and maintainability of federation logics can be dramatically improved. Furthermore, dynamic loading of these rules into the training system from different scenarios becomes possible, thereby increasing flexibility (see Section **Errore. L'origine riferimento non è stata trovata.**).

2.3. Integrating Federated Simulation into Training Systems

Crisis management training systems need to provide simulated information of physical systems on different levels of granularity. For crisis managers, a holistic overview of the situation within the crisis area is required at first. After obtaining situational awareness, detailed information may need to be retrieved by drilling down into specific areas and certain CI, depending on the tasks they need to perform. For instance, to evacuate an area, the availability and traffic information of certain streets need to be explored in more details.

Integrating federated simulation into crisis management training systems is a challenging task. The difficulties lay not only on the technical level but also on the functional level. Simulators per se are not designed for training application. Some of them run reasonably fast while others are quite slow. They need to be synchronised together with the other components of a training application. Simulation models need to be exchanged between the federated simulation environment and the training system to provide detailed information about the simulated physical system. On the other side, mitigation actions performed by the decision makers, crisis managers in particular, need to be injected into the federated simulation environment to change the simulation model and trigger another round of simulations. All these need to be coordinated and synchronised.

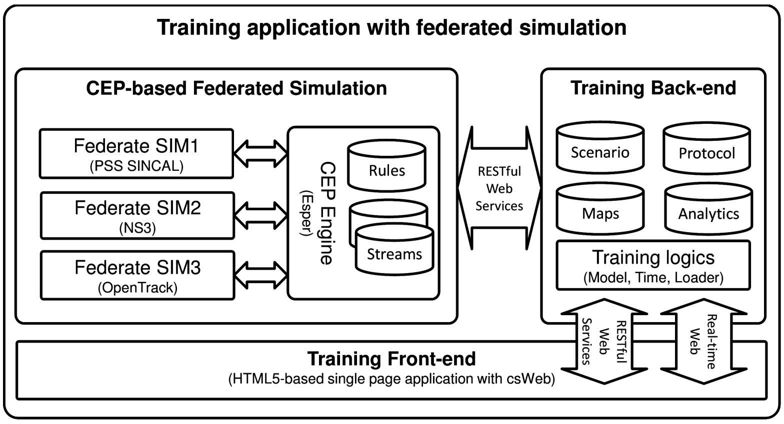


Figure 2: Integrating federated simulation into a training system using RESTful Web Services.

The federated simulation environment can be considered as a black box with certain services exposed to the outside world. Technically all the services exposed can be implemented as RESTful web services enabling loose coupling between heterogeneous IT systems. An overview of the integration is depicted in Figure **Errore. L'origine riferimento non è stata trovata.**. On the functional level, the major services exposed are:

- **Model Management** provides the functionalities to the training back-end to retrieve the simulation model, partially or completely, during system initialisation or for visualisation purposes. Furthermore it also provides write access to the service consumers to update the simulation model. For instance, the mitigation actions issued by crisis managers may influence the state of CI (like shut a substation down) and this needs to be flow perspective, this service provides a bi-directional read/write interface so that the external world can manipulate the simulation models hosted in the federated simulation environment.
- **Time Synchronisation** controls the simulation rhythm. As reflected in the simulation model in a timely way. To summarise, from the data mentioned above, simulators per se are not designed for federation. They need to be coordinated to perform a simulation of a larger system of systems. Time synchronisation is one of the key aspects that must be considered. The training back-end in a training system needs to consume this service to manage the overall simulation process like start, stop, pause, continue, step, and reset the federated simulation. Service consumers do not need to know the internal synchronisation mechanisms and the time synchronisation service provides a higher level of abstraction of the federated simulation as a single simulator to the external world.

Other training services, like providing feedbacks, scoring the outcome, and visually analysing the training results, etc. are also considered by this architecture. Because of the page limits however, they will be elaborated in other papers.

3. Scenario-driven Training

Scenarios play a central role in training applications, crisis management training in particular. Training applications should be able to handle multiple scenarios with minimum configuration efforts. In the proposed architecture of training applications, essential scenario elements discussed in a knowledge-driven approach [Xie et al., 2015] are considered. In this section, these scenario elements are briefly reviewed below followed by the discussion of dynamic loading of scenarios.

3.1. Elements of a scenario

A crisis scenario involving (inter-)dependent critical infrastructures contains a set of essential elements. They are the major cornerstones constituting a complete scenario. The following elements are provided as a brief summary of [Xie et al., 2015]:

- **CI Models** are fine-grained simulation models that are loaded into dedicated domain-specific simulators. The models can be retrieved and updated by other system components like the training application at runtime.
- **CI Dependencies** are the functional dependencies between different CI. For instance, the pumping facilities need electrical power to work during a flood. Modern pumps are even remotely connected with control centres, i.e. they also need communication services to operate properly. In the proposed architecture, theses dependencies can be modelled as declarative rules as illustrated in Section **Errore. L'origine riferimento non è stata trovata.**. With a comprehensive rule base, cascading effects of threats can also be addressed by the CEP engine.
- **Decision Makers** are specific roles in a crisis scenario, who are responsible to take mitigation actions like performing an evacuation. For the crisis management system, they are crisis managers.
- **Resources** are objects like rescue forces, police officers, hospitals, etc. They can be available or not during a crisis. Decision makers should have an overview of all of these resources.
- **Threat Models** are the computer models of threats like flood, cyber-attack, etc. Dedicated simulators can be used to provide fine-grained threat evolvement status.
- **Mitigation Actions** are the actions performed by decision makers to reduce or eliminate the negative effects and cascading effects of a threat. Typical mitigation actions are evacuating citizens of a region, send a group of rescue forces to another region, etc.
- **Storylines** are the potential developments of a scenario. It consists of a sequence of pre-defined events. Different storylines can be defined for a scenario.
- **Consequence Analysis** defines a set of criteria to quantify the consequences of damages to certain CI. The consequence analysis mainly focuses on the economic loss and impacts on human lives.
- **Context Information** refers to the information from outside world like the weather conditions, traffic situations, etc. in a crisis region. These are normally read-only information and can be retrieved at runtime via web services.

The proposed architecture should be able to handle scenarios with the essential elements mentioned above. On the functional level, modelling of these elements in scenarios is performed in a knowledge-driven fashion as described in [Xie et al., 2015]. Technically the scenario is structured as a tree as serialised in JSON [ECMA, 2015] in the proposed architecture as described in the following section.

3.2. Scenario Structure and Dynamic Loading

The crisis scenarios are recursively organised in a tree-like structure. The recommended technical implementation of such a tree-like structure is JSON [ECMA, 2015]. On the top-level, meta information of a scenario like `name`, `description`, and `location`, etc. as well as the essential elements like `CI models`, `decision makers`, etc. are listed. Detailed specifications of each essential element are provided on separate levels. To demonstrate the top-level structure, a simplified version of the JSON Schema is provided as follows:

```
{
  "type": "object",
  "properties": {
    "name": {"type": "string"},
    "description": {"type": "string"},
    "ci_models": {"type": "array", "items": {
       "$ref": "#/definitions/ci_model"
       }},
    "decision_makers": {"type": "array", "items": {
       "$ref": "#/definitions/decision_maker"
      }}, ...,
    "context_information": {"type": "object"},
```

```
    },
    "required": ["ci_model","decision_maker","resource",
      ..., "consequence_analysis","context_information"],
    "definitions": {
      "ci_model": {}, ...,
      "context_information": {},
    }
  }
```

During system initialisation the tree structure, which is materialised as a JSON scenario file, is loaded into the training system. The `scenario loader` is responsible to extract the simulation models from the scenario file and initialise the federated simulation environment. Furthermore, the threats model is also injected into the threat evolution engine for triggering the storyline. On the functional level, the threat evolution engine is also a domain-specific simulator. The initialisation process is depicted in Figure **Errore. L'origine riferimento non è stata trovata.**. The federation rules defined in the scenario file are pushed into the CEP engine during the system initialisation. After that, the training system is ready to be used for training purposes.

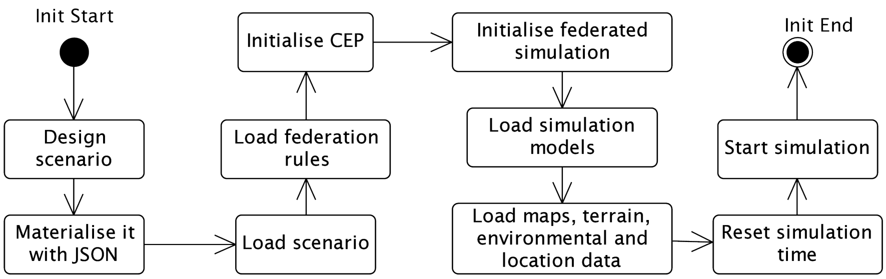


Figure 3: The initialisation process in the proposed architecture for developing collaborative training applications with federated simulation support.

4. Real-time Collaborative Training

Collaborative training in a timely fashion is an essential characteristic for modern crisis management training systems. In this section, the `csWeb` framework based on HTML5 is introduced, which is used as the technological base to build the frontend and part of the backend in the proposed architecture.

4.1. Introduction of csWeb

`csWeb`, short for CommonSense Web, is a localisable open source web-based GIS application[1] developed by TNO [2]. Common usage scenarios are city or country dashboards and Command Control (C2) applications that put a high emphasis on situational awareness. It provides casual users as well as business analysts and information managers with a powerful tool to perform spatial analysis. `csWeb` has a strong focus on usability and connectivity, be it connecting and sharing information with other users, or with services, simulations and models. Basic features are choosing your background map, filtering and styling your map layers. In addition, you can collaborate by sharing your layers with other users, or display sensor data in real-time.

`csWeb` is a single-page web application based on Google's framework Angular.js [3] written in Microsoft's superset of JavaScript, TypeScript[4], which compiles to regular JavaScript. Internally multiple libraries are integrated to provide various functionalities like spatial visualisation in 2D and 3D, real-time web communication, etc. For an overview of the `csWeb` architecture and the most important building blocks, see Figure 4. At the left, the two major packages are shown: `csComp`, which extends the client with GIS specific web components (i.e. Angular directives and services), and `csServerComp`, which does the same for the server. At the right, you see the example project, which is a thin layer on top of `csComp` and `csServerComp`. By separating the functional components from the actual application, we can very quickly create new applications, or update existing ones to use the latest version.

[1] https://github.com/TNOCS/csWeb
[2] http://www.tno.nl
[3] https://angularjs.org
[4] http://www.typescriptlang.org

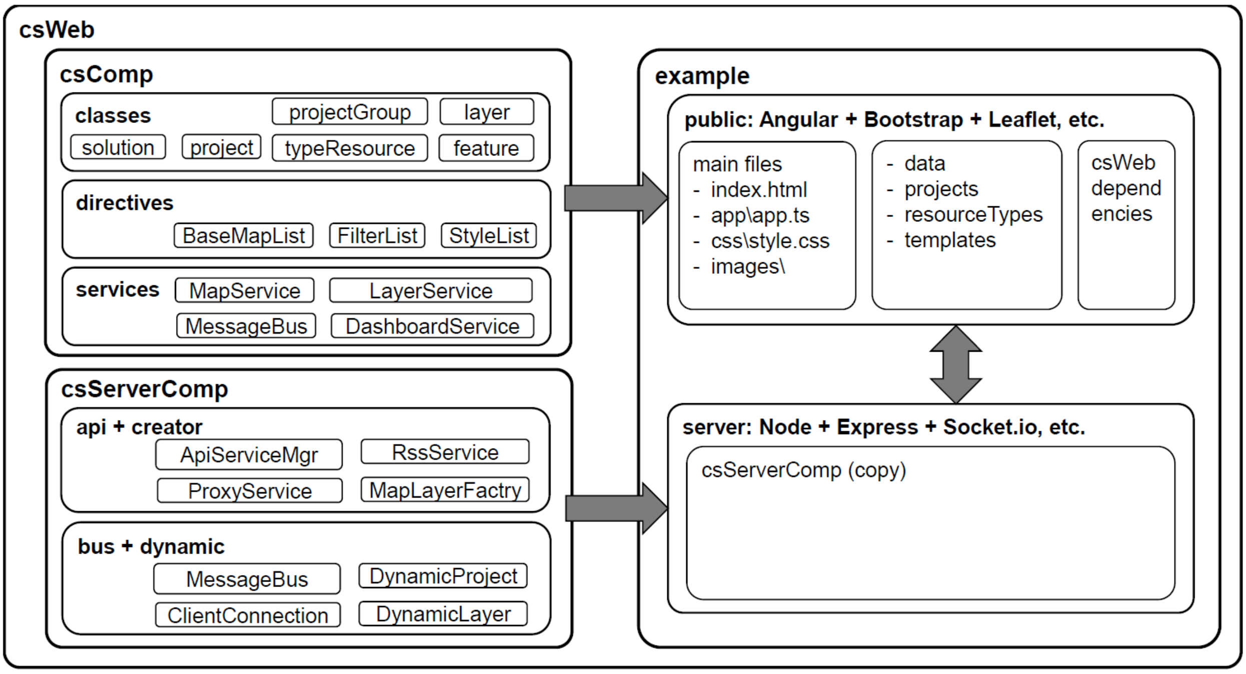


Figure 4: High level overview of the `csWeb` framework architecture, with its client (`csComp`) and server (`csServerComp`) components as used in a web application.

4.2. Support Real-time Collaboration with csWeb

As illustrated in Figure **Errore. L'origine riferimento non è stata trovata.**, the complete frontend in the proposed architecture is built on top of the `csWeb` framework introduced in Section **Errore. L'origine riferimento non è stata trovata.**. Besides, part of the server-side logics in the training back-end is also based on `csWeb` to support real-time collaboration without re-inventing the wheel again.

The training front-end is designed as a single page application, which is a direct derivation of the `csWeb` framework. Multiple training scenarios can be configured inside of the training environment. The users can choose one of them at runtime to start the training. The front-end for trainer and trainee are different. For the trainer, the functionalities focus on the trainee's behaviours, statistics, and aggregated views of all training participants. All trainees on the other side have the same view that shows the actions and their effects of other trainee in a timely fashion. The real-time communication components in `csWeb`, which are based on Socket.IO[5], provides a solid foundation for modern Web-based real-time communication. All actions performed by one trainee are transmitted to the training back-end and then further broadcasted to other trainees and the trainer in real-time. Socket.IO utilises modern standards like WebSocket if possible to provide reliable real-time communication in heterogeneous network environments. Moreover, on the back-end, any model change produced by the federated simulation environment are also reflected in the map database, which is further pushed, via the real-time component in `csWeb` again, forward to all the training participants in the current training session. The whole idea behind the scene is overcoming the heterogeneity in today's Internet environment to provide solid and reliable cross-border training experiences. The `csWeb` framework provides sufficient support to achieve this objective.

5. Related Work

Crisis management is an essential component in modern society. Plenty of work has been done during the last decades to understand the crisis management process and to improve the skills of different crisis management roles, the crisis manager in particular. A game-based crisis management training methodology was proposed in 2001 by the team of GAMMA-EC project [Stolk et al., 2001]. This approach focuses on game theory and its application in crisis management training. It tries to re-use the research done in the military domain and further apply the techniques for civil protections by integrating multimedia materials. Because of the technological limitations, real-time collaborative

5 http://socket.io

training and federated simulation are not the focus. A comprehensive discussion about the application of serious gaming on crisis management training is also provided in [Loreto et al., 2012]. However, no concrete system architecture is proposed. Apart from that, similar to the GAMMA-EC project, federated simulation and real-time collaboration have not been extensively addressed in this approach.

Similarly, ARGILE (Architecture for Representations, Games, Interactions, and Learning among Experts) proposed in [El Mawas and Cahier, 2013b, El Mawas and Cahier, 2013a] is focused on training medical first responders through serious gaming. The idea is that the trainers (domain experts) contribute to the co-conception of rules and certain objects of the game. The approach falls under interactive pedagogy, where the trainer watches players during the game and can add in real time new items or messages or sounds to complex the situation and to teach specific knowledge. However, this approach is not supported from simulation, nor it depicts critical infrastructures, just objects important for the specific scenario, without further formalisation. The SIMFOR project [Chambelland et al., 2011] aims to develop a prototype using another serious game oriented approach. Similar to the proposed architecture in this paper, it integrates several simulation models, though in a ad-hoc way, to provide fine-grained system dynamics of physical world. The focus of SIMFOR is however using artificial intelligence (AI), like multi-agent systems, to interact with humans to play the serious game. Furthermore, the technology used in SIMFOR is based on desktops applications, which makes it not well-suited for training with cross-border scenarios. IMACSIM [Benjamins and Rothkrantz, 2007] is an integrated simulation environment with multi-agents. Federated simulation is not integrated in IMACSIM to provide comprehensive (inter-)dependent system dynamics of CI. Another agent-based approach is used in RimSim Response[Campbell and Weaver, 2011], which uses computer-based role-play to improve emergency responder situation awareness for hospital evacuation scenario. To create the scenario, the user uses a coordinated map editor tool, where he can draw responsibility jurisdictions and the regions affected and assign emergency response resources [Campbell et al., 2010]. The visual interface provided enables an analyst to review communications and actions made by the trainees in response to the crisis, identify bottlenecks or delays that occurred and see how resources were moved during the time period visualised.

In the proposed architecture, the HTML5-based real-time collaborative framework `csWeb` is adopted to provide comprehensive support in real-time web, spatial visualisation, collaborative operations, etc. To our knowledge, no similar open source web applications exist: For example, Mapbox[6] allows you to create beautiful maps too, but is hosted and is not easy to extend the server side. Geomajas[7], an open source JavaScript mapping API for cool maps and apps, offers both server and client side, but it lacks C2 or collaborative functionality. ESRI also offers a solution [8], but its main use case is to connect it to the commercial ArcGIS server. Finally, Mapfish [9] is a flexible and complete framework for building rich web-mapping applications. It emphasizes high productivity, and high-quality development, but also lacks a server component and C2 and collaboration functionality.

6. Conclusion

Crisis management plays a central role in modern society. Various facilities like energy supply systems, telecommunication networks, transport systems, etc. are becoming more and more (inter-)dependent with each other. Cascading effects need to be analysed beforehand to avoid serious damages in case of emergencies and crisis.

Training provides a promising way to improve the know-how and skills of crisis management staffs, crisis manager in particular. With the advance of modern technology - sophisticated computer simulations and real-time web - provides a solid foundation for developing modern crisis management training systems. A system architecture is proposed in this paper for that purpose. It is based on federated simulation and real-time web offering sufficient flexibility, functionality and efficiency to facilitate the implementation of modern training systems. One of the primary advantages of the proposed architecture is its agnostics to training scenarios and hazards. With minimum effort, new scenarios and hazards can be developed and integrated into the training system developed according to the proposed architecture. Furthermore, as a side effect, this architecture can also be modified to support hot-phase decision making as long as the simulation system is fast enough. On the other side, the whole architectural design is restricted by the capability of simulators for fast rollback. During training, crisis managers usually try different courses of actions. For each course, the federated simulation environment must be able to roll back to a previous state in a timely fashion. This is however not well supported by all simulators. The long-running rollback operation can greatly influence the usability of systems developed with this architecture. This is one of the major limitations of the proposed architecture.

---

6 https://www.mapbox.com
7 http://www.geomajas.org
8 https://developers.arcgis.com/javascript
9 http://www.mapfish.org

In the future, the architecture will be extended to support multi-threats scenarios - more than one threat with combined impact to CI and other assets. Besides, uncertainty and probability of happenings is still a weak point and not well addressed in the proposed architecture. For instance, the declarative rules in the CEP engine are not able to describe uncertainty per se. More semantically rich structures can be developed to enhance this point. Finally, federated simulations are intrinsically distributed. With the proposed architecture, different simulators are federated with standard web protocols like HTTPS. On the semantic level however, it is ambiguous and cannot interoperate with each other automatically. Extension with Semantic Web technologies can be a promising candidate to further promote the architecture to the next level of training system design.

7. Acknowledgements

This research leading to these results was funded by the European Commission within the Seventh Framework Programme project CIPRNet and PREDICT. The authors thank all of the project partners for many constructive discussions and useful suggestions.

REFERENCES

[Adi and Etzion, 2004] Adi, A. and Etzion, O. (2004). Amit-the situation manager. *The VLDB Journalâ€"The International Journal on Very Large Data Bases*, 13(2):177–203.

[Benjamins and Rothkrantz, 2007] Benjamins, T. and Rothkrantz, L. (2007). Interactive simulation in crisis management. In *Proceedings of the 4th International Conference on Information Systems for Crisis Response and Management ISCRAM2007*, pages 571–580.

[Campbell and Weaver, 2011] Campbell, B. and Weaver, C. (2011). Rimsim response hospital evacuation: Improving situation awareness and insight through serious games play and analysis. *IJISCRAM*, 3(3):1–15.

[Campbell et al., 2010] Campbell, B. D., Schroder, K. E., and Weaver, C. E. (2010). *RimSim Visualization: An Interactive Tool for Post-event Sense Making of a First Response Effort*. ISCRAM.

[Chambelland et al., 2011] Chambelland, J.-C., Raffin, R., Desbenoit, B., and Gesquière, G. (2011). Simfor: towards a collaborative software platform for urban crisis management. *The IADIS Computer Graphics, Visualization, Computer Vision and Image Processing (CGVCVIPâ€™2011), Rome, Italy*.

[DIESIS, 2008]DIESIS (2008). The EU DIESIS Research Project.

[ECMA, 2015] ECMA (2015). JSON – JavaScript Object Notation.

[El Mawas and Cahier, 2013a] El Mawas, N. and Cahier, J.-P. (2013a). Co-designing a serious game to train emergency medical services. In *Collaboration Technologies and Systems (CTS), 2013 International Conference on*, pages 588–593. IEEE.

[El Mawas and Cahier, 2013b] El Mawas, N. and Cahier, J.-P. (2013b). Towards a knowledge-intensive serious game for training emergency medical services. In *Proceedings of the 10th International Conference on Information Systems for Crisis Response and Management (ISCRAM), Baden-Baden (Germany)*.

[EsperTech, 2012] EsperTech (2012). Esper - The Complex Event Processing Engine.

[IEEE, 2010] IEEE (2010). IEEE 1516-2010 - Standard for Modeling and Simulation (M&S) High Level Architecture (HLA) – Framework and Rules. Technical report, IEEE.

[IRRIIS, 2006] IRRIIS (2006). The EU IRRIIS Research Project.

[Kuhl et al., 2000] Kuhl, F., Dahmann, J., and Weatherly, R. (2000). *Creating computer simulation systems: an introduction to the high level architecture*. Prentice Hall PTR Upper Saddle River.

[Loreto et al., 2012] Loreto, I. D., Mora, S., and Divitini, M. (2012). Collaborative serious games for crisis management: an overview. In *Enabling Technologies: Infrastructure for Collaborative Enterprises (WETICE)*, pages 352–357. IEEE.

[Luckham, 2001] Luckham, D. C. (2001). *The Power of Events: An Introduction to Complex Event Processing in Distributed Enterprise Systems*. Addison-Wesley Longman Publishing Co., Inc., Boston, MA, USA.

[Nash and Huerlimann, 2004] Nash, A. and Huerlimann, D. (2004). Railroad simulation using OpenTrack. *Computers in railways IX*, pages 45–54.

[Nieuwenhuijs et al., 2008] Nieuwenhuijs, A., Luiijf, E., and Klaver, M. (2008). Modeling dependencies in critical infrastructures. In *Critical Infrastructure Protection II*, pages 205–213. Springer. four CI states.

[NS-3 Consortium, 2015] NS-3 Consortium (2015). NS-3 Network Simulator.

[Robert and Lajtha, 2002] Robert, B. and Lajtha, C. (2002). A new approach to crisis management. *Journal of contingencies and crisis management*, 10(4):181–191. crisis management training happens every one or two years.

[Rome et al., 2014] Rome, E., Langeslag, P., and Usov, A. (2014). Federated modelling and simulation for critical infrastructure protection. In *Networks of Networks: The Last Frontier of Complexity*, pages 225–253. Springer.

[SimTec, 2015] SimTec (2015). PSS SINCAL.

[Stolk et al., 2001] Stolk, D., Alexandrian, D., Gros, B., and Paggio, R. (2001). Gaming and multimedia applications for environmental crisis management training. *Computers in Human Behavior*, 17(5â€“6):627 – 642.

[Xie et al., 2015] Xie, J., Theocharidou, M., Barbarin, Y., and Rome, E. (2015). Knowledge-driven scenario development for critical infrastructure protection. In *Critical Information Infrastructures Security - 10th International Workshop, CRITIS 2015, Berlin, Germany, October 5-7, 2015*, Lecture Notes in Computer Science, page to appear. Springer.