Chapter 11
Federated Modelling and Simulation for Critical Infrastructure Protection

Erich Rome, Peter Langeslag and Andrij Usov

Abstract Modelling and simulation is an important tool for Critical Infrastructure (CI) dependency analysis, for testing methods for risk reduction, and as well for the evaluation of past failures. Moreover, interaction of such simulations with external threat models, e.g., a river flood model, or economic models enable consequence analysis and thus may assist in what-if decision-making processes. The simulation of complex scenarios involving several different CI sectors requires the usage of heterogeneous federated simulations of CIs. However, common standards for modelling and interoperability of such federated CI simulations are missing. Also, creating the required abstract models from CIs and other data, setting up the individual federate simulators and integrating all subsystems is a time-consuming and complicated task that requires substantial know-how and resources. In this chapter, we outline applications and benefit of federated modelling, simulation and analysis (MS&A) for Critical Infrastructure Protection (CIP). We review the state of the art in federated MS&A for CIP and categorise common approaches and interoperability concepts like central and lateral coupling of simulators. As examples for the latter two concepts, we will present in more detail an interoperability standard from the military domain, HLA, and an approach developed in the DIESIS project. Special emphasis will also be put on describing the problem of synchronising systems with different time models. Also, we will briefly assess the state of transferring MS&A for CIP

E. Rome (✉) · A. Usov
Fraunhofer IAIS, Schloss Birlinghoven, 53754 Sankt Augustin, Germany
e-mail: erich.rome@iais.fraunhofer.de
http://www.iais.fraunhofer.de

A. Usov
e-mail: andrij.usov@iais.fraunhofer.de
http://www.iais.fraunhofer.de

P. Langeslag
TNO Defence, Security and Safety, Oude Waalsdorperweg 63, The Hague, The Netherlands
e-mail: peter.langeslag@tno.nl
http://www.tno.nl

G. D'Agostino and A. Scala (eds.), *Networks of Networks: The Last Frontier of Complexity*, Understanding Complex Systems, DOI: 10.1007/978-3-319-03518-5_11,

research results to practical application by comparing the situations in the USA and in Europe.

Keywords Federated simulation · Modelling · Analysis · Interoperability · Critical infrastructures · HAL · DIESIS · OpenMI · XMSF · IDSim · I2Sim · Simulation · Time synchronisation

11.1 Introduction

Infrastructures operate globally and are increasingly dependent and interdependent: a breakdown or disruption of functions may have serious national or even multinational consequences [1]. The disruptions of the power grids in 11 European states and Morocco on November 4, 2007 affecting 15 million people are a case in point. It is for this reason that such infrastructures can be called Critical Infrastructures (CI). A CI is defined by [2] as an asset, system or part thereof that is essential for the maintenance of vital societal functions, health, safety, security, economy or social well-being of people, and the disruption or destruction of which would have a significant impact as a result of the failure to maintain those functions. Clearly, it is mandatory to protect these assets. In this respect, modelling, federated simulation, and analysis are of vital importance [3]. They are required for the investigation of CI and their dependencies, for training CI operators and crisis managers, as well as for the development of methods for Critical Infrastructure Protection (CIP).

Simulation is particularly well-suited for capturing dynamic effects within the complex system of interconnected critical infrastructure systems, like cascading effects. A failure or loss of service in one infrastructure, like power transmission, may cause a loss in a dependent infrastructure, like railway transport. The simulation and investigation of large scenarios with cascading CI failures affecting critical services in multiple CI requires the use of **federated** simulations consisting of simulators and suitable models[1] for each of the involved CI [3, 4]. In addition, models that may generate threats to the CI and models that analyse the consequences (e.g., economic loss, number of casualties, affected area), and visualisation as well as other real-time tools may need to take part in the federation. These simulators and components need to be coupled by means of a suitable middleware that allows the synchronisation of events and simulation times, the exchange of data, and the exertion of control functions like starting and stopping simulations (Fig. 11.1). That is, components of a federated simulation need to be **interoperable**. Different from the situation in most military simulations, current CI simulators are in general not interoperable and often lack proper interfaces.

Some federated simulations need to be realised as distributed systems, for various reasons. Depending on the computational demands of individual simulators that

[1] In this publication, we refer to 'simulation' as the dynamic part of a computer model, and we refer to 'model' as the static part of a computer model.

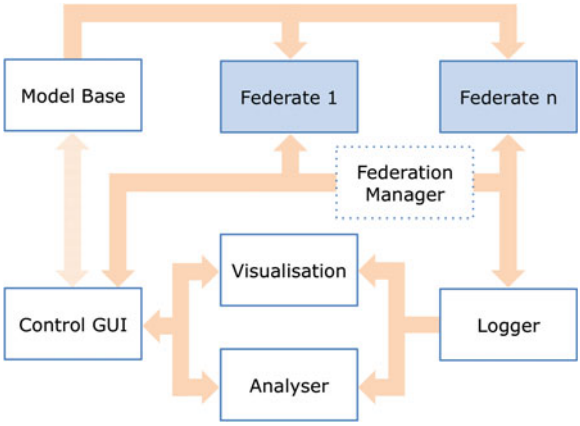

Fig. 11.1 A generic schema for modelling, federated simulation and analysis for CI. A model base contains abstract (conceptual) models for each of the CI simulators (federates). Simulation is started by a control GUI. Federates exchange data with each other and report status to control GUI. Intermediate states and results of the federates are logged by a data logger. A visualisation module can be employed for visualising results and simulation states online (while the simulation is running) or offline (after the simulation terminated). Analysis tools can provide additional information by evaluating the results online or offline. Sometimes, one of the federate components serves as an orchestrator or manager that controls the simulation steps of all individual simulators

comprise the federation, it is sometimes not commendable or possible to run the entire federation on a single computer, e.g. when federates require different operating systems. And lastly, in a collaborative research project individual simulators composing a federation may need to run on different partners' computers for license or financial reasons. **Distributed (federated) simulation** includes running a federation on a local computer grid or in a locally distributed fashion via the Internet or other communication infrastructure.

After the above brief characterisation of federated simulation systems, we will describe applications, technologies, state of the art, challenges, and standardisation of federated modelling, simulation and analysis (MS&A) in more detail in the remainder of this chapter. It is organised as follows. First, we describe application areas of federated MS&A. Then we take a look at the basic technical properties of federated simulations and describe current interoperability approaches and the synchronisation problem. Then we review the state of the art in federated MS&A for Critical Infrastructure Protection and describe in detail two different interoperability approaches, HLA and DIESIS. We conclude with summarising the main insights.

11.2 Applications of Federated Modelling, Simulation and Analysis

Federated modelling, simulation and analysis has a wide range of applications, with two foci, research—as aid for and subject of—on one hand and applications in security on the other hand. Both foci are closely related. The US American facility NISAC employs federated MS&A for homeland security. It emerged from a cooperation between two research institutes, the Sandia and Los Alamos National Laboratories in the year 2000 and is now part of the Department of Homeland Security. The research institutes successfully managed transferring their CIP expertise and federated MS&A technology to practical applications. In Europe, federated MS&A for CIP is still rather a subject of research, but there is a small community working towards a European facility, comparable to NISAC, that shall provide expertise and technology to offices, institutions, and people responsible for Critical Infrastructure Protection and Civil Security.

In this section, we will present both typical applications of federated MS&A and outline emerging applications. It should be noted here that developments of federated MS&A were and are strongly driven from applications in the military domain, for instance, the HLA middleware standard. That is, some of the state of the art presented here stems from the military domain, but is relevant also for CIP.

11.2.1 Investigating Dependencies Between Critical Infrastructures

For this chapter, we adopt the definition of dependent and interdependent CI given by Luiijf et al. [5, p. 304], "A CI dependency is the relationship between two CI products or services in which one product or service is required for the generation of the other product or service; a CI interdependency is a mutual CI dependency." A loss of service or reduction of quality of a service in one infrastructure may lead to loss of service in a dependent second infrastructure, this again may lead to loss of service in a dependent third infrastructure, and so on. This is called a cascading effect. Since CI form a very complex system of dependent systems at various scales (local, regional, national, continental, global), it is difficult to understand the nature and effects of these dependencies. A prerequisite of developing methods and measures for protecting CI or making them more resilient was an improved understanding of the dependencies. Consequently, this type of investigation was a major focus in the CIP research area over the last 10 years. While some dependencies are of static nature (by construction or local neighbourhood of CI elements), the more difficult cases are dynamic dependencies, occurring while CI are operating. Since it is prohibitive to use real CI for investigation, researchers started using conceptual models and computer simulation of CI for investigating dependencies and interdependencies of CI (Examples: [3, 4, 6–17]). An example of a dynamic effect is a delayed cascading effect, like a loss of power supply for a hospital that has a diesel generator as a backup

power supply, which fails several hours later after it runs out of fuel. Depending on the duration of the power outage, this cascading effect may or may not happen.

11.2.2 Soft Exercises and Training

One of the essential elements for protecting Critical Infrastructures is maintaining or achieving a high level of preparedness of staff responsible for security, like crisis and emergency managers. This requires practice in real emergencies and crises as well as training of simulated emergencies and crises in exercises. Typically, national or regional exercises take place once a year, while some enterprises do monthly exercises. Such exercises are necessary and useful. However, given the wide range of potential scenarios, annual practical exercises seem not sufficient for being prepared for even the most likely scenarios in an ever-changing world. Computer simulation would be suited as an additional means for exercising mitigation of crises and emergencies, and raising awareness of the role of CI in crises and emergency situation [18]. Similarly, federated M&S of CI could be used to train operators of CI for mitigating crisis and emergency situations. Here, scenarios and scripts of the simulations could be altered to cover a wide range of possible situations.

11.2.3 Decision Support

In cases of crises, crisis and emergency managers may encounter situations in which different courses of action are possible. Decision Support Systems (DSS) provide methods for assessing the consequences of certain decisions and thus may aid crisis and emergency managers in taking the right decisions. By using simulations, DSS can be enhanced to perform 'what if' analyses, that is, dynamically explore the different courses of action and their different consequences. In this way, these end users are enabled to plan the most effective use of resources in an emergency and to explore a variety of scenarios, for example:

- which region to evacuate first, which infrastructures to reinforce best/first,
- which transport or traffic infrastructures required for a mitigation plan will be affected by a disaster and what contingency planning is required,
- which infrastructures outside a region affected by a disaster need to be operational in order to supply that region and thus need to be protected too.

Examples of this type of application are I2Sim [19], described later in this chapter, and the work of Tolone et al. [15].

11.2.4 Environment for Testing and Benchmarking New Methods

The IRRIIS project ([20], cf. also below) developed a federated simulation of the interdependent power distribution and telecommunication networks of two

infrastructures operators. The simulation was orchestrated by means of an agent-based simulator called SimCIP [21]. During the simulation, early indicators for reductions of quality of service or loss of service were computed independently for each of the four simulated infrastructure topologies. The project investigated whether a risk reduction could be achieved by communicating the early risk indicators between the two infrastructure operators [22]. A potential future extension of this type of application, proposed in [23], is using federated MS&A for benchmarking competing new methods for risk reduction in or protection of CI.

11.3 Basic Technical Aspects of Federated Simulation

11.3.1 Interoperability Approaches

The assessment of simulator interoperability provided by Usov et al. still holds today: "In recent years, a large number of projects have investigated and tested methods for coupling simulators. As a general result it can be stated that the technological task of coupling simulators is highly demanding and that there are no ideal general purpose solutions for the coupling task, but the applied methods are strongly determined by the general requirements and the application task at hand" [24]. In this section, we will review some basic interoperability approaches.

A connection creates a communication link between two or more systems. Interoperability between these systems could be considered from a double point of view [25]: *technical connectivity* and *semantic connectivity*.

Technical connectivity considers in which way systems are able to solve the problem of sharing and exchanging data across multiple platforms. It is strongly related to the capability of systems to implement a common data structure and syntax in order to achieve a connection among them. This aspect of interoperability implies that the exchanged information is understandable by any other system not initially developed for the cooperation. So a common language is an essential requirement. It enables the description of the structure and syntax of the underlying data.

To achieve a meaningful connection among systems it is necessary to establish how they can exchange information or in which way they combine other information about resources and subsequently the way to process them, in a significant manner. Thus, **semantic connectivity** requires agreement on a wide variety of issues relating to the context within information is created and used. The aim is not only to allow information resources to be linked up, but also to give context to information in a scenario in which different systems have their own perspective on that. Only in this way information can be automatically understandable and, consequently, reusable by systems that were not involved in its creation. Thus, the semantic connectivity concerns the need to agree on common definitions and to understand information that is necessary to exchange.

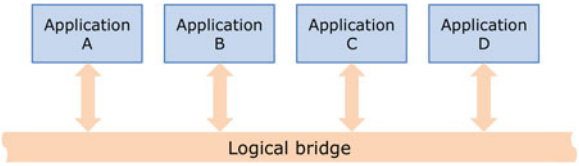

Fig. 11.2 Central coupling: Connected systems exchange information via a common logical bridge, using a standardised exchange format

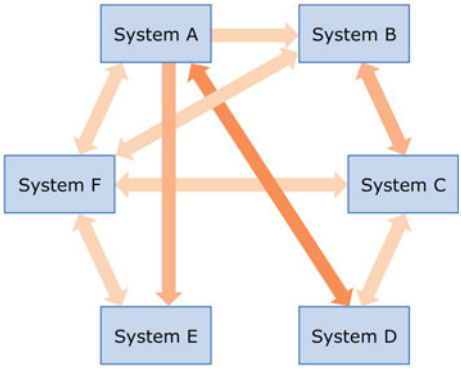

Fig. 11.3 Lateral coupling: Connected systems exchange information bi- or multi-laterally. Exchange formats may vary

For connecting simulations, two technologies can be considered: central coupling and lateral coupling. The **central coupling** topology is of a typical architecture oriented to the distribution of services, where the applications communicate each other through a logical channel or bridge (Fig. 11.2). From an architectural point of view, such logical channel is based on a software layer centralising functionalities. It supports synchronous and asynchronous communication based on messages and intelligent routing as well as data transformation and connectivity towards heterogeneous applications. Typically, such a logical channel is implemented by means of middleware, usually based on standards. It supplies the fundamental services for more complex architectures through event-driven and message passing mechanisms. Central coupling approaches with standardised exchange formats, such as HLA [26–29] or OpenMI [30, 31], are particularly suited when all federates support the exchange format standard. If this is the case, the integration of a new simulator is relatively easy and is limited to the implementation of an interface between this simulator and interoperability middleware which already contains ready-made solutions for communication, time management, etc.

Despite of its convenience, central coupling approaches are generally not applicable if federation members do not use one and the same interoperability standard. Especially the time management is often a challenging problem that has only one-off solutions for particular federations but is unsolvable in general, for arbitrary

combinations of models and technologies (see Sect. 11.3.2). Other than simulators in the military area, federations of CI simulations are often quite heterogeneous in terms of interfaces, modelling approaches and even time scales. For realisation of such a federation, a **lateral coupling** topology [24] is recommended. This architecture foresees the development of dedicated links between pairs of federates, according to their logical interconnections (Fig. 11.3). Besides of pairwise couplings, the resulting federation may also contain centrally coupled clusters of simulators that all support a certain interoperability standard. A drawback of the lateral coupling approach is obviously the large number of links that has to be developed for creating a federation as well as for adding a new member to an existing one. Creating potentially reusable links and storing them in a kind of repository is a possible solution for this problem. Application of lateral coupling architecture for federations of CI simulators and corresponding strategies are discussed in Sect. 11.4.5

11.3.2 Time Models and Synchronisation

The semantic connectivity described in the previous section has two essential aspects: regulation of data exchange between federates and correct interpretation of the exchanged data. Here, we will discuss the former aspect, and the latter one will be handled in Sect. 11.3.3. Regulation of data exchange means, in the first line, the preservation of causality. In other words, it is necessary to ensure that particular *events* (i.e., discrete messages that usually represent state changes) are processed by all federates in a logically correct order. This is an essential requirement for the reproducibility of simulation results for the same scenario, which is required for many evaluation techniques. For training applications, minor deviations from the correct event order can often be tolerated as long as perceived realism of simulation results is not violated [32].

The problem of time management for federated simulations is as old as the idea of federated simulation itself. First formulations of this problem and corresponding first solutions were published in the late 1970s (e.g., by Chandy and Misra [33]). However, since then, many new approaches or variations of already existing approaches have been developed and published. The reason for this continuing interest in the time management problem is that there is no universal optimal solution for all possible simulators with their different time and execution models. Irrespective of their *internal* time representations, the simulators may offer interaction capabilities and provide simulation results according to the following schemes:

- **Steady state:** a simulation runs until a steady state of the internal model (or some state where reasonable simulation results can be produced) is reached. This kind of computation is often employed for relatively fast processes and sometimes they even have no notion of time in their models (e.g., switching actions in a power distribution network).

- **Discrete events:** simulation results are produced after variable simulation time intervals that are determined by occurrence of some internal event or by reaching some state in which the simulator assumes that interaction with its environment is required.
- **Constant time steps:** simulators of this type are clocked and offer interaction after every constant simulation time period. Simulation steps are usually configurable: shorter steps increase the accuracy of results, longer steps improve the simulation performance.
- **Real time:** those simulations are usually employed for training with human in the loop. They often work with small constant time steps that are synchronised with the real time. Hence, their capability to wait for simulation results produced by other federates is very limited. Interaction with real time simulators places high demands on performance of other federates and of the middleware.

The choice of an appropriate time management approach depends on the combination of different time models within a federation. Another factor that determines the synchronisation strategy is the desired usage of simulation results: for exact results evaluation the synchronisation algorithm has to work exactly too. Finally, the ability of simulators to return to a time point in the past by rolling back recent state changes allows to employ advanced synchronisation techniques. Most time management implementations are based either on *conservative* or on *optimistic* synchronisation approaches. In this section, we will provide only a brief introduction into the core ideas of these techniques. More detailed description and an overview over related algorithms can be found, for example, in works by Fujimoto [32] and Riley et al. [34].

Conservative time management algorithms prevent causality violations by restricting event processing to "safe" events. An event can be safely processed by a particular federate only if it is absolutely certain that no other events assigned to an earlier simulation time point will be received by this simulator afterwards. An essential prerequisite for the application of this approach is the ability to compute a *lookahead* (i.e., remaining simulation time until the next "public" event) for all federates. Several solutions were developed in order to detect and avoid deadlocks as well as to minimise the communication overhead [35, 36] of conservative synchronisation algorithms.

Optimistic synchronisation approaches use another strategy for keeping the event order correct. Federates are explicitly allowed to process potentially "unsafe" events. However, if an event with smaller simulation time stamps arrives later, simulators must be prepared to roll back all effects of recently processed events and to reprocess events in a correct order. Optimistic synchronisation provides a higher degree of parallelism and potentially better simulation performance. However, its major drawback is a high demand on memory that is required for storing checkpoints for possible rollbacks. Furthermore, simulations may be significantly slowed down by an increasing number of rollbacks which is highly scenario-dependent. This problem is even exacerbated by the fact that processing an event may produce new events that have to be sent to other simulators. In this case, the rollback process includes the cancellation

of sent events and, hence, it initiates rollbacks at other federates. A "too optimistic" event processing strategy may unleash extremely time-consuming rollback cascades.

Unlike conservative approaches, the optimistic ones do not rely on the computation of lookaheads. On the other hand, the ability of federates to maintain checkpoints and to perform rollbacks is required. The Time Warp algorithm published by Jefferson in 1985 [37] was probably the first optimistic synchronisation approach. It foresees "anti-messages" for cancelling already sent events in case of a non-local rollback. In the early 90s, several modifications of the Time Warp algorithm as well as completely new ideas were developed in order to decrease memory consumption [38] and to avoid costly distributed rollbacks [39, 40].

Neither conservative nor optimistic time management algorithms can provide a universally optimal solution for arbitrary federations due to the fact that the efficiency in both approaches highly depends on scenarios (global event order), on federation topology (logical links among simulators) as well as on specific features supported by particular simulators (lookahead computation and rollback functionality). In the area of CIP, it is often required to simulate interactions of different infrastructures that are described by quite different physical and logical laws. The resulting federation can be extremely heterogeneous and may contain simulators that internally work with *completely* different time scales. Typical temporal intervals between events are milliseconds for power network and communication domains, seconds for urban traffic, minutes for evacuation and smoke propagation and hours for flooding simulations. Some simulators may not support rollbacks, since others may be unable to forecast their lookaheads. Hence, the choice of an appropriate time management solution for a federation of CI simulators is determined by the composition of this federation. Furthermore, there is no guarantee that a globally applicable solution exists. In this case, according to the idea of lateral interoperability, different time management approaches have to be employed for particular pairs or clusters of simulators within the federation (see Sect. 11.3.1).

11.3.3 Modelling for Federated Simulation

As stated in the previous section, the correct interpretation of the exchanged data is of utmost importance for a federation to become more than the sum of its federates. Also, this data has to be interpreted in the right manner. Another issue that rises is that the federates are created and validated for a special purpose. Using them as part of a federation may cause the validation to become insufficient and thus it may become necessary to redo the validation of the federates. In addition, the overall validation needs to be regarded. This chapter will handle both cases.

The easiest way to guarantee the information exchange to contain the correct data is to anchor this in the data exchange protocol, as it is done with DIS [41]. This can easily be accomplished with simulators in the same domain where linking simulators is common knowledge (DIS was developed for the defence domain). For the world of CI where multiple domains are involved and linking of simulators is not common, it

is unlikely that simulator developers will adapt to a common data exchange protocol. Also, a disadvantage of DIS is the use of broadcast for data exchange. All information is sent to every federate, which can result in a big bandwidth consumption.

A more flexible approach is used in HLA with the use of Object Models (OM) [42]. HLA object models are composed of a group of interrelated components specifying information about classes of objects, their attributes, and their interactions. Every federate has its own federate object model (FOM), but it has to be compliant with the object model of the federation, the simulation object model (SOM). HLA uses a publish and subscribe mechanism for data exchange. Therefore, network bandwidth can be adapted to actual needs.

Although the HLA method gives a much more flexible approach for linking simulators compared to DIS, it is still based on a common data model, which must be implemented by all federates. Also, it is a purely syntactic model which works well for use within one domain, but it does not provide the semantic information about the modelled domains needed in a multi domain environment. Masucci et al. [43] describe several modelling and simulation approaches to analyse critical infrastructure interdependencies and conclude with an ontology based modelling and simulation solution called the DIESIS KBS architecture. The DIESIS KBS design incorporates a meta knowledge "world" infrastructure ontology (WONT), infrastructure ontologies (IONTs), a federation ontology (FONT) and gateway components (see Sect. 11.4.5).

As Masucci et al. [43] describe, the DIESIS KBS is designed for creating abstractions of critical infrastructure domains and to represent and formalize their parameters and dependencies. The KBS is intended to be used in a federated simulation environment to study the behaviour of infrastructures and their components under different conditions and constraints. The resulting federated environment will support complex simulation scenarios involving multiple infrastructures with different semantics and granularities (or fidelities).

Making all the federates in the federation understand each other's data solves only half the problem. Also, the way they use the data and the level to which extend the model describes the real world domain should be taken in account because these end up in defining the credibility or the final outcome of the federation. For this reason, verification and validation (V&V) of the federation should be performed from the early begin of the process up to the end. SISO [44] provides a generic methodology for verification and validation to support acceptance of models, simulations and data. The objective of the V&V effort is to develop an acceptance recommendation that convincingly shows why a federate or federation is acceptable or not acceptable for the intended use. This V&V objective is articulated as an acceptance goal. This high-level goal should be translated into a set of concrete and assessable acceptability criteria for the federate or federation. Relevant and convincing evidence should then be collected or generated to assess the satisfaction of these criteria. When it is convincingly demonstrated to what extent the federate or federation does or does not satisfy all these acceptability criteria, a claim can be made on whether or not the federate or federation is acceptable for its intended use (i.e., acceptance claim).

11.4 State of the Art in Federated MS&A for CIP

In the last decade, the awareness has grown that Critical Infrastructures are in a greater or lesser extent dependent on each other. Investigating, exercising and training of CI behaviour in case of an event can not be done by one system alone. Therefore the need for combining the interdependent systems in a simulation environment has grown. In these past few years, several initiatives have been taken to combine (parts of) the different simulations for critical infrastructures in or across different domains. This chapter describes many of these efforts. The characterised works within this section 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, ...).

More elaborate presentations will be given for one example of a framework for central coupling (HLA, the High Level Architecture standard) and for one example of lateral coupling (the DIESIS approach). Older publications that include overviews on the state of the art in MS&A for CIP are [6, 45].

11.4.1 Special Purpose Federated Simulation Systems

With its SimCIP [21] modelling and simulation environment, the **IRRIIS** [20] project used an agent-based environment where components, subsystems and systems are represented by autonomous agents and the simulation is synchronised through a centralistic RTI-like simulation engine, LAMPS. Within the IRRIIS demonstrator, SimCIP orchestrates a federation of the SINCAL power transmission simulator and the NS2 network communication simulator, modelling the dependencies between power distribution and communication networks in a large European capital. The target application of IRRIIS was twofold, namely investigating dependencies and interdependencies, and risk reduction by communicating early risk indicators between operators of the two infrastructures [22]. The limitation of SimCIP though is that no standardised definition or workflow is proposed for the extension of the environment through new simulators.

The **EPOCHS** approach [46] was driven by the need to better understand the effects of integrating network communication systems into electric power control systems on the stability of the electric power systems. The task required the

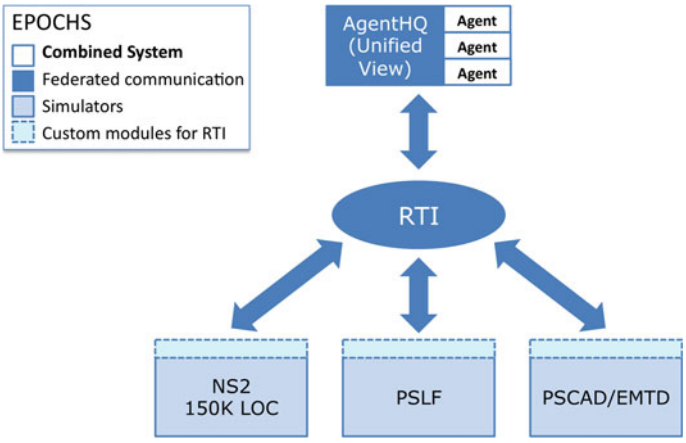

Fig. 11.4 Federation architecture of EPOCHS (after [46])

combination of three different high-fidelity simulation systems: A simulator for electromagnetic transient simulation (PSCAD/EMTDC), a simulator for electromechanical transient simulation (PSLF), and a network communication simulator (NS2 [47]). Each of these three simulators was designed as stand-alone simulator, that is, all three simulators lack built-in interoperability. Hopkinson et al. [46] implemented a specific solution for creating a federation of the three chosen simulators. It consists of three basic elements: a Run-Time Infrastructure for enabling time management and communication between the three federate simulators, an agent-based control interface for the user of the federated simulation, and individual extensions of the three simulators enabling the compatibility with the EPOCHS RTI in order to make them interoperable. The resulting architecture is shown in Fig. 11.4.

For enabling the compatibility with the EPOCHS RTI, the designers chose three different ways, based on the properties of the three different simulators [46, p. 5]. For the communication network simulator NS2, they used the fact that source code was available and added a new transport protocol for realising RTI access. The power simulator PSCAD/EMTDC allows external function calls. The EPOCHS designers used this feature for creating an external component that gets active at each time step of the simulator, reading and/or writing equipment values from/to the RTI before the simulation continues. For the power simulator PSLF, they used a similar solution. PSLF does not allow calling simulator functions, but allows writing extensions in its own programming language EPCL. The EPOCHS designers programmed a communication stub in EPCL that writes or reads simulation values to or from a file upon request from the RTI. The EPOCHS approach is an example of central coupling with non-standardised interfaces.

The motivation of Riley et al. [34] is the usage of simulation as a tool for analysis of communication network problems and validation of models of communication networks. For this task, the computer simulation of a communication network

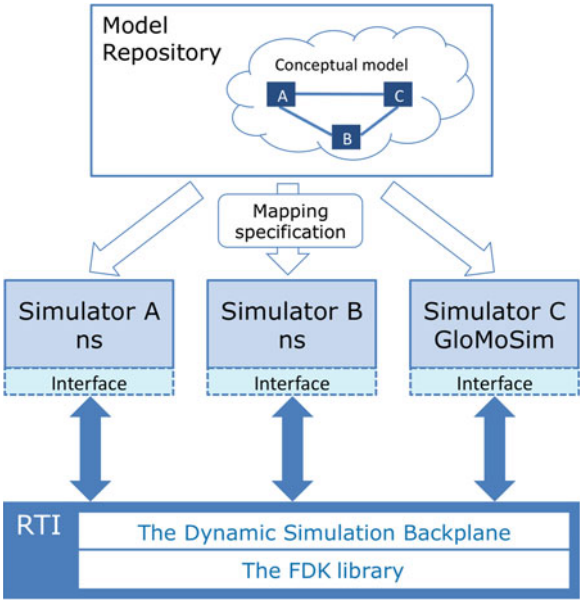

Fig. 11.5 Architecture for a high fidelity distributed federated communication network simulation, self-federating two instances of the Network Simulator ns (after [34])

requires a high degree of fidelity, which leads to high computational demand: "It would take several days to simulate just one minute of operation of this network" [34, p. 118]. Riley et al. propose a twofold solution that is suited to reduce the high ratio of simulation time versus real time. For the first part, they propose to distribute the simulation across multiple networked processors, which results in faster simulation. For the second part, they propose to setup the distributed simulation as a federation of several simulators, which allows simulating larger networks. In this case, the federates could also be several instances of the same simulator system (self-federated). Figure 11.5 shows the architecture concept that Riley et al. used for realising their distributed federated communication network simulation. It fits the generic scheme presented earlier in this chapter.

The Run-Time Infrastructure performs synchronisation and data distribution and uses a library called the Federated Simulations Development Kit (FDK library). A Dynamic Simulation Backplane ensures syntactic compatibility for data exchange between federates where possible, enables the exchange of meaningful event messages, checks for incompatibilities and provides more functions for communication between federates.

As an interesting experimental result, Riley et al. [34, p. 146] report that for two network simulators, PDNS and GTNets, they were able to show that self-federating each of these simulators enabled simulation of large network topologies (almost 2 million nodes) with "linear efficiency" up to 128 federates.

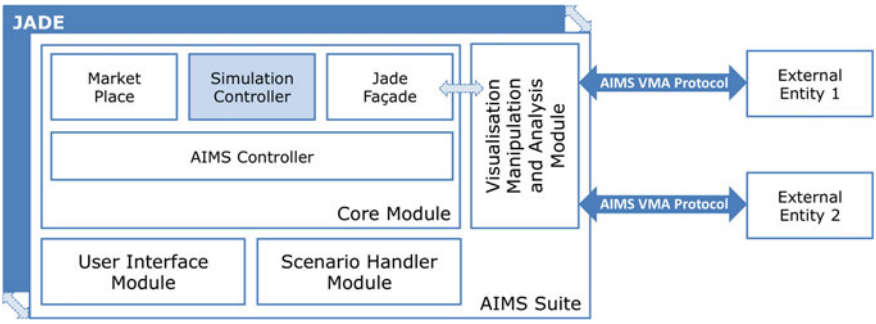

Fig. 11.6 AIMS modelling and simulation framework (after [49]). The agent-based CI simulation is executed in JADE (Java Agent Development Framework)

11.4.2 Frameworks for Modelling and Simulation of CI Using Dedicated Simulations

Bagheri, Ghorbani et al. developed the Agent-based Interdependency Modelling and Simulation **(AIMS)** suite [48, 49] especially for investigating (inter)dependencies between CI. Instead of employing off-the-shelf simulators, the authors used their own agent-based simulation, the AIMS simulator. Figure 11.6 shows the modules of the AIMS suite. They form a federation in the sense of our Fig. 11.1, but with only one dedicated agent-based simulator. For Bagheri et al., the analysis and visualisation part of MS&A were of particular importance, both on-line and off-line. They created a special middleware, the VMA (Visualisation, Manipulation and Analysis) entity protocol that allows integrating external entities, that is, analysis software modules, advanced visualisation tools, and scenario editors. In the on-line case, the latter entities enable users of the AIMS suite changing the models or scenarios while the simulation is running. All external entities need to conform to the VMA entity protocol in order to be integrated into the federation.

In [49], the authors describe their approach in detail. Also, they report the results of a case study of an electronic service provider that is dependent on two electricity suppliers and an Internet service provider (ISP). They use four scenarios in which they dynamically change the quality or availability of services of the electricity suppliers and the ISP in order to find out about the economical impact upon the electronic service provider. Finally, it is worth mentioning that Bagheri et al. are one of the few author teams that describe the workflow for setting up a federation (an instance of the AIMS suite, in this case). Since setting up federations requires considerable time and special know-how, this is valuable information.

I2Sim [19] provides a framework for discrete abstract modelling of dependent infrastructures from scratch. The I2Sim modelling ontology consists of ‘cells’, ‘channels’, ‘tokens’ and ‘controls’. A cell is a functional or production unit, like a hospital, a power station etc, that requires some input and produces some output. The behaviour of an infrastructure is described as human readable table (HRT), created

by experts (typically, the operators of the infrastructure). For an electrical power infrastructure, such a HRT specifies the available output power for a certain number of infrastructure operating states. Each operating state of the infrastructure—or the cell representing it—is described by a number of input-output relations.

Channels connect outputs of one (source) cell to inputs of other (consumption) cells and transport tokens from the source to the production cell. A channel may have a loss coefficient that characterises the percentage of tokens that get lost during transport (like loss of voltage due to resistance of a power line). After specifying all cells and their input-output relations, all channels and their loss coefficients, I2Sim sets up a mathematical model of the entire system. For the simulation part, I2Sim performs time-driven discrete event simulation [50]. Events occur at each time step and trigger the recalculation of the coefficients describing all cells' operating status. At each time step, the simulated system is described by a set discrete time equations [50], represented by a 'system transportation matrix'. This representation allows identifying strong dependencies or critical vulnerabilities. As a test case study, the I2Sim team modelled and simulated the campus of the University of British Columbia at Vancouver, which has the size of a small city. The model took into account buildings and water, gas, power and road networks of the campus. The test case system performed damage assessment in case of disaster, expressed in the number of casualties, economical losses and loss of campus functions. And finally, the test case system provided advanced decision support capabilities, including what-if analyses for first responders.

The I2Sim approach has at least three advantages: It allows modelling at different levels of abstraction, it preserves the privacy of the contributing infrastructures by not requiring revealing lots of technical detail, and it simultaneously reduces the required infrastructure domain expertise of the modelling experts. Small drawbacks are that the fidelity of the modelling is limited and that the infrastructure behaviour description cannot be validated by the modelling experts: They need to trust the domain experts.

To conclude this section, we want to mention that Tolone et al. [15, 51, 52] used their Integrated Modeling Environment (**IME**) framework for creating mixed federations of their own special purpose simulations and external simulators. IME allows both that developers of federated simulations do the entire modelling and simulation from scratch and that they use existing simulators for the federation.

11.4.3 More General Frameworks and Middleware for Modelling and Federated Simulation

OpenMI [30, 31], the Open Modelling Interface, is a context based request-reply architecture that defines an interface allowing time-dependent models to exchange data at runtime. A recent application example is described in [53]. Data exchange between models to be linked only takes place if the models are OpenMI-compliant.

The quantities that are to be exchanged must be identified and matched. The models can then be linked at runtime. The very generic character of OpenMI leads to similar restrictions (complexity overhead) as in the case of HLA (cf. below). The development of OpenMI originated from the field of water related research and is promoted by the OpenMI association [31].

IDSim [54], the Interoperable Distributed Simulation framework, is a middleware for federated simulation that has been designed for distributed federated simulation. Following one of its initial design requirements, IDSim uses standard open technologies. IDSim's communication middleware is built on the open standard OGSI, the Open Grid Service Infrastructure, and abstract simulation models are represented in XMSF-based documents (XMSF: Extensible Modelling and Simulation Framework, [55, 56]). IDSim has been employed by the US-American facility NISAC (National Infrastructures Simulation and Analysis Center [57]) for demonstrating that it is feasible to integrate a federation of distributed simulation and a federation of distributed collaboration in the homeland security domain [58]. Within this federation of two federations, IDSim employs the HLA approach to federate the BioDAC simulation environment with the agent-based N-ABLE environment into a single simulation platform [58]. The IDSim software architecture is depicted in Fig. 11.7. In this architecture, IDSim clients enable the federate simulations communicating via OGSI. All communication between federates is routed through a central IDSim server that functions as orchestrator of the federation. The IDSim server holds status information of the federation and provides also the services for distributed simulation [54]. Data that needs to be recorded while the distributed simulation is running is logged by a storage service, while simulation models and configuration parameters are kept in XML repositories using XMSF-based syntax. Once again, this architecture matches the generic federation scheme that we depicted earlier in Fig. 11.1.

The Extensible Modeling and Simulation Framework (**XMSF**) [55, 56] has been developed with the goal of running HLA compliant simulators in a distributed fashion over the Internet and make them interoperable with other components needed for a federation (see Fig. 11.1). For this purpose, XMSF allows adding web services to HLA compliant simulators.

Besides IDSim, the **WSIM** (Web Services Internet Management) architecture [59] is another example that makes use of XMSF. The authors address the need for a sophisticated **interest management** [60], a concept that has been developed for reducing the amount of data that has to be transmitted between federates in order to optimise performance. The basic idea is to transmit only the data needed by a certain federate and only at certain points in time. WSIM extends that concept by using aggregated information and role based access control to clients within the federation. Figure 11.8 shows the scheme of the top-level architecture of a federation using WSIM. The development of XMSF and WSIM has been driven by the military domain, while IDSim has been developed for homeland security.

Other frameworks include **ASimJava** [61, 62], a Java based framework for federated and distributed simulation of large-scale physical systems as well as the aforementioned **IME** of Tolone et al. [15, 51, 52].

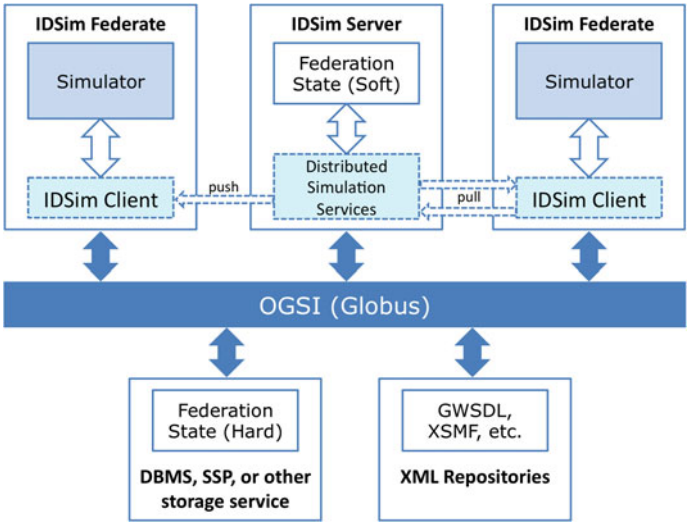

Fig. 11.7 Federation architecture of IDSim, using an IDSim server as orchestrator of the federation (after [54])

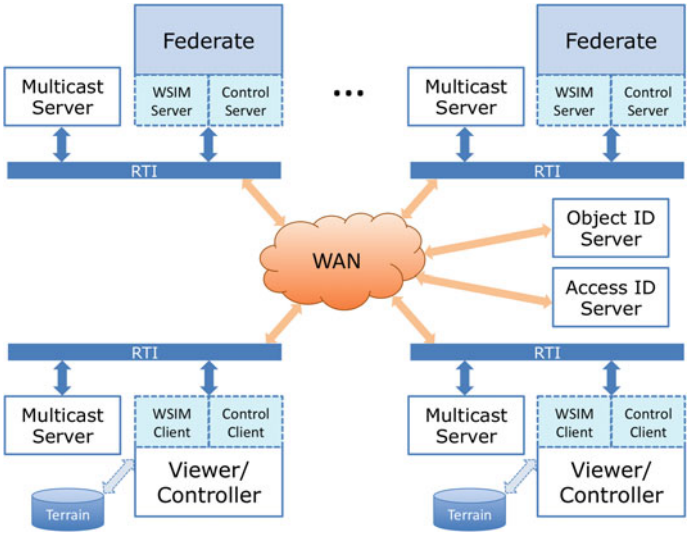

Fig. 11.8 Scheme of the distributed federation architecture WSIM using the XMSF framework (after [59])

11.4.4 The HLA Standard

In the Defence industry, coupling of simulators has been done over a long time and for various reasons. This paragraph describes the evolvement of this need and how it grew into its current standard called HLA [26]. It gives in short the benefits of the HLA approach, how it works and how the Critical Infrastructure Protection (CIP) community can benefit from these lessons.

In the early 1980s, training and education with the use of interactive simulators in virtual environments was very expensive. At the United States Defense Advance Research Projects Agency (DARPA) they realised that there was a need for multi-user simulation for real-time combat training. In order to achieve this in the most cost effective way, they came up with the idea to link single-user simulators in a multi-user environment. To prove this idea they developed the Simulator Network (SIMNET) as a wide area network of vehicle simulators like tanks, airplanes and helicopters together with computer generated forces (CGF). Based on the successes of SIMNET, a standard was derived for linking interactive simulators. This standard, called Distributed Interactive Simulation (DIS) is defined under IEEE standard 1278 in 1993 [41] and has had several revisions since. It is still used and a new version (version 7) is currently (2013) produced.

The DIS protocol is based on the principle that all participating simulators can act as a stand-alone simulator. Therefore all simulators keep all information necessary to create the (static part of the) virtual world. In order for the simulators to be able to interact with each other, every simulator sends the absolute truth about the (externally observable) state of the object it represents to all participants (broadcast). Every receiver has to decide for itself whether it is affected by these transmissions. For example, an airplane broadcasts its location. A radar receives this location and the internal algorithms determine whether this airplane is visible to the radar. The same holds for interactions like fire and detonations. The DIS standard contains information about update rates and dead reckoning algorithms. This allows simulators to join and resign the exercise without interrupting the others, and to lower the bandwidth consumption.

The the big advantage of the DIS standard is that the link is defined in a network protocol. This makes it easy to link simulators that comply to the standard. The downfall is that it is rigid: only the information contained in the standard is exchanged. Also it is limited to real time simulation and its broadcasting technique makes it network intensive. To overcome this, the United States Department of Defence (DoD) started the development of the High Level Architecture (HLA) in the late 1990s. HLA is defined under IEEE Standard 1516 in 2000 [26] and in 2010 revised as HLA evolved. HLA enables computer simulations to interact (that is, to communicate data, and to synchronise actions) with other computer simulations. The interaction between simulations is managed by a Run-Time Infrastructure (RTI) (Fig. 11.9).

Simulations used in HLA can be mathematical, rule-based, etc. and can be with or without human in the loop. If a simulator implementation is HLA-compliant, it is called a federate. HLA simulations, made up of federates, are called federations.

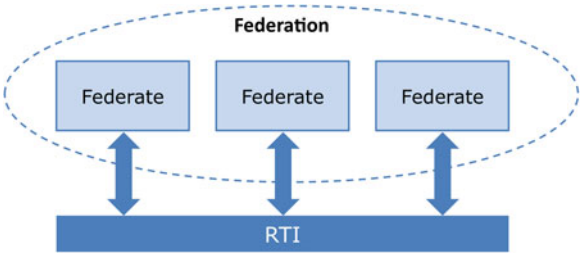

Fig. 11.9 Federation of simulators and Run-Time Infrastructure (RTI) middleware

Objects and interactions that are exchanged between federates in a federation need to be defined in a document. In HLA this is called the Object Model Template (OMT).

In the next paragraphs, we will present the HLA components: *HLA Rules*, *Interface Specification*, and *Object Model Template*.

11.4.4.1 The Rules of HLA

The core of HLA consists of a set of ten HLA rules which a federate or a federation must observe to be HLA-compliant [27]. The HLA rules are divided into two groups, five rules for HLA federations and five rules for HLA federates. The federation rules are aimed to create a federation, they include the following concepts:

- Documentation requirements: federations shall have a Federation Object Model FOM, documented in accordance with the HLA OMT.
- Object representation: all representation of objects in the FOM shall be in the federates, not in the RTI (Run Time Interface).
- Data interchange: during a federation execution, all exchange of FOM data among federates shall occur via the RTI.
- Interfacing requirements: during a federation execution, federates shall interact with the RTI in accordance with the HLA interface specification.
- Attribute ownership: during a federation execution, an instance attribute shall be owned by at most one federate at any given time.

The federate rules deal with the individual federates, they cover:

- Documentation: federates shall have a SOM (Simulation Object Model), documented in accordance with the HLA OMT.
- Control of object attributes: federates shall be able to update and/or reflect any in-stance attributes asw well as to send and/or receive interactions, as specified in their SOMs.
- Owner of object attributes: federates shall be able to transfer and/or accept ownership of attributes dynamically during a federation execution, as specified in their SOMs.

- Transfer of object attributes: federates shall be able to vary the conditions under which they provide updates of instance attributes, as specified in their SOMs.
- Time-management: federates shall be able to manage local time in a way that will allow them to coordinate data exchange with other members of a federation.

11.4.4.2 The Runtime Interface

The functional interfaces between federates and the runtime infrastructure (RTI) are defined by the interface specification. It has been adopted as IEEE standard (P1516.1). The RTI is not a part of the specification, but it is a software that matches the specification. In fact RTI provides the software services necessary for supporting an HLA-compliant simulation. There are different versions of RTI. The interface specification identifies not only the way federates will interoperate with the federation but also one with each other. The Run Time Infrastructure includes:

- Software providing common services to simulation systems
- Implementation of the federate initiated services in accordance with the HLA Interface Specification
- An architectural foundation encouraging portability and interoperability.

11.4.4.3 The Object Model Template

To achieve reusability and interoperability it is required that all objects and interactions, managed by a federate, are specified in detail with a common format. For this reason the Object Model Template (OMT) provides a standard to document the HLA Object Model information. In OMT three Object Models are defined:

- **The Federation Object Model (FOM)**: Every federation has only one FOM that introduces all shared information (e.g., objects, interactions). The FOM contemplates inter-federate issues (e.g., data encoding schemes).
- **The Simulation (or Federate) Object Model (SOM)**: Every federate has one SOM, hence a federation can have several. A SOM describes salient characteristics of a federate and presents objects and interactions that can be used externally. The SOM focuses on the federate's internal operation.
- **The Management Object Model (MOM)**: The MOM identifies objects and interactions used to manage a federation.

Although the HLA standard originally was developed by the defence organisations, it gives a general approach to connecting simulators of different fidelity and function. Therefore it's use can be much broader. The CIP community can have benefit from HLA as it is an open standard which provides all ingredients for linking simulators from different domains. Several commercial and non-commercial RTIs are available.

11.4.5 The DIESIS Approach to Semantic Interoperability

The EU funded project DIESIS (Design of an Interoperable European Federated Simulation network for Critical InfrastructureS) was a design study that assessed the technological, economical and organisational feasibility for European Infrastructures Simulation and Analysis Centre (EISAC) [16]. The distributed facility shall later be used by researchers, national security agencies and CI stakeholders in order to perform modelling, simulation and analysis for investigating a wide range of aspects of national and European CIs [24]. A prerequisite for establishing such a facility is the existence of flexible concepts for coupling heterogeneous simulation systems and their models. The DIESIS approach consists of an ICT architecture, a middleware layer for federating simulators and tools, a communication middleware for connecting distributed simulators and an ontology-based approach for achieving semantic interoperability (for detailed description of particular aspects see [24, 63, 64]).

Unlike HLA, the interoperability approach in DIESIS has been developed to fulfil specific requirements stated by the concept of EISAC: the ability to execute federated simulations of arbitrary (often large and complex) interconnected CIs in order to analyse their interplay, to identify dangerous situations and to assess risks under consideration of cross-domain dependencies. The proposed approach had to be able to get along with different time models and incompatible interface technologies provided by commercial closed-source simulators. Despite of its technological heterogeneity, the approach had to define a common superior modelling perspective that would allow to describe relations beyond the "world" of a single infrastructure. These requirements led to two basic concepts of DIESIS: lateral coupling of federates and separation of technical and semantic interoperability layers.

There are only few off-the-shelf CI simulators that support established interoperability standards like HLA. Experience shows that an attempt to find a common practicable interoperability solution for a set of CI simulators is often doomed to fail for several reasons. Firstly, the development of some specific features may require an enormous effort. Secondly, some particular couplings may be inefficient and significantly slow down the federated simulation. Finally, a desired global solution may not exist at all. Obviously, the increasing complexity of cross-domain dependencies and the growing number of different federates make the existence of a practicable solution even less probable. For this reason, DIESIS proposes a new interoperability approach that abandons the idea of a generic homogeneous architecture that uses a single RTI like HLA (see Sect. 11.4.4).

The proposed concept of **lateral simulator coupling** stands for the development of dedicated coupling links if and only if data exchange between the corresponding federates is required for the current analysis task (see Fig. 11.3). In such federation, pairwise couplings may coexist with clusters of centrally coupled simulators. The systematic development of coupling links implies a *scenario-oriented* federation design. This means that the specification of links is based on the knowledge about the involved domains and their interdependencies, about the runtime behaviour of particular federates as well as about the intended simulation output. In other words,

for realisation of coupling links, the connectivity of the federation has to be described both at technical and at semantic levels (see Sect. 11.3.1) by means of appropriate formalisms.

On the semantic level, the DIESIS approach employs an ontology-based representation to describe infrastructures, general dependencies, infrastructure elements and their relations [63]. DIESIS Knowledge Base System (KBS) uses Ontology Web Language (OWL) and Semantic Web Rule Language (SWRL) to define a *scenario* at three semantic layers [24]:

- **World Ontology** (WONT) is a template that provides basic logical concepts for describing infrastructures as well as their possible behaviours and interdependencies.
- **Infrastructure Ontology** (IONT) is based on the WONT template and describes one particular CI with its domain-specific properties and concrete elements (individuals). The KBS contains one IONT for each infrastructure represented in the federation. A IONT does not necessarily completely duplicate the underlying simulator model with all its facets. It is sufficient to model elements and relations that are involved in cross-domain activities.
- **Federation Ontology** (FONT) is dedicated to the modelling of dependencies among particular CIs. The FONT includes all IONTs and supports dependency modelling at general level (e.g., "a base station receives electric power from a power node") as well as in relation to concrete instances. The *FONT rules* (written in SWRL) express the dependency semantics (for example, "a base station is off if it gets no electric power") for particular relations.

The ontology-based model captures all facets of interplay among the infrastructures independently from the implementation of simulators and coupling links. However, its role is not limited to providing a guideline for link realisation. Data from KBS can be also used by the links at runtime for routing (i.e., sending internal state changes to the right federates according to dependency relations) as well as for automatic data transformation and filtering.

As already mentioned in Sect. 11.3.1, a problem of lateral coupling approach is the potentially large number of links that has to be developed for creating a federation as well as for adding a new member to an existing one. A possible solution is to implement similar links only once and to reuse them if possible. Creating lightweight links for particular tasks instead of complex "all-in-one" couplings significantly increases the probability that a resulting link can be reused for another pair of federates. The DIESIS architectural approach recommends the following four links types:

- **Time links** allow simulators to synchronise their internal clocks and to ensure the correct ordering of processed and sent events. A central synchronisation mechanism is possible but not required. Theoretically, a federation may contain clusters that internally use both conservative and optimistic synchronisation algorithms (see Sect. 11.3.2).
- **Data links** are used by simulators for exchanging their state changes and simulation results. Besides of individual implementations, it is possible to develop

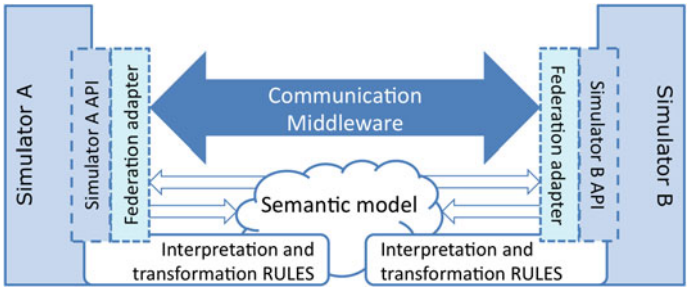

Fig. 11.10 DIESIS approach: simulator interoperability on semantic (*white blocks*) and on technical levels (federation adapters and communication middleware)

a common data routing, transformation and filtering algorithm that uses the dependency information from the KBS. Figure 11.10 shows the structure of such an "intelligent" data link (as it was created for the DIESIS demonstrator) that transforms data acquired from the simulator API according to KBS relations and rules before sending it to another federate.

- **Function links** serve the purpose of mutual invocation of function calls among federates. This may be used by a simulator, for example, to exploit the computational procedures of other simulators.
- **Control links** are employed to manage the runtime behaviour of federates, like starting, stopping or reconfiguring the simulators.

The realisation of links is considered as three-step process and follows the idea of clear separation of technical and semantic interoperability. In the first step, basic logical relationships among CIs have to be defined. In parallel to the formal ontology-based representation, this information can be visualised by means of *service networks*. A service network consists of *agents* that are connected by labelled directed *service links* (for more detailed definition and examples see [65]). In the second step, the required technological extensions (both agents and service links) have to be identified and added to the service network. Possible examples for such extensions are: central simulation control panel with corresponding control links, time management agents with their time links, visualisation and analysis modules, etc. In the final realisation step, the missing service network links and agents have to be implemented and deployed.

The realisation of a sufficiently complex demonstrator (four domains represented by simulators that do not support any common standards) showed the effectiveness and flexibility of the DIESIS approach. It turned out that the usage of a KBS in combination with a lateral coupling approach is a general advantage that significantly reduces the implementation effort.

11.5 Conclusion

Federated modelling, simulation and analysis is an invaluable method with various applications in the defence and civil security domains. It is a premier means of research and development of methods that aid in improving the resilience and protection of Critical Infrastructures, and, at the same time, a research subject of its own. Federated MS&A can be employed as exercise and training environment for crisis and emergency managers, and may serve as part of a decision support system for exploring different courses of action in case of a crisis or emergency. And finally, it can be used for testing and benchmarking new methods for CIP. The case studies and demonstrators reported in the state of the art literature expose impressive new capabilities that have a clear benefit for civil security and thus for society as a whole. The role model is the United States' NISAC that operates under congress mandate for ten years now. In Europe, capabilities like NISAC's are still missing.

A key enabling feature for federated MS&A is the interoperability of federates. They need to exchange data in a syntactically and semantically correct way, and at the correct points in simulation time. The latter requirement cannot be fulfilled for all possible combinations of possible different time models. Our review of the state of the art exposed that independent groups developed similar technical solutions for making federates interoperable, despite the existence of the interoperability standard HLA. As a matter of fact, HLA imposes strong requirements on HLA-compliant federate simulators, and the implementation of the Run-Time Infrastructure, a key interoperability technology, is not part of the standard. We conclude that these obstacles slow down the adoption of HLA in its original domain, defence. OpenMI is a well established M&S standard in domains related to water.

The situation is worse when it comes to interoperability of simulators for CIP. There are numerous simulators for several different infrastructures: railway simulators, electrical power network simulators, telecommunication network simulators and so on. However, almost all of these simulators have been designed for their domains only, not for becoming part of a federation. Some of them even do not have APIs. Although interoperability standards are desirable also for applications of federated MS&A for CIP, it is not likely that the makers of commercial simulators make investments into making their products compliant with some interoperability standard, as long as a convincing business model or a significant market for such an enhanced product is missing. As one solution to this problem, the DIESIS project suggests that the CIP research community joins resources and creates a repository of reusable interoperability solutions for CI simulators. Other groups avoided the problem by creating integrated simulators that cover several infrastructure domains. However, this approach is limited to a certain level of modelling abstraction. Whenever high-fidelity simulations are required, special purpose simulators are superior. Their integration into a federation then requires suitable interoperability middleware.

A second obstacle for a more wide-spread usage of federated MS&A is the fact that setting up federations is a time-consuming task that requires multi-disciplinary expertise. We appreciate that a few researchers have documented their expertise in

setting up federations and have created descriptions of the workflow for this task [49, 66]. However, standardising such workflows and training researchers in modelling and setting up federations adhering to such workflows should be considered as an aid to capacity building in European CIP research. This would be a first step to fill a security gap, as pointed out in [67]: Given the complexity of European CI systems, Europe would urgently need MS&A capabilities comparable to those in the USA.

Acknowledgments We would like to thank all our colleagues and project partners who collaborated with us over the last 7 years in several projects related to CIP. Our special thanks go to: Eric Luiijf, Marieke Klaver, Albert Nieuwenhuis, Patrick Hanckmann, Jeroen Voogd (TNO); Césaire Beyel, Uwe Beyer, Rüdiger Klein (Fraunhofer IAIS); Alberto Tofani, Vittorio Rosato, Paolo Palazzari, Elisa Castorini, Claudio Balducelli (ENEA); Paolo Servillo, Vincenzo Masucci (formerly CRIAI); Gökce Görbil, Erol Gelenbe, Ricardo Lent (Imperial College); and Sandro Bologna (AIIC). We also gratefully acknowledge that some of our own work cited here (IRRIIS, DIESIS) has been co-funded by the EU.

References

1. Luiijf, H., Klaver, M.: International interdependency of C(I)IP in Europe (Internationale Verflechtung von C(I)IP in Europa). In: Proc. CIP Europe 2005—Critical Infrastructure Protection, GI CIS Forum, Bonn, Germany. (2005)
2. EC: Council Directive 2008/114/EC of 8 December 2008 on the identification and designation of European critical infrastructures and the assessment of the need to improve their protection. OJEU, European Commission (2008)
3. Nieuwenhuijs, A., Luiijf, H., Klaver, M.: Modeling critical infrastructure dependencies. In Mauricio, P., Shenoi, S., eds.: Critical Infrastructure Protection II. Volume 290 of IFIP., Boston, MA, USA, Springer (2008) 205–214
4. Setola, R., Bologna, S., Casalicchio, E., Masucci, V.: An integrated approach for simulating interdependencies. In Papa, M., Shenoi, S., eds.: Critical Infrastructure Protection II. Volume 290 of The International Federation for Information Processing. Springer US (2009) 229–239
5. Luiijf, H., Nieuwenhuijs, A.H., Klaver, M.H., Eeten, M.J.V., Cruz, E.: Empirical findings on European critical infrastructure dependencies. Int. J. of System of, Systems Engineering **2**(1), (2010) 3–18
6. Pederson, P., Dudenhoeffer, D., Hartley, S., Permann, M.: Critical infrastructure interdependency modeling: A survey of U.S. and international research. Technical Report INL/EXT-06-11464, Idaho National Laboratory (August 2006)
7. Dudenhoeffer, D., Permann, M., Manic, M.: Cims: A framework for infrastructure interdependency modeling and analysis. In: Simulation Conference, 2006. WSC 06. Proc. Winter. (Dec. 2006) 478–485
8. Min, H., Beyeler, W., Brown, T., Son, Y., Jones, A.: Toward modeling and simulation of critical national infrastructure interdependencies. IIE Transactions **39**(1) (2007) 57–71
9. Laprie, J.C., Kanoun, K., Kaâniche, M.: Modelling interdependencies between the electricity and information infrastructures. In Saglietti, F., Oster, N., eds.: SAFECOMP 2007. Volume 4680 of LNCS. Springer, Berlin Heidelberg (2007) 54–67
10. Casalicchio, E., Galli, E., Tucci, S.: Federated agent-based modeling and simulation approach to study interdependencies in IT Critical Infrastructures. In: Distributed Simulation and Real-Time Applications, 2007. DS-RT 2007. 11th IEEE International, Symposium. (Oct. 2007) 182–189
11. Svendsen, N.K., Wolthusen, S.D.: Connectivity models of interdependency in mixed-type critical infrastructure networks. Inf. Secur. Tech. Rep. **12**(1) (March 2007) 44–55

12. Rosato, V., Issacharoff, L., Tiriticco, F., Meloni, S., Porcellinis, S.D., Setola, R.: Modelling interdependent infrastructures using interacting dynamical models. Int. J. of Critical Infrastructures **4**(1/2) (2008) 63–79
13. Rinaldi, S., Peerenboom, J., Kelly, T.: Identifying, Understanding, and Analyzing Critical Infrastructure Interdependencies. IEEE Control System Magazine **December** (2001) 11–25
14. Rinaldi, S.M.: Modeling and simulating critical infrastructures and their interdependencies. In: Proc. 37th Annual Hawaii Int. Conf. System Sciences (HICSS'04)—Volume 2, Washington, DC, USA, IEEE Computer Society (2004) 20054.1
15. Tolone, W.J., Wilson, D., Raja, A., Xiang, W.N., Hao, H., Phelps, S., Johnson, E.W.: Critical infrastructure integration modeling and simulation. In Chen, H., Moore, R., Zeng, E., Leavitt, J., eds.: Proc. 2nd Symposium on Intelligence and Security Informatics (ISI-2004). Volume 3073 of LNCS., Springer-Verlag (2004) 214–225
16. Rome, E., Bologna, S., Gelenbe, E., Luiijf, E., Masucci, V.: DIESIS—design of an interoperable European federated simulation network for critical infrastructures. In: Proceedings of the 2009 SISO European Simulation Interoperability Workshop (EURO SIW '09), San Diego, CA, USA, Simulation Councils, Inc. (2009) 139–146
17. Bloomfield, R., Chozos, N., Nobles, P.: Infrastructure interdependency analysis: Requirements, capabilities and strategy. Technical Report D/418/12101/3, Adelard LLP, London, UK (2009)
18. Luiijf, H., Klaver, M.: Critical infrastructure awareness required by civil emergency planning. In: Proc. 1st IEEE International Workshop on Critical Infrastructure Protection (IWCIP '05), Washington, DC, USA, IEEE Computer Society (2005) 110–118
19. Martí, J., Ventura, C., Hollman, J., Srivastava, K., Juárez, H.: I2Sim Modelling and Simulation Framework for Scenario Development, Training, and Real-Time Decision Support of Multiple Interdependent Critical Infrastructures during Large Emergencies. In: How is Modelling and Simulation Meeting the Defence Challenges out to 2015? Volume RTO-MP-MSG-060., NATO RTO Modelling and Simulation Group Conf., Vancouver, BC, Canada (October 2008) 16.1–16.14
20. Klein, R., Rome, E., Beyel, C., Linnemann, R., Reinhardt, W., Usov, A.: Information modelling and simulation in large interdependent critical infrastructures. In Setola, R.E.e.a., ed.: Proc. 3rd International Workshop on Critical Information Infrastructures Security (CRITIS '08). Volume 5508 of LNAI., Berlin, Springer-Verlag (2009) 36–47
21. Usov, A., Beyel, C.: Simulating interdependent critical infrastructures with SimCIP. European CIIP Newsletter **4**(3) (November/December 2008) 6–8
22. Balducelli, C., Pietro, A.D., Lavalle, L., Vicoli, G.: A middleware improved technology (MIT) to mitigate interdependencies between critical infrastructures. In de Lemos, R., Giandomenico, F., Gacek, C., Muccini, H., Vieira, M., eds.: Architecting Dependable Systems V. Volume 5135 of LNCS. Springer, Berlin / Heidelberg (2008)
23. Tofani, A., Castorini, E., Palazzari, P., Usov, A., Beyel, C., Rome, E., Servillo, P.: Using ontologies for the federated simulation of critical infrastructures. Procedia Computer Science **1**(1) (2010) 2301–2309
24. Usov, A., Beyel, C., Rome, E., Beyer, U., Castorini, E., Palazzari, P., Tofani, A.: The DIESIS approach to semantically interoperable federated critical infrastructure simulation. In: Advances in System Simulation (SIMUL), 2010 Second International Conference on. (August 2010) 121–128
25. Adinolfi, F., Monica, M.D., Masucci, V., Olivadoti, S., Servillo, P., Spizuocol, C., et al.: DIESIS Deliverable D2.2: Final technology analysis and assessment. Technical report, CRIAI (2009)
26. IEEE: IEEE 1516–2000: High level architecture (2000)
27. IEEE: IEEE standard for modeling and simulation (M&S) high level architecture (HLA)—framework and rules. Technical report, IEEE (2000)
28. IEEE: IEEE 1516–2000: High level architecture—framework and rules (2000)
29. IEEE: IEEE 1516–2000: High level architecture—federate interface specification (2000)
30. Gregersen, J.B., Gijsbers, P.J.A., Westen, S.J.P.: OpenMI: Open modelling interface. J. of Hydroinformatics **9**(3) (2007) 175–191
31. OpenMI Association: http://www.openmi.org last accessed 2013-01-16

32. Fujimoto, R.M.: Parallel simulation: parallel and distributed simulation systems. In: Proceedings of the 33nd conference on Winter simulation. WSC '01, Washington, DC, USA, IEEE Computer Society (2001) 147–157
33. Chandy, K., Misra, J.: Distributed simulation: A case study in design and verification of distributed programs. Software Engineering, IEEE Transactions on **SE-5**(5) (sept. 1979) 440–452
34. Riley, G.F., Ammar, M.H., Fujimoto, R.M., Park, A., Perumalla, K., Xu, D.: A federated approach to distributed network simulation. ACM Trans. Modeling and Computer Simulation **14**(2) (April 2004) 116–148
35. Mattern, F.: Efficient algorithms for distributed snapshots and global virtual time approximation. Journal of Parallel and Distributed Computing **18**(4) (1993)
36. Fujimoto, R., McLean, T., Perumalla, K., Tacic, I.: Design of high performance rti software. In: Distributed Simulation and Real-Time Applications, 2000. (DS-RT 2000). Proceedings. Fourth IEEE International Workshop on, IEEE (2000) 89–96
37. Jefferson, D.R.: Virtual time. ACM Trans. Program. Lang. Syst. **7**(3) (July 1985) 404–425
38. Sokol, L.M., Stucky, B.K.: MTW: Experimental results for a constrained optimistic scheduling paradigm. In Nicol, D., ed.: Distributed Simulation. Volume 22 of Simulation. Society for Computer Simulation (SCS), San Diego, CA (January 1990) 169–173
39. Steinman, J.S.: Breathing time warp. In: Proceedings of the seventh workshop on Parallel and distributed simulation. PADS '93, New York, NY, USA, ACM (1993) 109–118
40. Dickens, P., Reynolds, P.: SRADS with local rollback. Institute for Parallel Computation, School of Engineering and Applied Science, University of Virginia (1990)
41. IEEE: IEEE 1278–1993—standard for distributed interactive simulation (1993)
42. HLA-OMT: High-level architecture object model template specification version 1.3 (5 February 1998)
43. Masucci, V., Adinolfi, F., Servillo, P., Dipoppa, G., Tofani, A.: Ontology-based critical infrastructure modeling and simulation. In Palmer, C., Shenoi, S., eds.: Critical Infrastructure Protection III. Volume 311 of IFIP Advances in Information and Communication Technology. Springer, Berlin Heidelberg (2009) 229–242
44. SISO: SISO generic methodology for verification and validation (GM-VV) to support acceptance of models, simulations and data (2012)
45. Bagheri, E., Ghorbani, A.A.: The state of the art in critical infrastructure protection: a framework for convergence. Int. J. of Critical Infrastructures **4**(3) (2008) 215–244
46. Hopkinson, K., Wang, X., Giovanini, R., Thorp, J., Birman, K., Coury, D.: EPOCHS: A platform for agent-based electric power and communication simulation built from commercial off-the-shelf components. IEEE Trans. on Power Systems **21**(2) (May 2006) 548–559
47. NS2: The Network Simulator: http://www.isi.edu/nsnam/ns last accessed 2013-01-16
48. Bagheri, E., Ghorbani, A.A.: A service oriented approach to critical infrastructure modeling. In: Workshop on Service Oriented Techniques, NRC-Canada (2006)
49. Bagheri, E., Baghi, H., Ghorbani, A.A., Yari, A.: An agent-based service-oriented simulation suite for critical infrastructure behavior analysis. Int. J. of Business Process Integration and Management **2**(4) (2007) 312–326
50. Rahman, H., Armstrong, M., Mao, D., Marti, J.: I2sim: A matrix-partition based framework for critical infrastructure interdependencies simulation. In: Electric Power Conference, 2008. EPEC 2008. IEEE Canada, IEEE (Oct. 2008) 1–8
51. Tolone, W., Lee, S.W., Xiang, W.N., Blackwell, J., Yeager, C., Schumpert, A., Johnson, W.: An integrated methodology for critical infrastructure modeling and simulation. In Papa, M., Shenoi, S., eds.: Critical Infrastructure Protection II. Volume 290 of The International Federation for Information Processing. Springer US (2009) 257–268
52. Tolone, W.J., Johnson, E.W., Lee, S.W., Xiang, W.N., Marsh, L., Yeager, C., Blackwell, J.: Enabling system of systems analysis of critical infrastructure behaviors. In Setola, R., Geretshuber, S., eds.: Critical Information Infrastructure Security. Springer-Verlag, Berlin, Heidelberg (2009) 24–35
53. Betrie, G., van Griensven, A., Mohamed, Y., Popescu, I., Mynett, A., Hummel, S.: Linking SWAT and SOBEK using open modeling interface (OpenMI) for sediment transport simulation in the blue nile river basin. Trans. of the ASABE **54**(5) (2011) 1749–1757

54. Fitzgibbons, J.B., Fujimoto, R.M., Fellig, D., Kleban, S.D., Scholand, A.J.: IDSim: An extensible framework for interoperable distributed simulation. In: IEEE International Conference on Web Services 2004 (ICWS'04), IEEE (2004) 532–539
55. Brutzman, D., Zyda, M., Pullen, J.M., Morse, K.L.: Extensible modeling and simulation framework (XMSF) challenges for web-based modeling & simulation. Findings and recommendations report: Technical challenges workshop, strategic opportunities symposium, MOVES Institute, Monterey, CA, USA (Oct. 22 2002)
56. Pullen, J.M., Brunton, R., Brutzman, D., Drake, D., Hieb, M., Morse, K.L., Tolk, A.: Using web services to integrate heterogeneous simulations in a grid environment. Future Generation Computer Systems **21**(1) (2005) 97–106
57. NISAC, National Infrastructure Simulation and Analysis Center, USA: http://www.sandia.gov/nisac Last accessed 2013-01-16
58. Linebarger, J.M., Fellig, D., Moore, P.D., Goldsby, M., Hawley, M.F., Sa, T.J.: Integrating software architectures for distributed simulations and simulation analysis communities. Technical Report SAND 2005–6642, Sandia National Laboratory (2005)
59. Morse, K., Brunton, R., Pullen, J., McAndrews, P., Tolk, A., Muguira, J.: An architecture for web-services based interest management in real time distributed simulation. In: Distributed Simulation and Real-Time Applications, 2004. DS-RT 2004. Eighth IEEE International Symposium on, IEEE (Oct. 2004) 108–115
60. Morse, K.L., Bic, L., Dillencourt, M.: Interest management in large-scale virtual environments. Presence: Teleoper. Virtual Environ. **9**(1) (February 2000) 52–68
61. Sikora, A., Niewiadomska-Szynkiewicz, E.: A federated approach to parallel and distributed simulation of complex systems. Int. J. Appl. Math. Comput. Sci. **17**(1) (March 2007) 99–106
62. Sikora, A., Niewiadomska-Szynkiewicz, E.: FR/ASimJava: a federated approach to parallel and distributed network simulation in practice. J. Telecommunications & Information Technology **2006**(4) (2006) 53–59
63. Tofani, A., Castorini, E., Palazzari, P., Usov, A., Beyel, C., Rome, E., Servillo, P.: An ontological approach to simulate critical infrastructures. Journal of Computational Science **1**(4) (2010) 221–228 Class-AB
64. Görbil, G., Gelenbe, E.: Design of a mobile agent-based adaptive communication middleware for federations of critical infrastructure simulations. In Rome, E., Bloomfield, R., eds.: Critical Information Infrastructures Security. Volume 6027 of Lecture Notes in Computer Science. Springer, Berlin Heidelberg (2010) 34–49
65. Beyer, U., Usov, A., Rome, E., Beyel, C., et al.: DIESIS Deliverable D4.1b: Final architectural design. Technical report, Fraunhofer IAIS (2009)
66. Tofani, A., Usov, A., Castorini, E., Rome, E., Görbil, G., Palazzari, P., Servillo, P., Hanckmann, P., Beyer, U.: DIESIS Deliverable D4.2a: Proof of concept. Technical report, ENEA (2009)
67. Hämmerli, B., Renda, A.: Protecting Critical Infrastructure in the EU. Technical Report CEPS Task Force Report, Centre for European Policy Studies, Brussels (March 2011)