

Chapter 10

The Use of What-If Analysis to Improve the Management of Crisis Situations

Erich Rome, Thomas Doll, Stefan Rilling, Betim Sojeva, Norman Voß and Jingquan Xie

Abstract The EU FP7 Network of Excellence CIPRNet has developed CIPRTrainer, an application that provides a new capability for training crisis management (CM) staff. It enables exploring different courses of action and comparing their consequences (what-if analysis) in complex simulated crisis and emergency scenarios. The simulation employs threat, impact, and damage models and is based on federated modelling, simulation and analysis of Critical Infrastructures. In this chapter, we present an overview of the technical realisation of CIPRTrainer, embed the approach into the state of the art, and elaborate on CIPRTrainer’s user interface and the training experience. The chapter also explains how the models for the complex crisis scenarios have been created, what level of detail could be realised, and how cases of missing data could be handled. As an example, we use a cross-border scenario about a cargo train derailment disaster. In the final sections, the reader learns how to set up, start and perform a training session with CIPRTrainer, how to use ‘what if’ analysis, and how to read the results of consequence analysis.

E. Rome (✉) · T. Doll · S. Rilling · B. Sojeva · N. Voß · J. Xie
Fraunhofer IAIS, Sankt Augustin, Germany
e-mail: erich.rome@iais.fraunhofer.de

T. Doll
e-mail: thomas.doll@iais.fraunhofer.de

S. Rilling
e-mail: stefan.rilling@iais.fraunhofer.de

B. Sojeva
e-mail: betim.sojeva@iais.fraunhofer.de

N. Voß
e-mail: norman.voss@iais.fraunhofer.de

J. Xie
e-mail: jingquan.xie@iais.fraunhofer.de

List of Abbreviations

API	Application programmer interface
CA	Consequence analysis
CAM	Consequence analysis module
CEP	Complex event processing
CGE	Calculable general equilibrium
CI	Critical infrastructure
CM	Crisis management
DB	Deutsche Bahn (German railway operator)
DBMS	Database management systems
DE	Two-letter country code for Germany
ECI	European critical infrastructure
ENTSO-E	European network of transmission system operators energy
ESDB	European scenario database
fMS&A	Federated modelling, simulation and analysis
GDP	Gross domestic product
GUI	Graphical user interface
HTTP	Hypertext transfer protocol
ICE	InterCity express (German high-speed train)
IOM	Input-output model
NL	Two-letter country code for The Netherlands
NRW	North-Rhine Westphalia (federal state in Germany)
OLAP	Online analytical processing
OSM	OpenStreetMap
REST	Representational state transfer
ROOP	Resource oriented operation planning
SDL	Scenario description language
WIA	What-if analysis

1 Introduction—Role of Critical Infrastructures in Civil Crisis and Disaster Situations

The management of a disaster or crisis typically consists of cycles of situation update, analysis of the situation, decision taking, and planning and execution of response actions, sometimes under severe time pressure. At decision points, crisis managers often do not have just one option for action, but several. The challenge is to take a well-informed and most effective decision. Insufficient awareness of the role of Critical Infrastructures (CI) [1] and incomplete information on consequences of crisis or disaster evolution [2] contribute to that challenge. CI can play three main roles in crises or disasters. (1) The CI may be *affected* by a disaster. For instance, an extended flooding would most likely disable elements of the electricity, telecommunication, and sewer system infrastructures (and maybe more), with cascading effects on other

CI [3]. (2) An emergency or disaster may *emerge* from a CI. A technical failure in the electricity infrastructure may lead to a blackout and further cascading effects [4]. (3) An infrastructure may be a *resource* for response and mitigation actions. This might not be immediately obvious, but may come as a late insight when this infrastructure fails or even gets destroyed [5]. An example is a bridge in eastern Germany that was washed away by a fluvial flood. The local responders were no longer able to send forces to the other riverside, and did not have an alternative plan for that situation.

In most cases, it is not possible for crisis managers to revert a decision or an action already taken—in reality. However, in *simulation* it is possible to do exactly this: ‘go back in time’ and explore a different course of action. This allows answering hypothetical questions like ‘What would happen if I take a different decision or follow a different course of action?’. Therefore, this is also sometimes called ‘what-if analysis’. Since this type of what-if analysis requires simulation, it is rather suited for training purposes. Providing such what-if analysis as a capability to end-users is the essential idea behind the training system CIPRTrainer, which we will present in detail in the main part of this chapter.

CIPRTrainer is the software system that enables crisis managers to train decision-making in crises and emergencies involving cascading effects of CIs. CIPRTrainer constitutes an unprecedented training opportunity that complements standard command post, table-top, or physical exercises. The expected benefits would be increased awareness of crisis managers of the role and behaviour of interconnected CIs in disasters, emergencies, and crisis situations, and a better understanding of possible consequences of scenario evolution and the influence of own actions.

The remainder of this chapter is structured as follows. We continue with briefly embedding the CIPRTrainer approach to what-if analysis into the state of the art and then proceed with characterising the types of complex crisis scenarios that we designed for the CIPRTrainer prototype. Then we will give an overview of the building blocks of CIPRTrainer, explaining how we technically realised it. The following section will then provide an overview of how we realised impact and consequence analysis (CA) for the global assessment of damages and how it is employed for what-if analysis. We continue with explaining how CIPRTrainer is actually used and with an example of a training session. An outlook on the next version of CIPRTrainer and a conclusion end this chapter. For reference, we included a list of acronyms and a bibliography.

2 State of the Art: Critical Review of Literature on What-If Analysis and Federated Modelling and Simulation

What-if analysis, as a method for hypothetic data analysis, has been extensively investigated in the area of predictive business intelligence [6, 7], data warehouse [8], and in-database modelling and simulation [9, 10]. A what-if model is proposed

in [10]. It is argued in this work that the data is dead without using the what-if models to analyse them and discover insightful information from the data. Finally it drew the conclusion by pointing out that modern DBMS has certain degree of analytic support, however deep predictive analytics beyond the commonly used statistical methods are still missing. In the area of data warehouses, methodologies for what-if analysis have been proposed [8]. This methodology provides a systematic approach to design systems with what-if analysis support. A case study has also been provided to illustrate the practicality of the methodology. A dedicated online analytical processing (OLAP) query types—what-if query—was proposed in [7]. It aims to bring what-if functionalities into OLAP applications by providing a high-level syntactic structure to ease query construction. Tailored index structures have also been proposed to accelerate the query processing.

Methodologies with stochastic analysis supporting certain degrees of what-if analysis are provided in SimSQL [11]. SimSQL however focuses on the analysis of data with possible worlds in a stochastic way, which differentiates the application use cases for simulation-based decision support—as described in our approach. Nevertheless some ideas like using the possible worlds to represent and perform the simulation to gain and compare different insights of certain actions are similar. Probabilistic databases [12] provide a set of methods to handle imprecise and uncertain data with the concept of possible worlds. These systems provide built-in support for hypothetical queries, a.k.a. what-if queries to retrieve the data from different possible worlds. One example is the MayBMS [13], which is a state-of-the-art probabilistic database management system for scalable what-if queries. These works are more focusing on the efficiency of query processing in the database systems based on the probability of tuples stored in the database tables.

For modelling and simulating interconnected systems of heterogeneous CI, there are basically two approaches, namely *integrated* and *federated* simulation. In the integrated approach to modelling and simulating CI, the elements of the interconnected different CI are modelled using a single representation scheme. There is only one simulator that simulates the entire modelled CI system-of-systems. This approach is rather suited for models with a high degree of abstraction, that is, less model detail, in order to be efficiently manageable. On the positive side, the modelling and simulation is in one hand, but the designers of the models should consult domain experts for ensuring technically valid models.

An alternative approach is *federated modelling and simulation*. Here, for each considered CI a specialised domain simulator is employed. Several such simulators are then interconnected by means of some type of communication software (middleware). The whole setup is then called a federated simulation, and the component simulators are called federates. This second approach is the one that we have chosen for CIPRTrainer. An advantage is that the domain simulators are specialised on their domain and provide a correct simulation. Sometimes, it is even possible to acquire a ready-made model and just read it in as a data file in a simulator. Another advantage is that such specialised CI simulators allow for a fair level of detail and thus provide better scalability than integrated simulations. A drawback is that

typically all the models or data formats of the federates are different and require familiarisation and domain expertise.

Practically all federated CI simulations are results of research projects. Some of them have been used and are being further developed in agencies or national or EU labs. Rome et al. [14] have provided an elaborate state of the art chapter on federated modelling and simulation. They write:

The characterised works [...] can be divided roughly into three—not entirely disjunct—categories:

1. Special purpose federated simulation systems, consisting of a number of simulators (CI and others), additional system components, and a dedicated middleware for communication and synchronisation (IRRIIS, EPOCHS, ...),
2. Frameworks for modelling, simulation and analysis of CI using dedicated—for instance, agent-based—simulations (I2Sim, AIMS, IME, ...),
3. More general frameworks for setting up distributed federations and more general middleware for communication and synchronisation within federations (IDSim, ASimJava, ...), including (quasi-)standards (OpenMI, HLA, ...), and sometimes accompanied by proofs-of-concept (DIESIS, XMSF, WSIM, ...).

We would recommend the reader to resort to [14] for an in-depth review of state-of-the-art federated modelling and simulation frameworks and systems.

3 What-If Analysis—A New Capability for Training Crisis Management Staff

The what-if analysis capability of CIPRTrainer enables trainees to explore different courses of Crisis Management (CM) actions in a computer-based simulation (Fig. 1). CIPRTrainer displays information on events that happen in the simulation, like a derailment of a cargo train. The system has an inventory of actions available

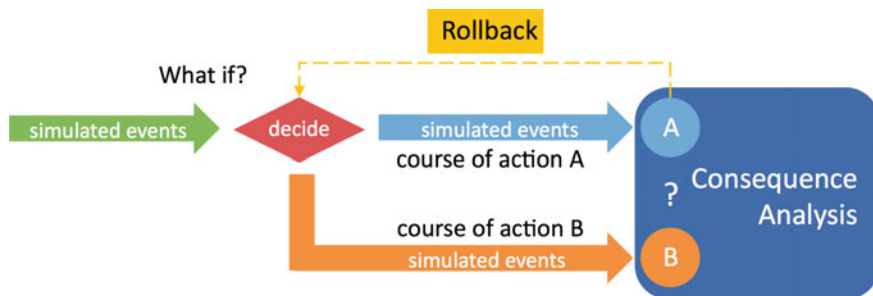


Fig. 1 What-if analysis: after taking course of action A, the trainee may perform a rollback to a decision point, and take a different course of action B. The trainee can use consequence analysis to compare the overall consequences of both scenario evolutions

for reacting on the occurring events. Rules within CIPRTrainer provide some additional flexibility. For instance, if a certain response action is being performed by the trainee within a given time window, then it would prevent some disastrous event from happening.

At any time after the simulation started, the trainee may choose to ‘go back in time’—or, as we call it, perform a *rollback*—and explore a different course of action. In order to do this, the trainee must select one of the previously performed actions, and then perform the rollback. CIPRTrainer then resets the simulation into the state that it had before the selected past action. By following a different course of action, the trainee creates another version of the simulated ‘world’.

Such rollbacks can be performed multiple times. Since the history of all performed actions is recorded, the generated courses of actions form a tree-like structure. CIPRTrainer can display this structure for providing an overview of the training activities.

A core element of the training is evaluating the training session and the performed courses of action. The trainee shall be enabled to find out how the chosen courses of action influenced the overall outcome or consequences of the simulated crisis or disaster. For doing this, the tree-like visual representation of the courses of action serves as starting point for performing CA.

CIPRTrainer contains a Consequence Analysis Module (CAM), which enables the user to understand the consequences (in terms of harm to humans, degradation of CI services and monetary losses) of the simulated impacts and of the chosen actions (or inactions). The CAM utilises data from the CIPRTrainer database, and an array of methods implemented for calculating the consequences for the population, and the critical and non-CI in the affected region.

4 Scenarios for Training

One design goal of CIPRTrainer was a wide applicability of the system, including crisis situations with cross-border effects. We picked a region spanning both sides of the border of two countries represented in the CIPRNet consortium: Germany and The Netherlands. The geographical location is restricted to the Kleve district in Germany and the city region of Arnhem-Nijmegen in the Netherlands. The area is prone to flooding by high water levels of the river Rhine. Also, it contains a number of infrastructures, like the railway line connecting Rotterdam harbour with the European hinterland. In this setting we designed two storylines in a complex scenario with cross-border effects [15]. One is the derailment of a cargo train in the German city of Emmerich, and the second one is an extended flooding of the area by the river Rhine.

For the development of the scenarios, we started with research on information and data from the considered regions. Data are the basis for modelling the scenario on the computer. Some of the modelled CI networks are fictive for two reasons:

first, we did not have data on some of these networks and second, for security reasons, since we did not want to disclose sensitive information. We employed the domain expertise of the consortium, including electrical and telecommunications engineers, security professionals, and experts in railway security, cyber security, crisis management, and the water domain. External expertise was provided by the head of the fire-fighters in a large German city, and experts from CIPRNet's international advisory board. Later in this chapter, we will describe in more detail two specific aspects of modelling: (1) the modelling of networks of interconnected CI and (2) modelling for CA. More technical details of the modelling activities can be found in CIPRNet deliverables D6.2 [16], D6.3 [17] and D6.4 [18].

5 CIPRTrainer

The CIPRTrainer system consists of *software* and *data*. The software part, the application or computer programme called 'CIPRTrainer', can be considered the machinery that performs the simulation. The data part, stored in CIPRTrainer's database, consists of the computer models of the crisis or disaster scenarios, that is, artificial 'worlds' based on data and information of real geographical locations and hypothetical dangerous incidents. Understanding the new what-if analysis capability requires a basic understanding of scope and limitations of both parts. Therefore, this section will address both the scenario models and the CIPRTrainer system.

5.1 System Description

CIPRTrainer is a software system that provides training services to crisis managers for decision-making in crisis situations. Its strength is the ability to simulate complex crisis scenarios including cascading effects of CI disruptions. It is designed with flexibility in mind. Federated simulation is adopted to enhance the training by providing realistic system dynamics. Geo-spatial information is integrated seamlessly into this system to enhance the location-aware situational awareness. Complex Event Processing (CEP) provides a declarative means to glue the dynamics of different components. Finally, all of the system components are technically integrated with the lightweight RESTful Web Services. From the functional perspective, CIPRTrainer consists of two major building blocks, a **design engine** and a **training engine**, which will be elaborated in the following sub-sections. An overview of all the building blocks and the data flow between them is depicted in Fig. 2.

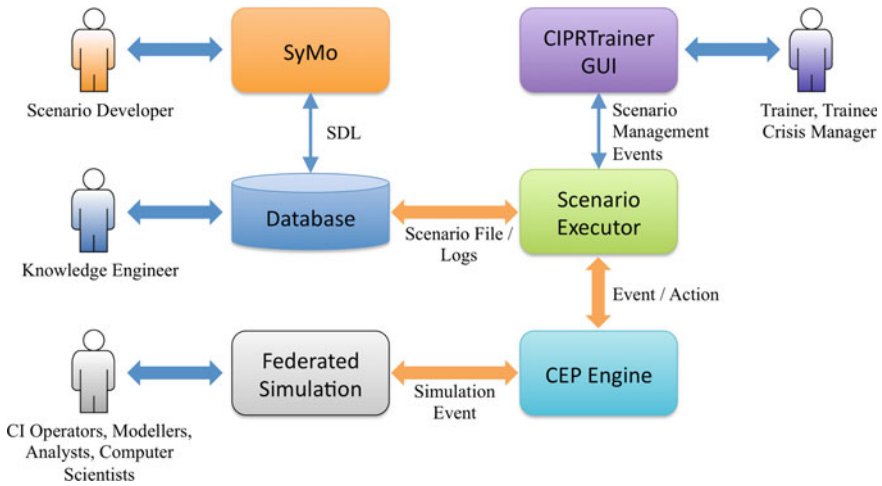


Fig. 2 Building blocks of CIPRTrainer. The CIPRTrainer graphical user interface is the part with which trainer and trainees interact. ‘SyMo’ refers to the scenario editor, the tool with which the static modelling activities are started. The other four components constitute CIPRTrainer’s ‘backend’, that is, the internal simulation machinery. The heterogeneous modelling activities for setting up a scenario precede the regular usage for training. A knowledge engineer adds model parts to the database, and domain experts provide CI models to the federated CI simulation

5.1.1 Design Engine

For the CIPRNet project SyMo (System Modeller) is used as a scenario editor. SyMo is a tool developed by Fraunhofer since 2008 and it is used in various projects for modelling and analysis purposes. The main advantage in using SyMo for the creation of the scenarios is that all necessary elements, tools for concatenation, sequence control, syntax checks and even semantic examination are already implemented and incorporated inside a graphical user interface. Scenario models in SyMo are two-part and consist of a static model and the dependencies between elements of the static model. Typically, the tree-like static model (Fig. 3) may contain components like an organisational structure, a taxonomy, technical systems, events, resources etc. The model representation generated with SyMo contains some variables and parameters, which allow creating different storylines within the scenario.

Modelling with SyMo consists of three steps:

1. Create a static model and a process model of the scenario
2. Configure the model by choosing concrete values for variables and parameters
3. Export the configured SyMo model into a scenario file and store it in the CIPRTrainer scenario database.

The modelling of the scenario storylines for CIPRNet follows an approach that is called resource oriented operation planning (ROOP). The basic idea is that

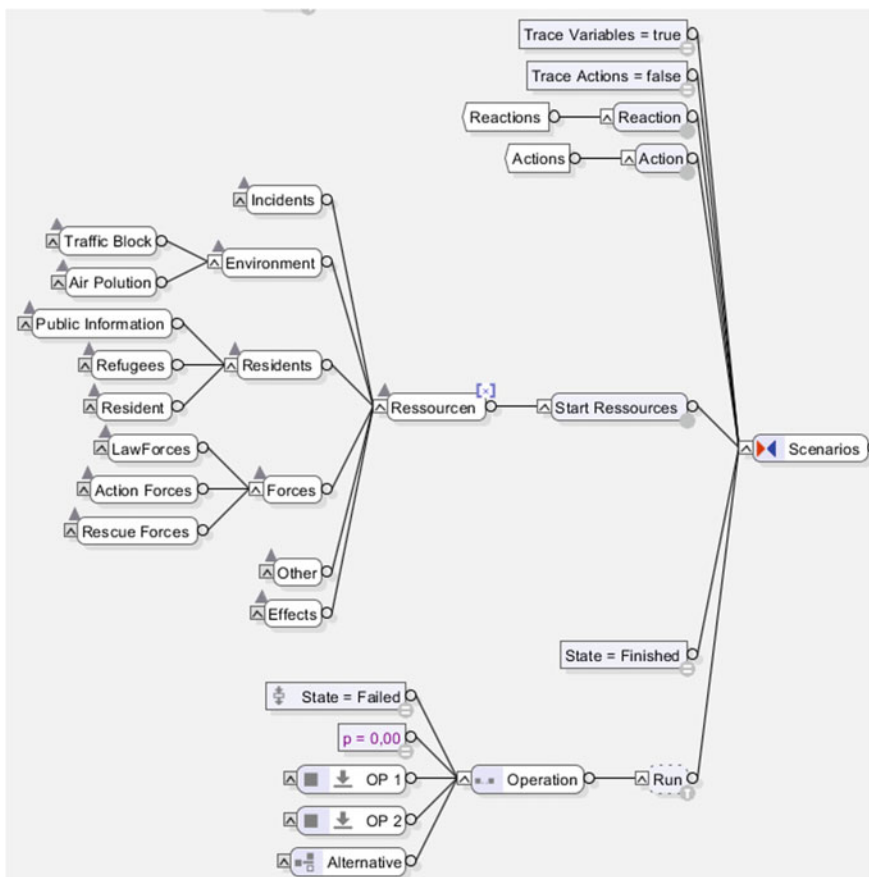


Fig. 3 Snapshot of a scenario model created with the Scenario Editor based on SyMo

counteracting a disaster is a matter of available resources, remaining time, and situation of the disaster (like location and effects/impacts of incidents). The available resources at any given point in time limit the possibilities of response and mitigation actions. The situation of the disaster determines what would need to be done to counteract (or fight) it. Thus it is important to keep track of the used resources and the evolution of the disaster. Responders and action forces are considered and modelled as resources. In a uniform way, the attributes of the situation of the disaster are also modelled as “resources”. This is a legacy from using SyMo in the military context. The disaster could be considered a “foe” and the responders as “friend”. Both have resources and “use” them to “fight” each other.

For modelling the disaster incidents and the disaster management and response actions in the affected area, it is important to also know the locations of the resources. In order to facilitate the modelling in this respect, we start with partitioning the area where the disaster happens into zones. Using zones allows a simpler and quicker

processing of geographical interactions. The zone borders are manually defined and oriented along landmarks such as rivers, main streets, railway tracks, historic city centres etc. Having done this, we just need to know in which zone which action shall be applied or response forces are located and their strengths. We do not need to know the exact positions of, for instance, each fire-fighter at any given point in time. The Emmerich scenario area is arbitrarily divided into 15 zones.

The top-level model of the “scenario components” consists of resources, locations, effects and two technical elements, namely measurement units and a resource generator. The top-level model of the “incident related aspects” consists of reactions, actions, action patterns, parameters and a technical component named “scenarios”. After defining the involved scenario components as attributes and variables, the interaction of the components is modelled. Different operators are applied to model the incidents, reactions and actions performed in a given time span. There are seven operators for different tasks:

- Sequence
- Parallel
- Race
- Action
- Alternative
- Iterator
- Call

The sequence operator executes the subsequent tasks sequentially. The parallel operator performs the tasks all at once. The race operator will execute the tasks in parallel and only evaluate the task that is finished first. The action operator simply executes the given task. The alternative operator leaves a second choice for the case that the first task cannot be successfully performed. The iterator operator is used for defining tasks that are then executed repeatedly in a loop. The call operator behaves like the action operator. The only difference is that a sub function is called in contrast to executing an action directly. With these operators it is possible to model the incident related aspects of the scenario. For instance, the actions are modelled as a sequence of operations with alternatives for deploying the forces and parallel operations for moving the different forces from different locations.

After modelling the scenario details within SyMo and configuring and creating the scenario file, the resulting file can be parsed and serialised into a flat file conforming the Scenario Description Language (SDL) [15]. The event-processing engine for initialisation of the start resources can then read the different operators, variables and timestamps and in addition events and actions are read and written into the event queue. The events and actions are ordered using the given timestamps from the SyMo model. Execution of all events and actions with timestamp = 1 can then be executed by starting the event processor. The given events are then processed using special rules. These rules decide which events should be forwarded to the simulators and what actions the simulators have to perform as reaction.

5.1.2 Training Engine

The training engine of CIPRTrainer is a modern web-based application, which accepts the scenario description files from the design engine. It basically contains two parts: the front-end GUI and the backend machinery.

The front-end GUI is a standard web application, which uses modern web technologies [19]. It is accessible through the regular HTTP/HTTPS protocol. The front-end is implemented using a variant of the classic MVC (Model-View-Controller) framework; see the right part of Fig. 4. The content is loaded dynamically by sending asynchronous requests and receiving push notifications from the application server.¹ The system embraces a three-tier-architecture, which contains a presentation, logic and data-tier (see Fig. 4 the left). The front-end (presentation-tier) is implemented using the AngularJS framework. It provides models that are bound to the view-layer. These models can be manipulated through its controller-functions. Its service-functions handle typically the communication to the services. The web or application-server (logic-tier) incorporates an event-driven runtime environment. It incorporates the application- and business logic, and provides a RESTful Web Services for what-if analysis and other capabilities. The application server also includes scenario services, and the federated simulation controller that is able to set up, start and stop the federated simulation. The web-server has access to the databases, which serialize spatial- and socio-demographic data, CI models, user configurations and training protocols. In general, the front-end GUI contains the following functional blocks:

- **System authentication.** Each training session starts with an authentication. Using the application require user authentication: users have to launch the application by opening the browser and entering the domain name on which the CIPRTrainer web-server is listening. The landing page offers a navigation-bar on which the user is able to log into the system entering username and password. Based on the user role, he or she may enter the training mode or the trainer dashboard.
- **Trainee view.** CIs or resources are represented as GIS markers containing CI- or resource-specific icons. Icons are carefully chosen in order to avoid misinterpretations. Crisis managers use specific tactical symbols that represent events and current states on the map. Moreover, for a crisis manager it is important to immediately know the operational status of CIs.
- **Trainer dashboard.** The trainer is able to log into the dashboard that monitors the evolution of the running training session including information about the trainee, the trainee's actions, scenario state, and CA results. The user also has the possibility to choose, start and stop scenario, and assign a participant to a training session. Moreover, the computed CA and training protocols are downloadable in CSV or JSON format.

¹<https://nodejs.org>.

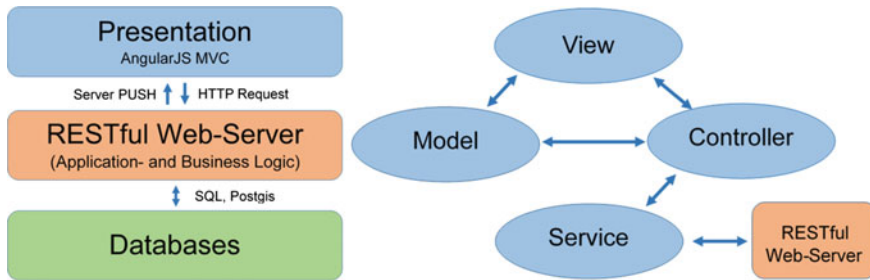


Fig. 4 System architecture of the CIPRTrainer (*left*) and AngularJS MVC pattern (*right*)

- **Timeline.** The timeline displays a set of different events in a chronological order. These events can be pre-defined scenario events (accident, explosion, etc.) or events that are calculated by the federated simulators. Also, it can display different kind of actions, which can be performed by the crisis manager. Lastly, external sources of information are displayed on the timeline, which can origin from other agencies such as police or fire-fighters. A crucial advantage of displaying a chronological set of events is that the crisis manager can keep track of all kind of information sources. The user is able to focus on specific time intervals and thereby focus on important events and hide less important ones by dragging and zooming onto the timeline. Therefore, the user is able to comprehend the complete scenario and thus make better decisions.
- **Internationalisation support.** The CIPRTrainer supports various languages (currently Dutch, German, and English). The user can choose a desired language by clicking on the listed flag on the navigation bar. The CIPRTrainer is able to depict tactical symbols of resources like police, fire-fighters or hospitals based on the end-user's localisation. Crisis managers from the Netherlands utilise different tactical symbols than German crisis managers. The CIPRTrainer includes a set of tactical symbols for each country. Currently, German and Dutch tactical symbols are incorporated into the CIPRTrainer.
- **Action execution.** Actions influence the state of the critical infrastructure and the result of the consequence analysis. The trainee has two types of actions: (1) First responder actions; (2) Crisis management actions. The first type of action involves actual forces/resources, which are spread in the region of Emmerich. The capacities are presented as triple (leader, sub-leader and forces). The accumulation of these three values refers to the capacity strength of a specific unit. The user is able to send resources with a certain capacity to the crisis region. The second type of actions is suited for crisis managers. A crisis manager is able to alarm public authorities and the general public as well as evacuate critical regions. Each action influences the consequences in the scenario.
- **Action tree.** The CIPRTrainer allows the end-user to select and compare consequences of different courses of action. Performed actions and their

chronological order are visualized as a multidimensional tree. Each node represents a performed action. Whenever the user jumps back to a prior action x_t (perform a rollback), the CIPRTrainer automatically creates a new branch on action x_t , on which upcoming actions will be added. The result is a multidimensional tree that reflects the trainee's decisions throughout the entire training session. The leaves of the n -dimensional tree are the last performed actions of which each the user is able to acquire the consequence analysis of the entire action chain back to the initial state node of the tree.

The backend of CIPRTrainer contains basically two parts: (1) the **business logic layer** that reside in the application servers and the rule base of the CEP engine; (2) the **persistence layer** where all relevant data is stored and managed in a single instance of PostgreSQL database.

The business logic of CIPRTrainer backend is mainly developed as a NodeJS application. User sends requests to the web server, which then redirects the request to a private IP address on which the application server listens. Typically, a web application consists of various configuration files (server and database configuration, etc.), a front-end implementation, set of views and routes, and a logic tier. The main configuration of the server incorporates an HTTP web-server definitions and references to the RESTful endpoints. Any other sort of configurations that do not deal with the application logics, such as database connections and web mapping configurations, are separated in other configurations. The server implementation including route end-points, views, and server specific services are located in the server folder.

Scenarios allow crisis managers to outline a sequence of events and provide the basis for the performing the CA, thus evaluating susceptibilities of CIs by revealing dependencies, interdependencies and cascading effects (see [20–22]). Part of the scenario is the storyline, a set of events that could happen during the scenario running. Scenario executor controls the heartbeat of the whole system. It maps the simulation time and real wall time. For instance, to accelerate the simulation, the scale can be 60:1, i.e. 60 simulation seconds should be done within one real time second. Under this setting, two situations can happen:

- The system is fast enough and the actual execution time is less than one real time second. Therefore some kind of `sleep` mechanisms will be introduced before the simulation for the next 60 simulation seconds is started.
- The system is not fast enough to finish the simulation within the given time frame. That means, it is not possible to simulate 60 s within one real time second. The scaling factor will be modified based on the best system performance, e.g. 10:1—just simulate 10 simulation seconds instead of 60.

The trainer can initialise (load the storyline of the scenario) and start or stop the scenario executor. Each event in the storyline is annotated with a timestamp t_i . Once the simulation time t_s passes t_i , the scenario executor notifies CEP-Engine and the CIPRTrainer by sending a HTTP push-request containing event-specific data (see Fig. 5). This way, the trainee can see events on the map including GPS coordinates,

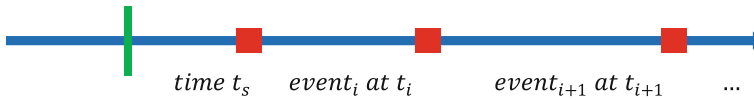


Fig. 5 Once the simulation time t_s passes t_i , CEP-engine and CIPRTrainer will be notified by sending a HTTP push-request containing event-specific data

event-specific information and a tactical symbol describing the event. The trainee can pause and continue the training.

In addition to the scenario executor, the **mapping service** is also one of the core stones in the business layer. The WMS standard is used for acquiring rasterised spatial data such as base layer maps. Another important standard is the Web Feature Service Interface Standard that provides an interface specification for requesting spatial features. It provides vectorised data in various formats such as shapefiles and GeoJSON, etc. MapServer is an open source platform of providing spatial features for GIS applications that incorporates both OGC standards WMS and WFS (see Fig. 6). CIPRTrainer uses it to receive vectorised features like the CI models of the simulators and resources (police, fire-fighters, hospitals, etc.). We use the WMS standard to show flooding on the base layer.

In order to expose the simulation model to the CIPRTrainer, a facade database is developed that aggregates all the involved CI models. The WFS/WMS services extract the relevant information at runtime and push it to the CIPRTrainer front-end. The information includes:

- Geospatial information of CI elements like the coordination of a transformer, the polygon of a railway main station or the polyline of a railway track.

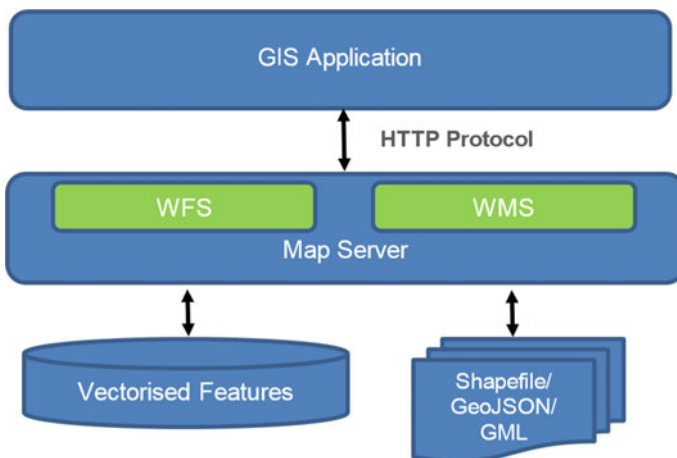


Fig. 6 MapServer implements the OGC Standards WFS and WMS that provide rasterised and vectorised spatial data for rich GIS applications. Spatial data can be stored in relational databases (e.g. PostgreSQL + PostGIS extension) or files that can have various formats (GeoJSON, ESRI Shapefile)

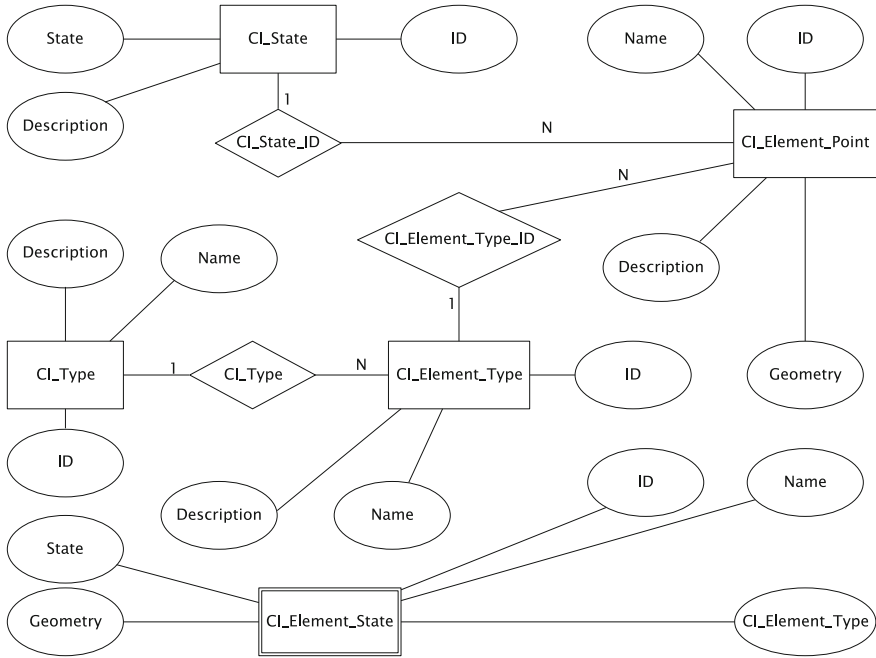


Fig. 7 Entity-relationship (ER) diagram of the CI element state database

- The state information of CI elements like **normal**, **stressed**, **failed** and **recovery**.
- Meta-information like the name, a short description of the CI.

The database design is illustrated in Fig. 7 as an Entity-Relationship diagram in Chen syntax. The design follows strictly the database normalisation form to remove redundancy information storage. Redundant information is provided as various database views (without physically materialise it to the physical storage like hard disks) to ease the access from outside. In general there are several entities listed below:

1. The entity `CI_State` denotes the possible states of a CI or CI element. Basically in the implemented database table, the `State` column contains the four states defined in, i.e. `normal`, `stressed`, `failed` and `recovery`.
2. The entity `CI_Type` contains the domains of CI like electrical network, telecommunication network and railway network.
3. The entity `CI_Element_Type` is about concrete CI elements like a transformer in the electrical network or a router in a telecommunication network. It is different than the `CI_Type` entity. A `CI_Type` can contain multiple types of `CI_Element_Type`. For instance, if the CI is an electrical network, then its element types are transformers, sub-stations, power poles, etc.
4. The entity `CI_Element_Point` models the real instance of the CI elements. It includes the name of the CI element, a short description, the element types and

most importantly the state information and the geo-location. The current design of the database distinguishes CI elements with geometry type POINT, POLYLINE and POLYGON. The reason for this kind of different handling lies in the efficient modelling capability provided by PostGIS, which is used in the database system to handle geospatial objects. In order to efficiently query and store different kinds of spatial objects, the types must be provided during the schema generation phase. In Fig. 7, only the CI element with geometry type POINT is illustrated.

5.2 Federated Modelling and Simulation

For achieving a plausible simulation of the behaviour of CI under perturbations, including failures and cascading effects that propagate failures to other dependent CI, CIPRTrainer employs two commercial simulators (SIEMENS PSS© SINCAL for electricity networks and OpenTrack for railway networks) and one free simulator (ns-3 for telecommunication networks). All these simulators are supplied with models of CI in the scenario area, which are either real or realistic artificial CI models. Information on dependencies between interconnected infrastructures, like which electricity CI element supplies which telecommunication CI element with power, are stored in a database. A failure of the former element triggers a stressed state or failure of the latter element.

Such state changes are represented by software ‘events’ in CIPRTrainer. Each of the simulators is connected to the rest of the CIPRTrainer system by a special ‘connector’ that translates ‘events’ into a format that the simulator can understand. Such a setup of connected stand-alone simulators is called a *federated simulation*. The ‘connectors’ are also employed for synchronising the simulators and for enabling the rollback.

5.2.1 Building CI Simulation Models

The federated simulation environment consists of CI models, dedicated domain specific simulators including both CI simulators and threat simulators, and the simulator connectors that enable the communication with other CIPRTrainer components. In its current state, the CIPRTrainer system’s simulation component is able to simulate electricity, telecommunication and railway infrastructure through the interconnection of dedicated simulators for these types of infrastructures.

For each simulator, a simulation model that reflects the real-world conditions needs to be set up. This process can be compared to the content creation step of traditional video games and involves, due to the needed expert knowledge, a significant part of manual work. To build up a realistic or at least plausible simulation model that matches the real-world infrastructure, data sources like maps or construction plans need to be taken into account. During the modelling process, it

turned out that a part of the information needed is available to the public and on an appropriate level of detail. However, the publication of data revealing the details of CIs might raise security issues, therefore, a significant amount of data is not available to a public audience. For example, the details of electrical distribution networks are not traceable through the Internet.

For the implementation of the cross-border derailment and flooding scenario in the Emmerich area, we used the simulators PSS[®]SINCAL [23] for the modelling and simulation of the electrical transmission and distribution network, the Network Simulation version 3 tools (ns-3) to model and simulate the telecommunication network [24], and the railway simulation software OpenTrack [25] to model and simulate the railway infrastructure.

The model of the electrical transmission network around Emmerich was created manually based on the available data using the graphical user interface of the PSS[®]SINCAL software. The data adopted to build up the model is partially based on OpenStreetMap (OSM) and most of this data seemed to be valid (we checked this for instance, by comparing the geo location with other Google satellite images). The ENTSO-E database can furthermore serve as a means for verification. However, as volunteers collect OpenStreetMap data, there is always no guarantee that OSM data always match the real world. The model of the electrical distribution network within the area of the city of Emmerich is purely fictive, due to the lack of available data sources. The model was built by experts in the field and took a typical city with the size of Emmerich as foundation. An overview of the distribution network model can be seen within Fig. 8.

The structure of this model is based on a typical distribution network of a small city, the constraints and particularities given by the transmission network and the topographic structure of the city were taken into account. The distribution network is modelled up to the 20 kV medium voltage distribution network layer, with cabinet feeders as the endpoints of the network. Each of the cabinet feeders transforms the 20 kV electrical voltages to 400 V low voltages that are delivered to the single houses. As the model does not cover the low voltage network, each cabinet feeder provides power supply to a specific small area within the city and therefore usually supplies several houses with power. In dedicated zones, i.e. industrial areas or the Emmerich harbour area, the model comprises single cabinet feeders, which provide higher output voltages.

The process of creating an imaginary, but plausible CI model was also carried through for the creation of the telecommunication infrastructure model, as in this case, similar to the electrical distribution network, no useful data was available to the public. The model consists of routers, cell towers and interconnecting telecommunication lines. Figure 9 shows an overview of the telecommunication network within the city of Emmerich. The model was also set up by experts in the field, for the positioning of the according routers; the infrastructure of the city was taken into account. For example, several routers were placed close to police stations, schools or hospitals or close to power substations to simulate the need of optimal access to the telecommunication network by these facilities. The course of the telecommunication cables was also adjusted to the need of the CIPRTrainer's

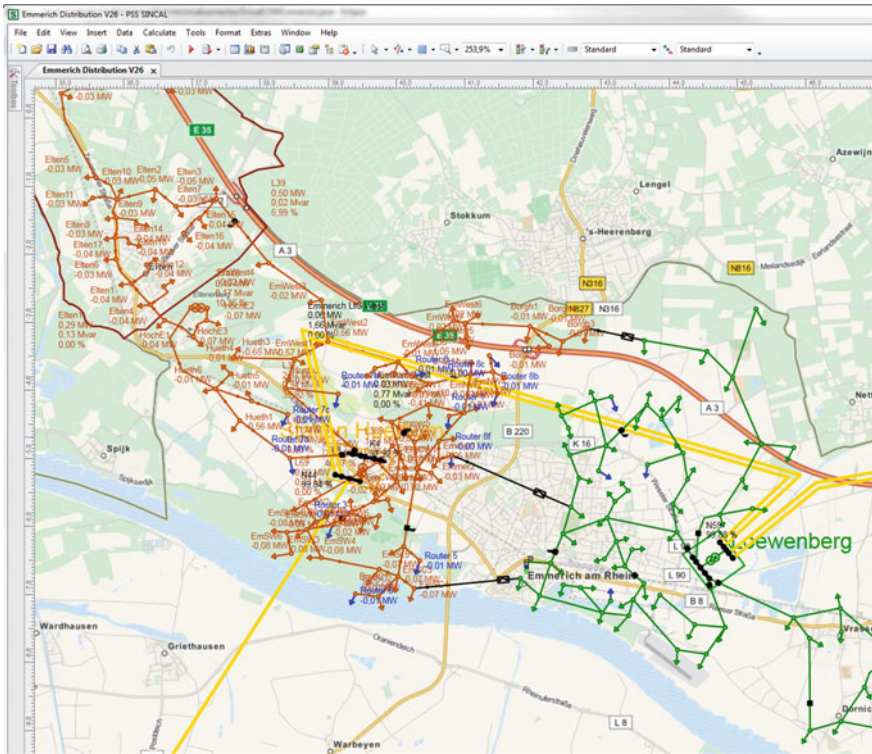


Fig. 8 Screenshot of the PSS[®]SINCAL software. The graphical user interface shows the elements of the distribution and transport network and their interconnections. The view can be enriched with geographical maps to support the modelling of real-world conditions

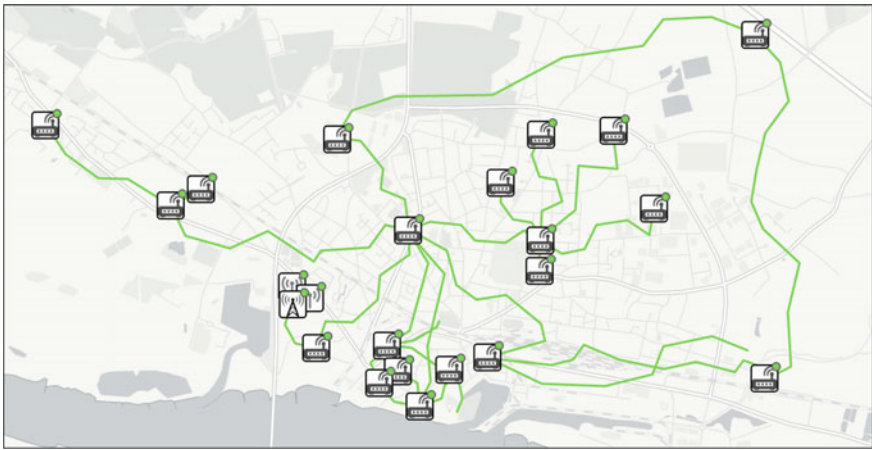


Fig. 9 Visualisation of the fictive telecommunication network within the city of Emmerich. Routers, mobile communication sending masts and telecommunication cables are depicted in an iconographic style

derailment scenario, where an important telecommunication cable gets destroyed during the simulated incident.

Compared to the electrical and telecommunication infrastructure, a number of publicly available data sources for the required modelling activities for the railway network could be used. We used information provided by the main German operator Deutsche Bahn and its daughter companies that are responsible for maintaining the railway network infrastructure. Other information was provided by DB Schenker, a logistics daughter of Deutsche Bahn, Keyrail, and others. Information on local railway traffic in the Emmerich area could be found on the mobility portal of the federal state of North Rhine-Westphalia (NRW) in Germany.

The model was built around the city of Emmerich, including parts the Netherlands and the Rhine-Ruhr area. The most important railway track in the scenario area is the Betuwe route, a double track freight railway from Rotterdam to Zevenaar, with extensions to Germany and the European hinterland (Rhine-Alps corridor of the railway network). It is an important European Infrastructure, as since 2011, nearly 80% of all goods trains between Rotterdam and the Dutch-German border took the Betuwe route. On the same route, also passenger trains are running, including international ICE lines. On the German side, the extension of the Betuwe route runs through the entire district of Kleve, with the city of Emmerich as the north-most station and the city of Wesel as the south-most. The part of the Betuwe route that runs through the incident region is the track between the cities of Arnhem (NL) and Emmerich (DE). Figure 10 shows an overview of the railway network covered by the simulation model.

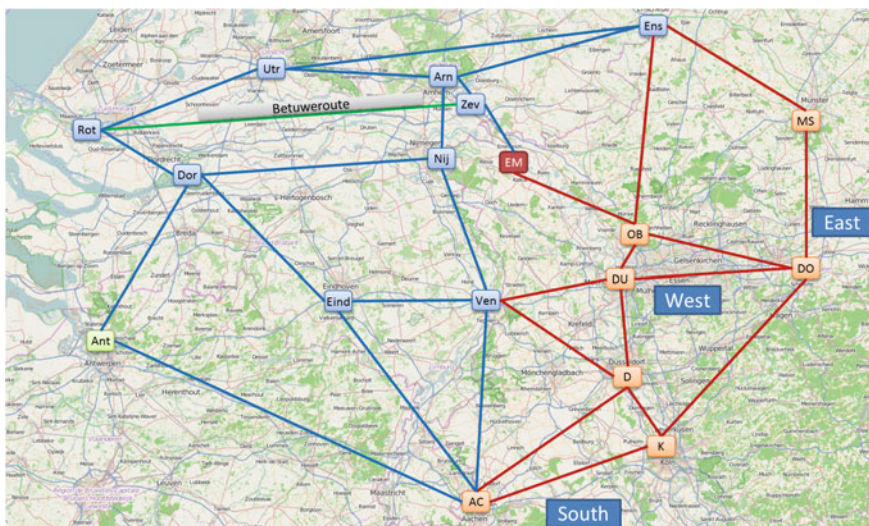


Fig. 10 The model of the main railway traffic lines from Rotterdam (Rot) to the German Ruhr region (OB, DU, DO) and further on southbound via Cologne (K). EM marks the city of Emmerich. Red connections depict railway lines in Germany, blue connections those in the Netherlands and Belgium

The model of the railway network in the OpenTrack software consists of the actual railway network, including tracks, stations and railway infrastructure like signals. Besides the railway network, the model includes the rolling stock data and the timetable information for each simulated train.

5.2.2 The Federated Simulation System

The interdependencies between individual CI simulators are implemented through the CEP engine of the CIPRTrainer software system. The dependencies between single infrastructure elements are described within the rule base of the CEP engine. Simplified, a rule describes a correlation in the form “if cabinet feeder X is inactive, router Y is inactive”. The CEP engine is connected to the CI simulators through an event system. For each simulator involved in the federated simulation, a unified access layer to the basic simulator functions needed within the CIPRTrainer system is implemented, the so-called *Simulation Connector*. The simulation connectors provide access to the specific CI simulators used by the CIPRTrainer and implement functionality for the control of the simulators and for the retrieval of simulation data. As each simulator usually provides its own specific access mechanism, a dedicated simulation connector has to be implemented for each CI simulator used within the federated simulation environment. A schematic overview of a simulator connector is shown in Fig. 11.

Besides the common set of functionalities to implement the connection to the CEP engine, each simulator connector comprises a specialised connection module to the concrete simulator. As each CI simulator provides its own interface, this module has to be implemented for each simulator used within the federated simulation environment, while the common sub-modules can be reused for each new

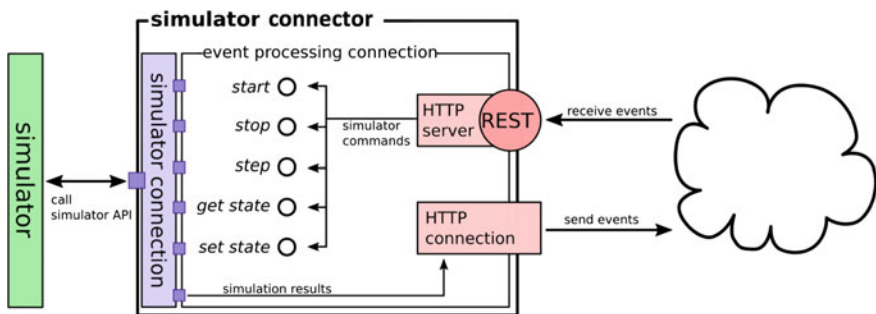


Fig. 11 Graphical overview of the simulation connector. Events are sent and received through the network via HTTP. A REST-based interface to the simulator commands is implemented through a HTTP server. CI element state changes as the outcome of a simulation are sent to the CEP engine using HTTP requests. The simulator connection sub-module realises the concrete connection to a specific simulator

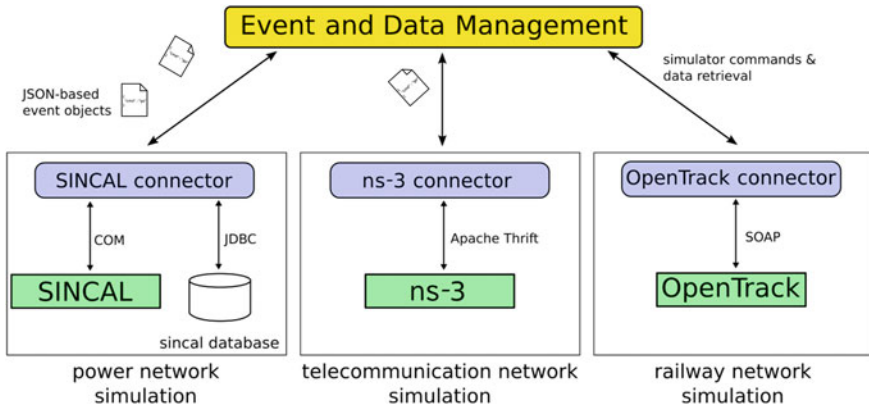


Fig. 12 The simulator connectors (*violet colour*) implemented for the CIPRTrainer with their connection to the specific simulators and to the CEP system (*green colour*)

simulation connector. In Fig. 12, an overview of the three CI simulators used in the CIPRTrainer software system, *PSS SINCAL*, *NS3* and *OpenTrack*, as well as their corresponding simulation connectors and the simulator-specific API (Application Programmer Interface) mechanisms are shown.

CIPRTrainer’s what-if analysis functionality for CM training is based on the fast rollback of various simulated worlds. CIPRTrainer’s different software components including the simulators (both domain-specific CI simulators and threat simulators), visualisation module, time management module and spatial objects support the rollback functionality. However, this functionality is implemented differently within the specific components. For the CI simulators, approaches like the adaption of the software versioning system git, or an internal implementation within the simulation connector are used. For other components like time management, visualisation and spatial information, the spatial-temporal features in the PostgreSQL database management system is used.

6 Impact and Consequence Analysis for the Global Assessment of Damages

In this section we will provide an introduction to the CA approach used in CIPRTrainer; explain which data we used; discuss issues in data acquisition, sanitation, and usage; explain how we displayed the results of CA in CIPRTrainer; discuss how the results should be interpreted.

6.1 Goal of the CA

The overall goal of the CA is to provide the CIPRTrainer users with the capability to understand the broader consequences of their action and inactions during and at the end of a training session. CA goes beyond impacts, as it clarifies the meaning of impacts and the inoperability of critical and non-CI for the population and businesses. A complete and detailed Ca of everything is not possible and not desirable. The CA therefore focuses on the CIPRTrainer user and the information he needs to learn and perform better than before. The CA module (CAM) of the CIPRTrainer is not intended to be a finished product readily usable for wide variety of conditions. Just like CIPRTrainer as a whole, it serves as a working demonstrator for the specified scenarios. But in general it should be possible to adapt the CAM to different scenarios. Therefore it needs to be conceptualised and implemented in a way that allows later modification.

6.2 General CA Concept

For the CAM we distinguish between impact and consequence. Impact is the direct outcome of an event, for example the destruction of a private house or the reduction/loss of function of an infrastructure. An impact has consequences, for example the rebuild cost of a private house or the economic losses due to the reduction/loss of the infrastructure function for infrastructure stakeholders. Impact can be differentiated in direct and indirect impacts. Direct impacts are the direct damages to CI elements or other assets. Indirect impacts are the cascading effects.

Consequences can also be differentiated into direct and indirect consequences. Direct consequences are directly related to the impact, for reconstruction cost of a flooded building. Indirect consequences are indirectly related to the impact, for example the number of homeless persons. A further differentiation is possible in CI related and other consequences. CI related consequences are related to the impacts of an incident on CI, for example the recovery costs for a CI operator, the GDP loss produced by a power outage or the number of households without electricity. Other consequences are related to the impacts on everything else, for example the re-construction cost of flooded houses. CA therefore comprises the estimation and assessment of these types of consequences of impacts.

6.3 Geographical Dimension of the Analysis

The CAM takes the geographical dimension of the impacts and consequences into account. For the CAM it is important where an impact has happened and where the consequences occur, which is not necessary the same area (cascading effects and



Fig. 13 Area of Emmerich with German 1 km × 1 km grid and power nodes and lines (OpenStreetMap)

indirect consequences). In the CAM we use two-dimensional grids to locate people and the built-up area (buildings, infrastructure and environmental areas). These grids are INSPIRE compliant.² For Germany we use a 1 km × 1 km grid (see Fig. 13), for the Netherlands a 500 m × 500 m grid, which are the standard grid sizes that these countries use.

For both grids census data from 2011 [26] is used, which comprises data on residents and residential buildings per grid cell. For German business data we had access to a derived spatial business dataset on street level. The data set was derived from different databases from commercial data providers:

- Deutsche Post Daten
- NAVTEQ (HERE)
- Microm consumer marketing

From the derived dataset we extracted the number of firms with specific NACE code on street level and allocated these firms to the grid.

²The INSPIRE directive (Infrastructure for Spatial Information in the European Community) aims to create a European Union (EU) spatial data infrastructure. This will enable the sharing of environmental spatial information among public sector organisations and better facilitate public access to spatial information across Europe <http://inspire.ec.europa.eu/>.

For data of land use we use CORINE land cover data from 2006 (see [27]), which are available free of charge. Infrastructure data with geographic coordinates could be derived from OpenStreetMap. It has to be mentioned that this data is far from complete as it depends on OSM users to insert the data sets. Some regions are more detailed than other regions. But for the purpose of CIPRTrainer it was sufficiently complete for the district of Emmerich.

For the use of grids in the CAM one important assumption was necessary: If a hazard has an impact on a grid cell, the whole grid cell is affected, e.g. if the cell is only partly flooded the assumption is that the whole cell is flooded. This assumption is necessary as we have no information about where in the cell the resident or buildings are located. This can lead to an overestimation of the consequences. With more detailed data it would be possible to relax this assumption.

6.4 *Determining Impacts*

The technical federate simulators are only able to provide impacts and cascading effects for their domain (electricity, telecommunication and train traffic) and the flood simulator does not produce any impacts by itself, it only calculates the geographical extent of the flood with attributes for height and rise rate. So we needed a separate impact module for the flood impacts and all other impacts of the different scenarios. It has to be noted that some impacts are not calculated; instead they are part of the storyline and therefore predefined. An example is the train derailment in the Emmerich Scenario, where the amount of damage to humans and buildings form the train crash is defined in the storyline.

Impacts on humans can lead to injuries and death. For operationalization mortality functions can be used. We based our approach on a general framework for loss of life estimation from Jonkman et al. [28]. Basis principle is to look at the exposed individuals to a certain hazard. If the people are informed they can shelter (i.e. keep the door and windows closed when a chemical cloud is coming, going upstairs in a flood etc.). They are exposed to the threat if they cannot shelter or self-evacuate themselves. Emergency forces can evacuate them if present (depends on trainee decision), otherwise they are exposed until the end of the threat. The effects of the impact on the exposed people are calculated with hazard specific mortality functions. The more intense an impact is (e.g. high flood depth and rise speed of water during a flood) the more casualties are to be expected. The inherent mobility of humans brings some conceptual issues. Usually residential data is used to assess impact on humans. But this leads to an overestimation of impacts on residential areas in the daytime, as normally a big part of the residents is at work (or school, university) or pursue other activities (shopping mall). There are different solutions discussed in the literature. More static approaches use a simplified binary distinction between daytime and night time distribution (see [29, 30]). Others try to develop models of dynamic behaviour of residents, which is a more difficult task (see [31]). Regarding CIPRTrainer scenarios the data available are not sufficient for

the dynamic modelling of the population. Then, a simplified approach is used considering only residential data.

The impacts on buildings, infrastructure elements and environment are conceptualised in a similar way to impacts on human. First these objects need to be physically exposed to a hazard, e.g. a house must be in the flooded area. Second the object must be vulnerable to the hazard. The damage depends of the intensity of the hazard (e.g. flood depth) and the degree of sensitivity of the object to the specific threat (e.g. the main material of the building: wood vs. brick). Some damage functions for specific threats and specific objects are available in the literature. For flooding we could draw upon the ‘Standard Method 2004 Damage and Casualties Caused by Flooding’ from the ‘Ministerie van Verkeer en Waterstaat’ [32] of the Netherlands and the book ‘Hochwasserschäden’ [33] for Germany. But not for all types of objects and hazards are damage functions readily available. In these cases we have made assumptions on the basis on available damage functions.

6.5 *Evaluating Consequences*

For the CA we decided to evaluate the consequences on humans only as the number of injuries and deaths. As an economic evaluation of life is impossible and would be a highly ethical issue, we refrained from doing so.

To assess the direct consequences on a specific building, infrastructure element or environmental area, we need a metric to express the value of the damage. We decided to use reconstruction cost for this purpose, because information about the potential reconstruction cost of residential, commercial, industrial and public buildings are derivable from official data on build cost in Germany and the Netherlands. For infrastructure elements and the environment the data is not readily available. So we had to rely on diverse pieces of information in different studies, surveys, books and websites to generate artificial data. To calculate the actual reconstruction cost for a specific object a damage factor is needed. This is conceptualised as a value between 0 and 1, with 0 no damage and 1 total destruction. This damage factor is determined by the impact module (e.g. flood-depth-functions). The actual reconstruction cost of a specific element is defined as a function of the damage factor. Figure 14 shows an example of a flood damage function for low-rise dwellings from [32].

For indirect economic consequences there are basically two major streams in economic theory: input-output models (IOM) and calculable general equilibrium models (CGE). Both modelling approaches address the interaction of the different economic sectors. They differ however in which manner these sectors interact and how the sectors react to external shocks [34, pp. 43–44, 35, pp. 116–118]. IO-models focus on the interrelations of production, where a sector needs inputs from other sectors to produce goods. In the basic IO-model prices don’t play any role. On the other hand focus CGE models on the effects price variations to the supply and demand in the different sectors [34, p. 44]. The basic IO-model is demand driven. A disaster can therefore only be modelled as reduction of the final

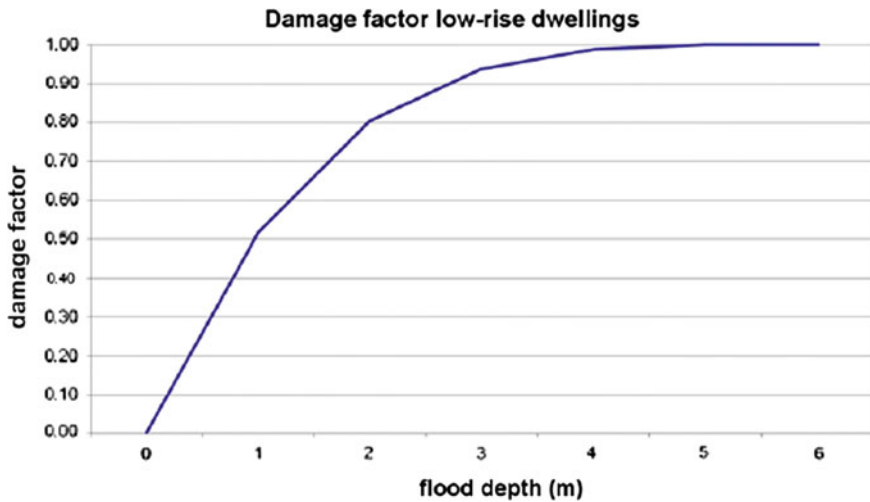


Fig. 14 Example of a flood damage function for low-rise dwellings [32]

demand. This causes a decrease in the production of final goods and subsequent in all dependent sectors who supply intermediate or raw goods for this final goods. A loss of production of a supplier firm due to a disaster has correspondingly to be modelled ‘in reverse’. In newer IO-models this restriction has been relaxed. In contrast to IO-models there are no explicit flows of goods in a CGE model. The economic system is always perfect balanced due to the price mechanism of the markets. A reduction in production capital due to a disaster leads to a decrease of supply and a subsequently to a price increase. This leads in turn to a demand reduction and to a new equilibrium. Moreover, in CGE models production factors can be substituted in short term. This induces a fast adaption of firms to mitigate the disaster effects. CGE models are thus more optimistic than IO-models, where production technologies are fixed in the short term. One limitation of both approaches is the high aggregation level. Sectors are the main “economic actors”. If one sector suffers from a disaster, all businesses aggregated in this sector suffer the same consequences, regardless of spatial location. There are no distinct production functions and no explicit supply chains modelled in the sector [34–36].

In the CAM the method of IOM is used to calculate the indirect effects of the disturbance of economic sectors in the CAM. In the course of time many variations and extensions of the basic IO-model were proposed in the literature (e.g. inoperability IOM [37, 38], supply driven IOM [39], but we decided to start with the basic model as it is the least data hungry and easiest to understand for the CIPRTrainer user. In the future an enhanced IOM could improve the explanatory power if needed.

One obstacle for the use of IOM in the CIPRTrainer is the lack of regional input-output data. There are proposals in the literature how to regionalise national data (see), but using one of these methods is a very complex and time consuming

task, hence not feasible in the timeframe of the CIPRNet project. Our approach was to assume that the regional input-output structure is the same as on the national level. The absolute values for the different sectors are proportional to the GDP share of the region. For the district 'Kreis Kleve' the GDP share was 1.3% of the national GDP in the year 2011. The IOM uses this 'regionalised' IO-table data.

Another obstacle is the lack of output data on business level, i.e. how much a business is producing in one year. We had to refer to a combination of value-added data on the district level and data on the number of firms with a specific NACE code in the district to generate a dataset on value-added per firm with a specific NACE code.

7 Using CIPRTrainer

CIPRTrainer is accessible through a regular Internet Browser. If all installation procedures are completed successfully, then the CIPRTrainer should be accessible on <http://host-machine-ip/ciprtrainer>. CIPRTrainer is a multi-user training system, which includes following user roles: trainees and trainer. Before a training session starts, the trainer prepares training scenarios for trainees. Whenever the trainer starts the scenario simulation, the pre-defined storyline will be executed and trainees can perform training. In the following, user roles, trainee and trainer modules will be described precisely.

7.1 User Roles

A study of the EU project PREDICT showed that although the CM governance structures in different countries vary to a great extent, there are some common roles of CM staff. CIPRTrainer supports the most essential of these roles. Typically, there are one or more persons responsible for collecting information on the situation ('situational awareness'). One or more persons are in charge of the CM ('decision taker' or commander or head of CM staff etc.). CM teams include people commanding the responders, like the head of the fire-fighters, head of police etc. ('operations'), and people heading municipal departments, like the head of the school department ('administration').

For this purpose, there are four different roles for trainees in CIPRTrainer: Situational awareness, operations coordinator, and administrative coordinator operate CIPRTrainer simultaneously. We called the latter two roles 'coordinator', since they combine functions that in reality would be assumed by more than one person. For pragmatic reasons and for feasibility, we restricted the number of simultaneous users to three (plus trainer). For each of the three roles, a specific set

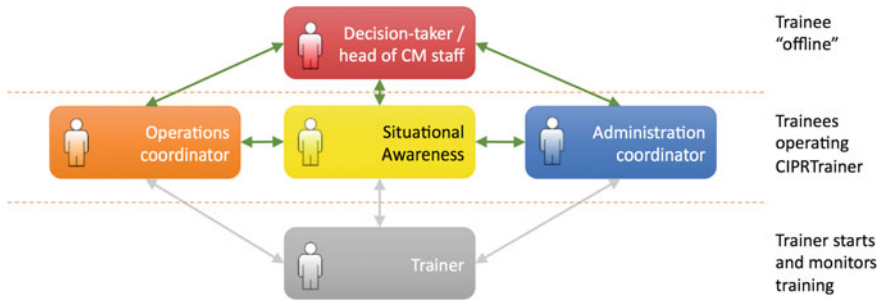


Fig. 15 Trainees and trainer user roles in CIPRTrainer

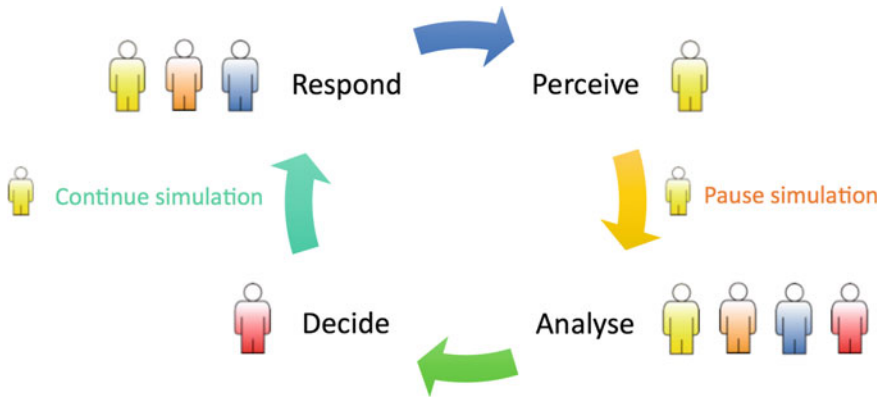


Fig. 16 CM cycle of perceiving, analysing, deciding, and responding. User roles in CIPRTrainer related to the phases of the cycle

of actions can be performed in simulation. A fourth trainee, the decision taker (or commander or head of CM staff), stays in the background (Fig. 15). CIPRNet has chosen this approach for supporting the wide applicability of CIPRTrainer.

The four trainees simulate the repeated CM cycle (Fig. 16) of situation update (perceive), situation analysis (analyse), decision taking (decide), action planning and execution (respond) [40]. The trainee acting as situation awareness staff member pauses the simulation for initiating the next cycle.

7.2 Trainee Module

Trainees have various options to interact with the system including:

- Pausing and continuing the scenario
- Performing various kind of actions (response, crisis management, administrative)
- Performing rollbacks (jumping into a prior state of the scenario)

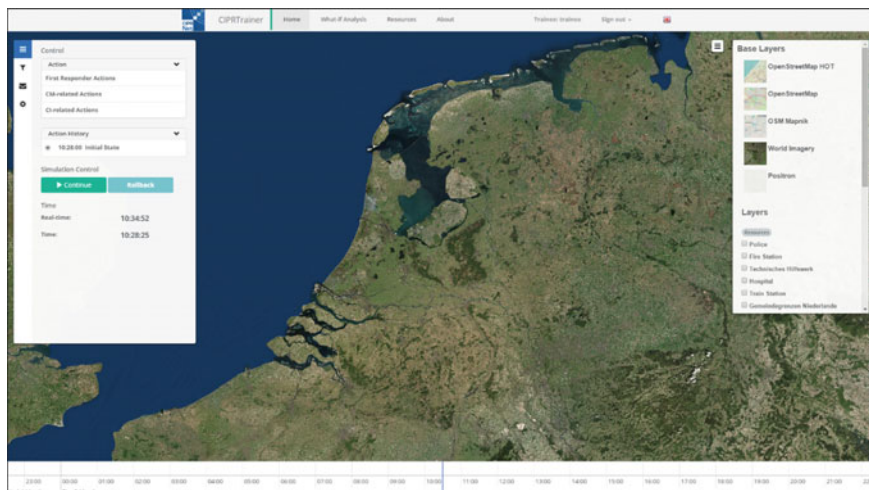


Fig. 17 Main view of CIPRTrainer UI for trainee consists of a navigation bar, two sidebars (*left* and *right*), a time-line (*bottom*) and a standard GIS map (*centre*). The *left* sidebar contains the *green* “Continue” button for resuming a paused simulation

- Examining and mapping CIs
- Observing and keeping track of the evolution of the scenario
- Customising CIPRTrainer view components (layer panels, timeline, etc.)
- Conducting CA
- Communicating with other participants

Pausing and continuing the scenario

On the left control panel of CIPRTrainer the user can click on the button Pause/Continue to pause or continue the scenario (Fig. 17). Other participants like trainees and trainer are notified when the scenario is paused or continued.

Performing various kind of actions (first responder and crisis management)

CIPRTrainer provides various kinds of actions including first responder and CM related actions. A user-specific list of actions is shown on the control panel. For instance, the operations coordinator is able to perform first responder related actions such as mobilising fire brigades and so on. In the following all actions are described in detail.

First Responder Actions incorporate resources like police, fire brigades and technical relief services (THW) and can be performed by operations coordinator (Fig. 18). The trainee can order different resources to perform an activity at a predefined location. Actual forces or resources are spread in the region around the scenario location (in this case Emmerich).

CIPRTrainer provides a map that depicts the different resources and their capacities. The capacities are presented as triple (leader, subleader and forces,

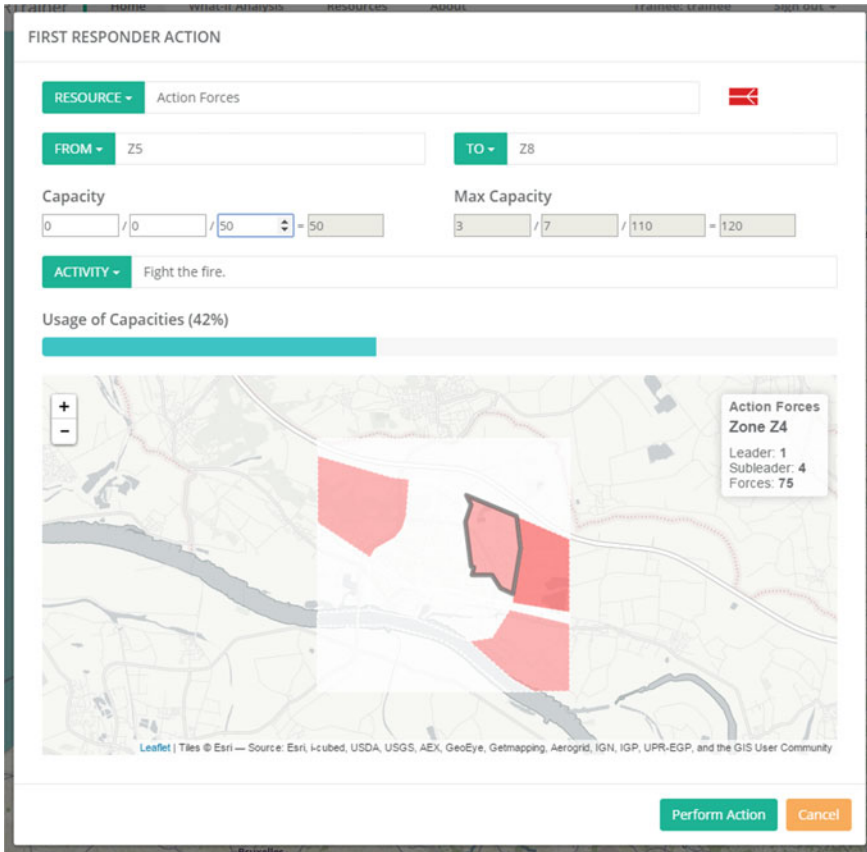


Fig. 18 Interface for performing first responder actions

see Fig. 19). The accumulation of these three values refers to the capacity strength of a specific unit. The user is able to send resources with a certain capacity to the crisis region. In addition to that, the trainee can select an activity that the forces perform in the crisis zone (e.g. fight the fire). Every participating trainee gets a notification whenever an action is performed (Fig. 18).

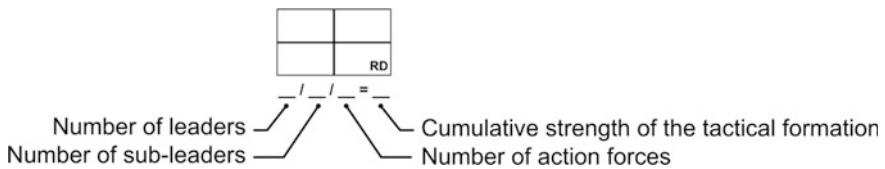


Fig. 19 Tactical symbol notation for strength details of units

CM actions are referred to as the second type of actions that CIPRTrainer provides. This type of actions can be performed by operations coordinator, administrator coordinator and the trainee who is responsible for the situational awareness. However, the actions between the trainees differ and are based on their decision-making authority. For instance, the trainee for situational awareness is able to alarm public authorities and the general public as well as request more emergency forces from other districts. Each action influences the consequences in the scenario. When the trainee decides to not inform the general public, the number of injured people will rise. On the other side, if the crisis manager performs this action in an early state of the scenario, then less people will be injured, which mitigates the consequences. Each action triggers predefined rules based on the action, the time and the underlying socio-economic data. The following lists available CM related actions for the different trainees.

Trainee for situational awareness:

- Request electricity power cut-off from electricity supplier in <place>
- Request locking the railroad track from train authorities
- Inform companies in the area that work with dangerous goods
- Contact European emergency response capacity
- Contact chief administrative officer of the district
- Request more emergency forces from other districts

Administration coordinator (see Fig. 20):

- Inform hospitals to prepare for casualties
- Prepare evacuation
- Support evacuation
- Inform the public by media (press, radio, television)
- Request support from municipal transport services for evacuation
- Block all critical bypass roads like tunnel and bridges

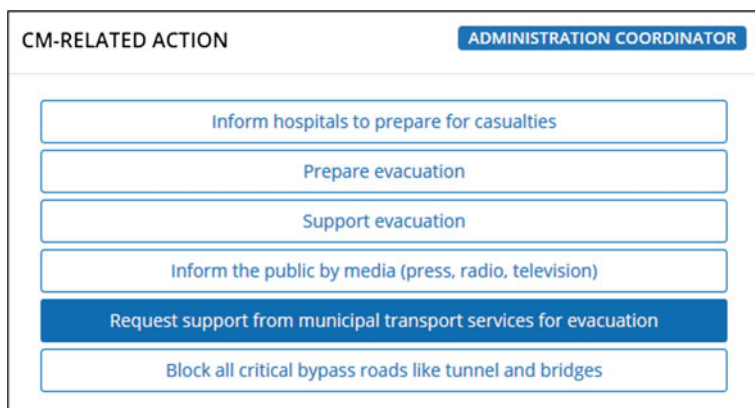


Fig. 20 Interface for performing CM actions by administration coordinator

The operations coordinator:

- Inform the public by sending action forces with speakers and sirens
- Fight fire
- Recover affected victims/humans
- Evacuate the accident site (initiate evacuation)
- Request special forces
- Block all critical bypass roads like tunnel and bridges
- Warn the public by using air raid sirens

Performing rollbacks (jumping into a prior state of the scenario)

To perform a rollback the user needs to decide first on which state the scenario should be reinstated. Therefore, CIPRTrainer adds every performed action in the Action History list. To perform a rollback the user needs to select an action in the Action History list and click on the button “Rollback”. The scenario is then restored to the time the action has been performed.

Examining and mapping CIs

CIs can be examined using GIS overlays. The user is able to visualise and hide CI overlays by using the layer panel on the right side. To examine an entire CI model, for instance power networks, the user can click the list element highlighted with a grey background. By doing so, all components of a power network (e.g. cabins, substations, transformer, etc.) will be checked as well, and shown on the map. CI elements on the map are visualised using icons representing the entity with an underlying LED light on the upper right corner. This light indicates the operational status of the element. Table 1 shows the relation between colour and operational status of a CI element.

Observing and keeping track of the evolution of the scenario

The user has three options to observe and keep track of the evolution of scenario, namely using:

1. GIS map
2. Timeline
3. Notification Logs

Table 1 LED lights on the upper right corner of CI elements indicate the operational statuses

Colour	CI State
Green	Normal state; service up
Yellow	Service partially shut down; no substantial damages
Red	Service completely shut down due to damages
Blue	Service completely shut down, but currently no damages

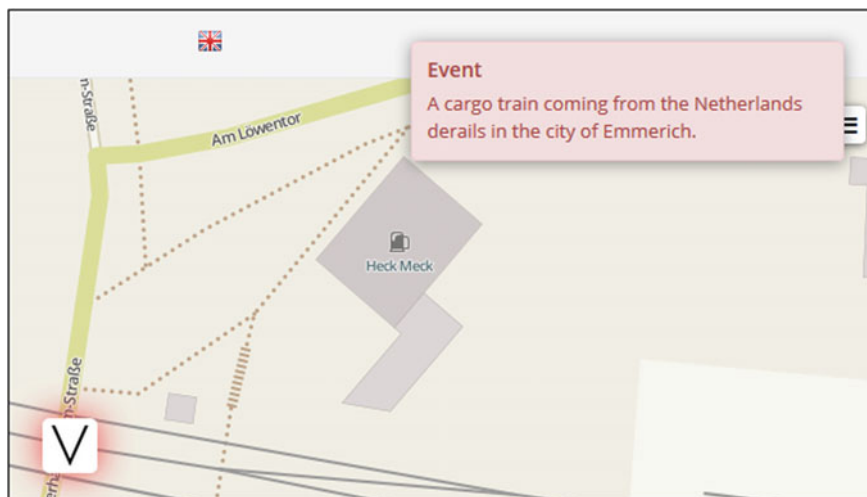


Fig. 21 Event notification label on the *upper right corner* depicts a short summary of the event

CIPRTrainer GIS map visualises storyline events and status reports as GIS elements using the corresponding tactical symbol on the map. When CIPRTrainer receives an event by the Scenario Executor, the team will get a notification showing up on the upper right corner containing a short description of the event. Simultaneously, the event appears on the map, too (Fig. 21).

Clicking on the item reveals a pop-window with more detailed information showing a short description and accident time (Fig. 22). To gain more information, the user can click on the 'Read More' button. A new window pops up including all information of this event (Fig. 23). To close the window pop-up, the user can push key ESC or click outside the message box. To hide the all events on the map, the user can toggle the list element 'Critical Events' on the layer panel.

The CIPRTrainer timeline (Fig. 24) shows various kinds of events in a chronological order. Different event types have different colours (Table 2). To know more about the event, the user can click on the label. A window pop-up shows up and can be closed by pushing the key ESC or click outside of the window.

Managing user account and customising CIPRTrainer view components

The trainee can change first name, last name, email address and password in the settings panel. To customise CIPRTrainer, the user is can show or hide certain UI components by checking or uncheck the desired component on the list, respectively. Following components are listed:

1. Base Layers
2. Layers
3. Timeline

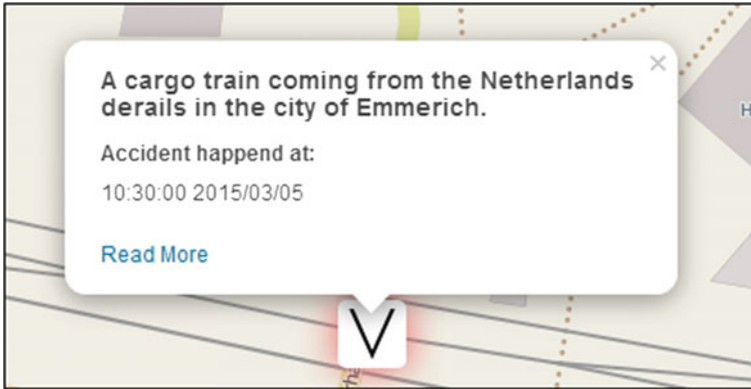


Fig. 22 Information panel of a GIS element appears by clicking on it

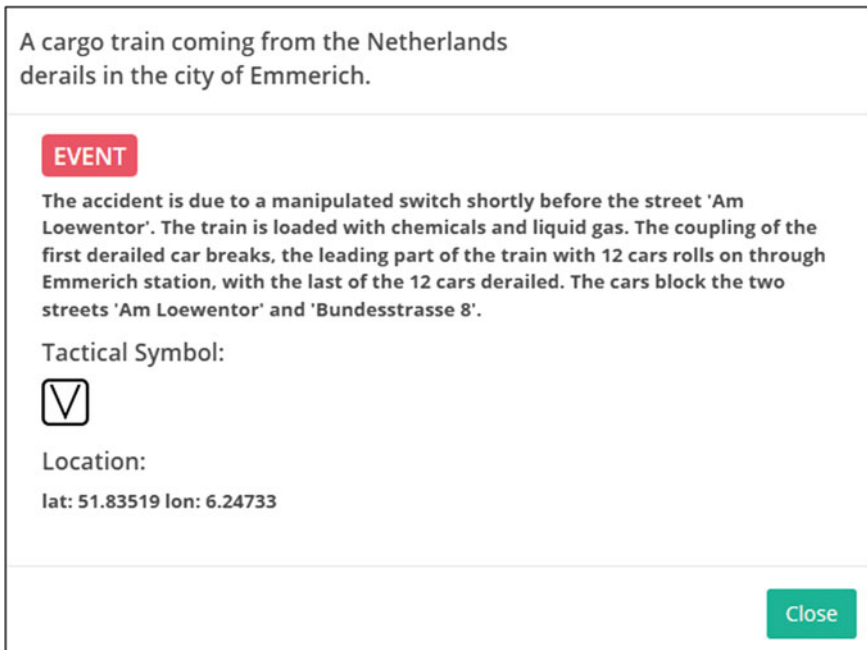


Fig. 23 Window panel containing all information shows up by clicking on the element on the timeline or “Read me” button

Conducting CA

For using the CAM, the team can click on the button ‘What-if Analysis’ on the navigation bar. The first section contains information about performed actions and rollbacks. The graph is an n -dimensional tree containing x nodes, where n reflects

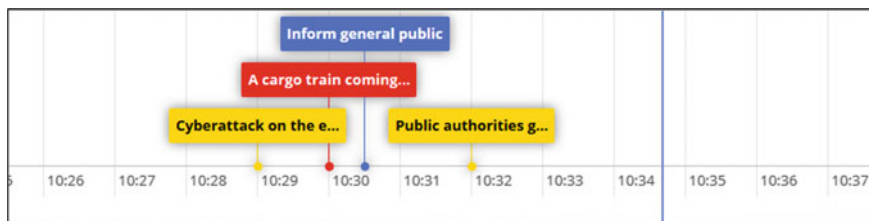


Fig. 24 Timeline depicting various types of events

Table 2 CIPRTrainer incorporates various types of events: action events, events defined in the SDL, events produced by the federated simulators and general non-georeferenced events

Event Type	Colour	Location
Action events (performed by trainee)	blue	Partially
Events defined in the scenario	red	Yes
Events produced by the federated simulation	red	Partially
Non-georeferenced events and other events	yellow	No

the number of performed rollbacks and $x - 1$ the number of performed actions. The start node (marked as yellow in Fig. 25) illustrates the initial state of the scenario. Green nodes represent final actions, on which CAM can be performed. To examine the graph, the trainee can click on the nodes to read more details about the actions and use the navigation controls of the network panel.

Please note, whenever the what-if analysis view is active the simulation automatically pauses, if it was running, or remains in the state stopped, if trainer stopped the simulation before. To conduct the CA, the trainee can click the button “Acquire CA Results”. Several computation processes are started in the backend. CIPRTrainer shows CAM results using:

1. Tables
2. Diagrams
3. GIS map

The tables contain information about various kinds of damages without geospatial context (Fig. 26), whereas the GIS map depicts several spatial-related results using colour schemes to support map diagnostics (Fig. 28). Diagrams are used to compare training results (Fig. 27).

The trainer’s main tasks are choosing, starting and stopping scenario, and acquiring CA results and training protocol. Therefore, CIPRTrainer provides a

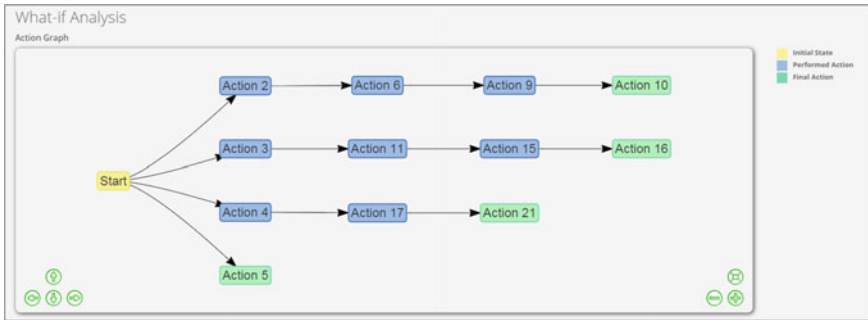


Fig. 25 Example of action graph. Each node of the n-dimensional tree refers to an action. The rollback capability creates an additional branch, on which following actions can be added. Final actions are marked as *green* nodes. The root node corresponds to the initial state of the scenario. In this example, the end-user performed four rollbacks

TOTAL COSTS AND DAMAGES	
Category	Value (EUR/Amount)
Reconstruction Cost Residential Building	0
Reconstruction Cost Business Building	0 EUR
Reconstruction Cost Infrastructure	0 EUR
Value of Lost Loads Households	0
Emergency Forces Cost	0
Number of Injured Humans	2372
Number of Dead Humans	0

ACTIONS		
Action Cost (EUR)	Action Hours (h)	Number of Forces

Fig. 26 CA table showing CA results that are non-spatial

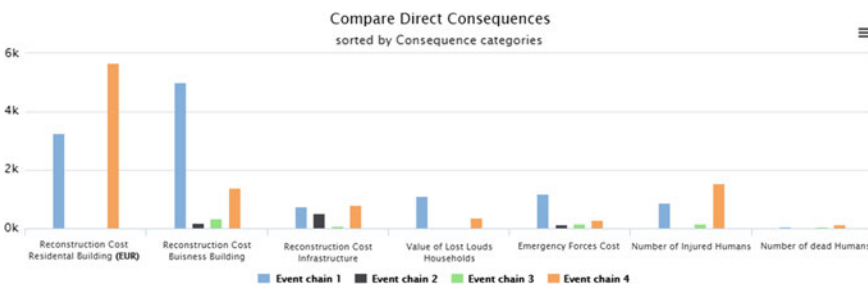


Fig. 27 CA diagram showing four different training sets

dashboard that includes all important information for performing the major tasks (Fig. 29). The dashboard includes:

- List of possible scenarios to train with
- Scenario status (paused, continued), simulation time and real time
- Online/offline status of the participants

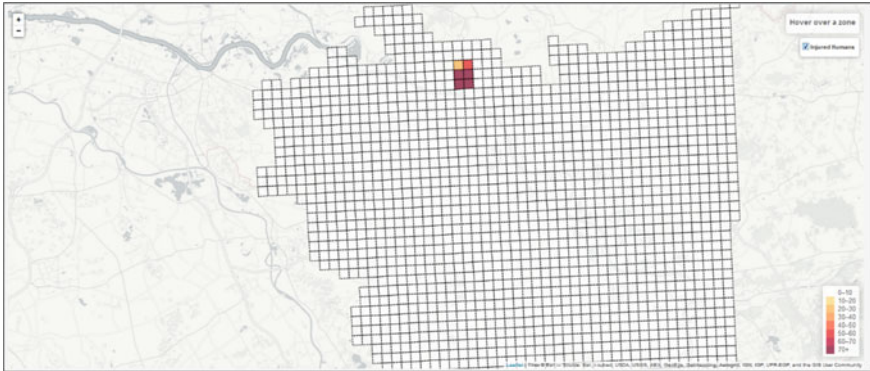


Fig. 28 GIS map showing spatial results

CIPRTrainer TRAINER Sign out

TRAINER DASHBOARD

Status: **running** Scenario Time: **11:55:35** 2015/03/05 TRAINER: **Zeus** online since: 19:52:29

ADMINISTRATION COORDINATOR: **Apollo** online since: 19:52:54
OPERATIONS COORDINATOR: **Poseidon** online since: 20:01:22
SITUATIONAL AWARENESS: **Hebe** online since: 20:01:44

Train Derailment 2015/03/05 10:15:00
Derailment in Emmerich am Rhein
Emmerich am Rhein, Germany

Flooding 2013/03/05 12:11:00
Cross border flooding with a major breach
North Germany and the Netherlands

Train Derailment 2015/03/05 10:15:00 Emmerich am Rhein, Germany

The accident is due to a successful cyberattack on the central network of the railway company. A railway switch was manipulated and adjusted to a false position. When the train passes over the wrong angled switch with 90 km/h speed, it comes to a derailment. The train consists out of 42 railway cars and its length is 700 meters. The train has loaded liquid gas and other inflammable chemicals. The coupling of the locomotive car breaks after the 12th waggon. The leading part of the train with 12 railway cars rolls on through Emmerich station. The second half of the train is lead to the wrong railway track and is partially derailed. 20 of the remaining 30 cars are crashing into the buildings along the left and right side of the railway tracks. The streets "Am Löwentor" and "Bundesstrasse 8" are blocked due to the derailed cars.

100 m
car 3
car 14
car 35

Training Logs

Filter Protocol Logs

- Info: 14:09:03 (real time): Apollo Apollo went online.
- Warning: 10:29:00 (scenario time). Cyberattackers manipulate a railway switch.
- Warning: 10:31:00 (scenario time). The train derailment destroys the railway track and 5 buildings. Also 4 humans are dead and 54 are L...
- Warning: 10:32:00 (scenario time). Police arrives and cordons off the accident site
- Warning: 10:31:00 (scenario time). Some of the waggons crash onto the street and also into different buildings. They are marked as if t...

Fig. 29 The trainer dashboard contains information about the running scenario, the statuses of the participants, a list of possible scenarios, of which the trainer can choose one to run; control functions for starting and stopping the scenario, and the possibility to download results of the CA and training logs

- List of training logs including trainees interactions with the system
- Download panel for acquiring training analysis results

Setting up the scenario

Trainer can choose a scenario from the scenario list. By clicking on the item, a description of the scenario will show up and a button for starting or stopping the simulation. The button can only be activated when the team is complete.

Download results

For downloading the CA results and training logs, the trainer can navigate to the download panel and click on “Download All” to acquire a single JSON file with both CA results and training logs.

8 Example of a Training Session

In this section we will briefly describe the train derailment scenario storyline; describe what actions a trainee could perform; what critical decision points the scenario contains; explain how one anonymous trainee used the system; explain which what-if actions the trainee performed; explain what insights were gained in that session.

In the train derailment storyline, we assume a sudden derailment of a cargo train in the city centre of Emmerich am Rhein, caused by a malfunctioning switch point due to a cyber-attack on the electronic railway control centre Emmerich. Fire, spilled chemicals and a toxic gas cloud affect citizens, built infrastructure and CIs. The immediate impacts of the crisis take place within a few hours, while the remedy of the impacts takes days and weeks. The scenario consists out of a sequence of events that simulates the different stages of an accident that leads to the destruction of components. The different events require actions that are supposed to minimise the expansion of destruction and injuries.

The scenario is initiated with a cyber-attack event that manipulates a railway switch near the control centre in Emmerich. This manipulation leads to a derailment of a cargo train that has cars loaded with chemicals and liquid gas. Those cars crash onto the street and also into different buildings. After this accident happens the information is published through different channels. At first the train conductor informs the railway control station and in parallel the general public inform the police and fire brigade. Since Emmerich is quite small the police arrives contemporary and starts to cordon off the accident site. Also the mayor officially calls the disaster event.

Until this point the CIPRTrainer trainee user was not able to interact with the scenario storyline. Now the user is able to select different actions that may support the operation and lead to less destruction and injuries within the scenario. The following lists some of the available actions:

- Send action forces/rescue forces/law forces from <location> to <location>
- Inform the public by media (press, radio, television)
- Inform the public by sending action forces with speakers and sirens
- Inform hospitals to prepare for casualties
- Cordon off the scene of accident
- Recover affected victims/humans
- Evacuate the accident site
- Evacuate population from <location> to <location>
- Request special forces
- Request locking the railroad track from train authorities
- Inform companies in the area that work with dangerous goods
- Block all critical bypass roads like tunnel and bridges
- Request more emergency forces from other districts.

Most of the actions are generic and only few actions need a parameter for the location from where the resources are taken and assigned to. First thing that would be expected by the trainee is to act accordingly to the recommend disaster principal actions: danger recognition, cordon off the area, recover victims and request appropriate Special Forces.

The next storyline event that influences the disaster area are chemicals flowing out from the railway cars and ignite in the middle of the street near several buildings. It destroys the area around the accident location and part of the railway tracks infrastructure. There is an imminent risk of further explosions of chemicals. A toxic gas cloud emerges from the fire and the wind blows it in north-eastern direction. In between the railway control station switched off the power from the overhead electricity, this allows fire-fighting operations in the centre of the disaster area. At this time an additional railway car that was loaded with liquid gas explodes. Because of this, nearby buildings get destroyed, chemicals flow into the sewer system and ignite. Power lines inside the sewer systems are destroyed as well as telecommunication components. The toxic gas cloud diffuses in the area of Emmerich and affects humans and businesses. The Rhein-Waal Terminal in the port of Emmerich has to stop working and the harbour is inoperable.

While all these events appear inside the CIPRTrainer GUI the trainee is able to execute actions. Some of the destruction events can be avoided by executing the appropriate actions that obviate further destructions. For instance the destruction of power lines and telecommunication components inside the sewer system can be avoided by sending enough fire forces to the disaster location. In case that the trainee send 30 or more fire-fighter, the explosion of the railway car can be avoided and the subsequent destruction events will not occur. The toxic gas cloud requires a wide-spread evacuation and information action so contamination of humans can be reduced.

The what-if analysis enables the trainee to try different courses of action in the scenario during a training session. The underlying concepts are 'rollback' and 'consequence analysis', which have been described before in detail. During the training the trainee can request the CA and get the results for the current training branch. After that he rollback to a former time point in the simulation, try out

different actions and request once again the CA. This enables him to compare the different outcomes of the incident evolution and his own actions. In our example the trainee could compare the results of sending different numbers of fire-fighters to the hotspots in the scenario.

After the training the trainer can give feedback to trainee about his performance during the training. Besides the result comparison, the trainee can learn how he reacts under uncertainty and time pressure. In contrast to real-world crisis the trainee can learn from trial and error, due to the what-if analysis capabilities of CIPRTrainer.

9 Outlook

CIPRTrainer provides new advanced training capabilities to crisis managers for decision-making in crisis situations. What-if analysis and CA implemented in CIPRTrainer provide a comprehensive solution for scenario-driven CM training. Despite the sophisticated functionalities provided in CIPRTrainer, there is still room for improving it such that it better meets the needs of CM training.

- Multi-user training support will be improved to enable a better collaborative training experience.
- A community-driven European scenario database (ESDB) built on top of the Scenario Description Language SDL will be developed. Combining CIPRTrainer with ESDB will provide a large set of training options for different crisis scenarios.
- Decoupling simulators from the CIPRTrainer core engine. Other organisations and institutions will be able to plug-in their own simulators. The open communication interface will be published, so that third-party simulator providers can use CIPRTrainer as a training platform, which uses the simulators during the training session. Use CIPRTrainer as a kind of cloud-based training for crisis management.
- Advanced visualisation of the CAM is currently under development and will be available in the future release.
- To deploy the system on other sites, a significant amount of efforts and know-how is still needed. Easy deployment with container-based solutions like Docker will be integrated. Due to data privacy and security, cloud-based solutions are not optimal.

Moreover, the limitation of syntactic checking will remain for the time being. Semantic checks are an option for the future work. Extensions of the federated simulation system would require the development of connectors if new simulators need to be added. This is a design feature of the way we implemented federated

simulation (cf. also [DIESIS]). Developing new connectors requires a deep understanding of the RESTful interfaces and the simulator interfaces.

To maximise system performance, several functionalities will be moved into the database management system as extensions to avoid the overhead of network transmission of data, similar to what has already been done in the CAM. Validating SDL including rules and other scenario elements on a semantic level is a challenging task and formal ontology with dedicated Description Logic reasoners can facilitate this task. Impacts and consequences are in fact a function of time, i.e. they change as time evolves. This issue could be addressed as well in future versions of CIPRTrainer. Reusability of several components in CIPRTrainer is a long-term goal, especially reusing the CM scenarios encoded in SDL, as a kind of European scenario database, and the federated simulation environment, as the proposed EISAC, for other similar research projects and even production environments.

10 Conclusion

This chapter presented a comprehensive description of CIPRTrainer, an innovative application designed for CM training. CIPRTrainer provides the novel capability of what-if analysis for exploring different courses of actions in complex simulated crisis scenarios involving CI. A trainee that uses CIPRTrainer can ‘go back in time’, revert a decision, and can choose a different course of action. This is possible in simulation, but not in reality. For comparing the consequences of the scenario evolution and assessing the outcomes of the chosen courses of action, CIPRTrainer uses Consequence Analysis methods. Federated simulation of CI provides information on disaster impacts like CI outages and resulting cascading effects.

The realisation of this new what-if analysis capability is based on several core technologies and innovative methods. CIPRTrainer’s core building blocks that are essential for providing the functionalities are:

- **Scenario management.** It includes the creation, conversion and execution of CI-centric CM scenarios with an emphasis on dependency modelling of CI. Scenarios are created within SyMo in an offline fashion. The results can be exported as SDL, which is the Scenario Description Language developed for modelling CM scenarios. Scenario Executor, which is part of the scenario management components, can import and execute SDL.
- **Declarative dependency handling with Complex Event Processing.** Cascading effects caused by sophisticated dependencies between CI are deduced by executing the declarative rules encoded in EPL—Event Processing Language. The open source event-processing engine Esper has been adopted to interpret the rules. It has been seamlessly integrated into the CIPRTrainer backend.

- ***Federated simulation system with models, simulators and connectors.*** It provides the technical basis for performing rollback and what-if Analysis, since in real-world context rollback of what has already happened is not possible. The simulation backend consists of three domain-specific CI simulators—SINCAL, ns-3 and OpenTrack—plus one threat simulator, the flooding simulator. For each simulator, a CIPRTrainer connector has been developed, enabling the RESTful communication with other components.
- ***Consequence Analysis.*** CM involves performing various kinds of user actions like initiate an evacuation under crisis situations, send a unit of first responders (police, fire fighters, medical emergency services, etc.) to a certain location, etc. The impacts and consequences of performing these actions are provided by this module. Technically, it is implemented inside of the database management systems to maximise the system performance.
- ***HTML5 Web front-end for interaction with system users.*** Advanced Web technologies are adopted to provide a pervasive user experience. This includes responsive design that enables an optimal ‘Look and Feel’ with different browser configuration and mobile devices. HTTP Push technology minimise the delay of event visualisation by avoiding constantly queries the CIPRTrainer backend. In addition, an internationalised user interface makes CIPRTrainer useful for cross-border scenarios.

From the technical point of view, all these components are loosely coupled with RESTful Web services with a high-level of scalability—in terms of both development productivity and system running performance.

Besides the technical implementation of the CIPRTrainer software system, there were two more major challenges in the design and realisation. One was the modelling activity for creating complex and realistic scenarios, and the other was designing the user interaction in a way that makes CIPRTrainer usable for training in a wider range of countries.

The modelling of the complex scenarios was a heterogeneous activity covering roughly three different aspects: (1) Static modelling for creating an ontology of elements to be considered, including resources, crisis management actions, threats, and more; (2) impact and damage modelling for consequence analysis; and (3) models of interconnected CI. For all these modelling activities, data needed to be acquired. When such data were missing, plausible artificial models, e.g. for electricity distribution networks, needed to be created with help from domain experts.

The model of the user interaction was guided by analysis of the crisis management governance structure in several European countries. Though these structure were sometimes vastly different, it is possible to identify common roles in several CM governances. These include decision-taking (or command), situational awareness, response leaders and leaders of administrative departments.

So far, CIPRTrainer has been demonstrated several times, and has also been used for two training exercises at the Master of Homeland Security study at Università Campus Bio-Medico di Roma and at the Fraunhofer campus in Sankt Augustin, Germany.

Acknowledgement and Disclaimer This chapter was derived from the FP7 project CIPRNet, which has received funding from the European Union's Seventh Framework Programme for research, technological development and demonstration under grant agreement no. 312450.

The contents of this chapter do not necessarily reflect the official opinion of the European Union. Responsibility for the information and views expressed herein lies entirely with the author(s).

References

1. Luijff E, Klaver MHA (2009) Insufficient situational awareness about critical infrastructures by emergency management. In: Symposium on C3I for crisis, emergency and consequence management IST-086/RSY-019, May 2009, Bucharest, Romania, NATO Research and Technology Organisation, RTO-MP-IST-086, paper 10. <http://www.rta.nato.int/pubs/rdp.asp?RDP=RTO-MP-IST-086>, 10 p
2. Klaver MHA, Luijff HAM, Nieuwenhuijs AN, Van Os N, Oskam V (2016) Critical infrastructure assessment by emergency management. In: Rome E, Theocharidou M, Wolthusen SD (eds) Critical information infrastructures security, 10th international workshop, CRITIS 2015, Berlin, lecture notes in computer science, vol 9578. Springer, Heidelberg, pp 79–90
3. Barrett C, Beckman R, Channakeshava K, Huang F, Kumar V, Marathe A, Marathe M, Pei G (2010) Cascading failures in multiple infrastructures: from transportation to communication network. In: 5th international conference on critical infrastructure (CRIS), pp 1–8
4. Petermann T, Bradke H, Lüllmann A, Poetzsch M, Riehm U (2010) Gefährdung und Verletzbarkeit moderner Gesellschaften – am Beispiel eines großräumigen Ausfalls der Stromversorgung. Endbericht zum TA-Projekt. Hg. v. TAB (141)
5. Luijff E, Klaver M (2005) Critical infrastructure awareness required by civil emergency planning. In: Proceedings of the 2005 first IEEE international workshop on critical infrastructure protection (IWCIP'05). IEEE, p 8f
6. Deutch D, Ives ZG, Milo T, Tannen V (2013) Caravan: provisioning for what-if analysis. In: CIDR, 2013
7. Lakshmanan LVS, Russakovsky A, Sashikanth V (2008) What-if OLAP queries with changing dimensions. In: IEEE 24th international conference on data engineering, 2008 (ICDE 2008), pp 1334–1336, April 2008
8. Golfarelli M, Rizzi S, Proli A (2006) Designing what-if analysis: towards a methodology. In: Proceedings of the 9th ACM international workshop on data warehousing and OLAP. ACM, pp 51–58
9. Chaudhuri S, Narasayya V (1998) Autoadmin “what-if” index analysis utility. In: Proceedings of the 1998 ACM SIGMOD international conference on management of data, SIGMOD'98, New York, NY, USA. ACM, pp 367–378
10. Haas PJ, Maglio PP, Selinger PG, Tan WC (2011) Data is dead... without what-if models. PVLDB 4(12):1486–1489
11. Cai Z, Vagena Z, Perez L, Arumugam S, Haas PJ, Jermaine C (2013) Simulation of database-valued markov chains using SimSQL. In: Proceedings of the 2013 ACM SIGMOD international conference on management of data, SIGMOD'13, New York, NY, USA. ACM, pp 637–648
12. Dalvi N, Ré C, Suciu D (2009) Probabilistic databases: diamonds in the dirt. Commun ACM 52(7):86–94
13. Huang J et al (2009) MayBMS: a probabilistic database management system. In: Proceedings of the 2009 ACM SIGMOD international conference on management of data. ACM
14. Rome E, Langeslag P, Usov A (2014) Federated modelling and simulation for critical infrastructure protection. In: D'Agostino G, Scala A (eds) Network of networks: the last frontier of complexity. Springer, Cham, Heidelberg

15. Xie J, Theocharidou M, Barbarin, Y, Rome E (2016) Knowledge-driven scenario development for critical infrastructure protection. In: Rome E, Theocharidou M, Wolthusen SD (eds) *Critical information infrastructures security*, 10th international workshop, CRITIS 2015, Berlin. Lecture notes in computer science, vol 9578. Springer, Heidelberg, pp 91–102
16. EU FP7 CIPRNet, CEA, Deliverable D6.2—Application Scenario. Gramat, France, 2014
17. EU FP7 CIPRNet, Fraunhofer, Deliverable D6.3—Federate CI Models. Sankt Augustin, Germany, 2015
18. EU FP7 CIPRNet, Fraunhofer, Deliverable D6.4—Implementation and integration of the federated and distributed cross-sector and threat simulator. Sankt Augustin, Germany, 2016
19. Sojeva B (2015) Using Web GIS for designing added-value training systems for crisis managers. Master's thesis, University Koblenz-Landau, Germany
20. Nieuwenhuijs A, Luijff E, Klaver M (2008) Modeling dependencies in critical infrastructures. In: Goetz E, Shenoi S (eds) *Critical infrastructure protection*, IFIP Series, vol 253, pp 205–214
21. Rinaldi S, Peerenboom J, Kelly T (2001) Identifying, understanding and analyzing critical infrastructure interdependencies. *IEEE Control Syst Mag* 21(6):11–25
22. Rinaldi S (2004) Modeling and simulating critical infrastructures and their interdependencies. In: 37th Hawaii international conference on system sciences, vol 2, USA. IEEE
23. The SIEMENS PSS® SINCAL Home Page <http://www.sincal.de> (last access on 16/07/2015)
24. ns-3 home page <http://www.nsnam.org> (last access on 16/07/2015)
25. The OpenTrack Home Page <http://www.opentrack.ch> (last access on 16/07/2015)
26. European Commission (EUROSTAT) (2014) The census hub: easy and flexible access to European census data. doi:10.2785/52653. ISBN 978-92-79-37803-4. <http://ec.europa.eu/eurostat/de/web/population-and-housing-census/census-data/2011-census>
27. CORINE Land Cover (CLC2006); Federal Environment Agency, DLR-DFD 2009
28. Jonkman SN, Lentz A, Vrijling JK (2010) A general approach for the estimation of loss of life due to natural and technological disasters. *Reliab Eng Syst Saf* 95(11):1123–1133. doi:10.1016/j.res.2010.06.019
29. Freire S, Aubrecht C (2012) Integrating population dynamics into mapping human exposure to seismic hazard. *Nat Hazards Earth Syst Sci* 12(11):3533–3543
30. Leung S, Martin D, Cockings S (2010) Linking UK public geospatial data to build 24/7 space-time specific population surface models. In: *GIScience 2010: sixth international conference on geographic information science*. University of Zürich, Zurich
31. Polese M et al (2014) CRISMA. Version 2 of dynamic vulnerability functions, systemic vulnerability, and social vulnerability. Deliverable D4.32, CRISMA-Project
32. Kok M, Huizinga H, Vrouwenvelder A, Barendregt A (2005) Standard method 2004 damage and casualties caused by flooding. DWW-2005-009. Ministerie van Verkeer en Waterstaat
33. Thieken AH (ed) (2010) *Hochwasserschäden. Erfassung, Abschätzung und Vermeidung*. Oekom, München
34. Hallegatte S (2014) *Natural disasters and climate change. An economic perspective*, Chapter 2. Springer, Berlin
35. Okuyama Y (2007) Economic modeling for disaster impact analysis: past, present, and future. *Econ Syst Res* 19(2):115–124. doi:10.1080/09535310701328435
36. Rose A (1995) Input-output economics and computable general equilibrium models. *Struct Change Econ Dyn* 6:295–304
37. Lian C, Haimes YY (2006) Managing the risk of terrorism to interdependent infrastructure systems through the dynamic inoperability input–output model. *Syst Eng* 9(3):241–258. doi:10.1002/sys.20051
38. Santos JR, Haimes YY (2004) Modeling the demand reduction input-output (I-O) inoperability due to terrorism of interconnected infrastructures. *Risk Anal* 24(6):1437–1451. doi:10.1111/j.0272-4332.2004.00540.x

39. Smith BJ, Vugrinm ED, Loose VW, Warren DE, Vargas VN (2010) An input-output procedure for calculating economy-wide economic impacts in supply chains using homeland security consequence analysis tools. In: Proposed for presentation at the North American regional science council 57th annual North American meetings of the regional science held 10–13 Nov 2010. Sandia National Laboratories
40. Karsten A (2014) Führen durch die Chaos-Phase. Bevölkerungsschutz 4(2014). Themenheft “Kritische Infrastrukturen”. BBK, Bonn, pp 32–35. ISSN 0940-7154

Open Access This chapter is licensed under the terms of the Creative Commons Attribution 4.0 International License (<http://creativecommons.org/licenses/by/4.0/>), which permits use, sharing, adaptation, distribution and reproduction in any medium or format, as long as you give appropriate credit to the original author(s) and the source, provide a link to the Creative Commons license and indicate if changes were made.

The images or other third party material in this chapter are included in the chapter’s Creative Commons license, unless indicated otherwise in a credit line to the material. If material is not included in the chapter’s Creative Commons license and your intended use is not permitted by statutory regulation or exceeds the permitted use, you will need to obtain permission directly from the copyright holder.

