Representing the CRUTIAL project domain by means of UML diagrams *

Davide Cerotti[1], Daniele Codetta-Raiteri[2], Susanna Donatelli[3],
C. Brasca[4], Giovanna Dondossola[4], Fabrizio Garrone[4]

[1] Dipartimento di Informatica, Università del Piemonte Orientale,
Via Bellini 25/G, 15100 Alessandria, Italy
davide.cerotti@mfn.unipmn.it
[2] Consorzio Nazionale Interuniversitario per le Telecomunicazioni (CNIT), Italy
raiteri@mfn.unipmn.it
[3] Dipartimento di Informatica, Università di Torino,
Corso Svizzera 189, 10149 Torino, Italy
susi@di.unito.it
[4] CESI Ricerca, Via Rubattino 54, 20134 Milano, Italy
{claudio.brasca, giovanna.dondossola, fabrizio.garrone}@cesiricerca.it

Abstract. The paper proposes the representation in form of UML class diagrams of the electric power system (EPS) intended to be composed by two kinds of interdependent infrastructures: the physical infrastructure for the production and the distribution of the electric power, and the ICT infrastructure for the control, the management and monitoring of the physical infrastructure. Such work was developed inside the EU funded project CRUTIAL pursuing the resilience of the EPS. The paper first motivates the use of UML. Then several UML class diagrams representing the EPS organization are presented and described. An example of critical scenario is represented by means of UML diagrams.

Keywords: electric power system, UML, class diagrams, control system scenario, modelling, CRUTIAL

Acronym list:

ATS	Area Telecontrol System
AVR	Area Voltage Regulator
CD	Class Diagram
DSO	Distribution System Operator
EHV	Extra High Voltage
EPS	Electric Power System
HMI	Human Machine Interface
HV	High Voltage
ICT	Information Communication Technology
IED	Intelligent Electronic Device
LAN	Local Area Network

* This work has been supported by the EU under Grant CRUTIAL IST-2004-27513.

LV	Low Voltage
MCDTU	Monitoring Control and Defense Terminal Unit
MV	Medium Voltage
NTS	National Telecontrol System
NVR	National Voltage Regulator
PQR	Reactive (Q) Power Regulator
RTS	Regional Telecontrol System
RVR	Regional Voltage Regulator
SCADA	Supervisory Control and Data Acquisition
TSP	Telecommunication Service Provider
UML	Unified Modelling Language

1 Introduction

In this paper, we resort to the Unified Modelling Language (UML) [1] in order to represent several aspects of the *Electric Power System* (EPS). This work was developed inside the European project named CRUTIAL [2] addressing the EPS intended to be composed by two infrastructures: the physical infrastructure consisting of all the artifacts realizing the electricity transportation from the generation plants to the consumers, and the ICT infrastructure for the management, the control and monitoring of the physical infrastructure.

These two kinds of infrastructure are considered to be interdependent [3, 4] meaning that an accidental failure or a malicious attack affecting an infrastructure may negatively influence the behaviour of the other one. For instance, an attack to a communication network may compromize the information or command exchange among the substations connected by that network; as a consequence, such attack may compromize an automation function of the EPS, such as the Teleoperation or the voltage regulation, causing damages to the physical infrastructure or interrupting the electric power supply.

The general purpose of the CRUTIAL project is investigating the possible ways to realize the resilience of the EPS; this goal is pursued by carrying out several activities, such as the investigation of architectures preventing faults and attacks, together with the identification, the modelling and the quantitative analysis of critical control system scenarios. Such a scenario consists of a particular event sequence occurring in a certain portion of the EPS as a consequence of a failure or an attack.

In the CRUTIAL project, we are particularly interested in the critical scenarios where both infrastructures are exploited in order to perform a certain automation function, and where interdependencies between infrastructures arise. In order to identify such scenarios, we must first define the relevant characteristics of the project domain; this means the identification of: 1) the general organization of the EPS infrastructures; 2) the elements of each infrastructure; 3) the automation functions realized inside the EPS; 4) the elements exploited to per-

form a certain function; 5) the relations and the interactions between elements belonging to different infrastructures.

In this way, the possible interdependencies between the EPS infrastructures can be put in evidence and consequently the scenarios of interest can be identified and investigated. Such a definition of the project domain requires a standard language suitable to represent the EPS characteristics that we are interested to: UML has been chosen to this aim.

2 The role of UML in the domain definition

UML is adopted to represent the project domain; this choice is motivated by the fact that UML is becoming widely accepted within some industrial communities as a standard design language [5]; for instance, UML diagrams support the documentation of several standards for power control systems (IEC 61970, IEC 61850 [6], IEEE C37.115, etc.). Moreover, UML was adopted in the past in other projects concerning the electric field such as the DEPAUDE project [7]. In [8], the way to use UML to represent both the system structure and its dependability requirements, is investigated.

The reports documenting the EPS domain contain lots of concepts, definitions and descriptions. Their UML representation allows to summarize and connect in a graphical way the information spread across the documentation, for instance by means of *Class Diagrams* (CD). In a sense, our UML representation acts as a base of knowledge where each element of the diagram incorporates a specific aspect of the EPS domain (component, function, etc.), and where the correlations between aspects are put in evidence.

Several partners are involved in the CRUTIAL project and they come from different disciplines, mainly from Electric Engineering and Computer Science. The UML representation of the EPS infrastructure allows to express the nature of the EPS in a common standard language: the CRUTIAL partners coming from fields different from the electric one, can design and manipulate the UML diagrams in order to express their vision of the EPS organization; at the same time, people expert of the electric domain can verify the semantic of the UML diagrams and in this way, they can eventually correct the erroneous interpretations of the EPS concepts by the other partners.

Actually our UML diagrams do not take into account every possible aspect of the EPS: the UML representation of the EPS allowed to determine which aspects are effectively relevant to the project purposes, together with the necessary level of detail, and the possible points of view of each aspect. In a sense, the UML representation of the CRUTIAL domain acts as a glossary shared by the project partners and collecting the concepts from the electric and the ICT field that are relevant to the project.

Besides the representation of the EPS domain, we exploit UML to deal with control system scenarios: by means of UML diagrams, we can identify the scope of the scenario; this means putting in evidence all the key aspects of a scenario such as the components, the functions, etc., involved in the event sequence. The

graphical nature of UML allows to represent the scenario in a intuitive, clear and evident way, such that people coming from different fields can all interpret the semantic of the UML diagram and consequently the scope of the scenario. In Sec. 4, we provide an example of scenario representation by means of a UML object diagram and some state chart diagrams; other forms of UML diagram are exploited in [9] to this aim.

In this way, while examining the scenario, we can concentrate only on the aspects that are relevant to the scenario, ignoring the other elements of the domain. This can be useful when we face the analysis or the simulation of the control system scenarios by means of quantitative models such as Stochastic Petri Nets [10], with the purpose of performance or dependability evaluation of the scenario. These forms of evaluation may be expensive from the point of view of the model construction and in particular from the point of view of the computational cost of the model analysis or simulation. Therefore the necessity to limit our attention only to the elements effectively relevant to the scenario, arises in this phase: the quantitative model may be derived from the UML representation of the scenario in such a way to fit its scope. In [9], about twenty scenarios have been proposed in order to be evaluated.

However in this paper, we deal only with the UML representation of the EPS domain and scenarios; quantitative models will be the object of future work.

3 UML class diagrams of the EPS domain

In this section, we report some of the CDs representing the EPS domain; other CDs can be found in [9]. Actually we could represent the EPS domain by means of a unique CD, but we decided to split the representation into different CDs, each dedicated to a certain aspect of the EPS. Some classes are shared among the CDs in order to be the point of connection between the CDs. The semantic of each CD is described and in this way we provide the description of the domain of the CRUTIAL project.

3.1 The general architecture

Fig. 1.a shows the UML CD of the EPS general architecture. The main class is *ElectricPowerSystem* representing the whole EPS; this class is the aggregation of the following classes representing the main subsystems of the electrical system:

- *PowerGeneration* represents the generation of electric power;
- *PowerGrid* represents the infrastructure used to transport the electric power from the power plants to the consumers; this class is the aggregation of these classes:
  - *TransmissionGrid* represents the grids transferring the electric power from the power plants to the distribution grids;
  - *DistributionGrid* represents the grids transferring the electric power to the consumers.

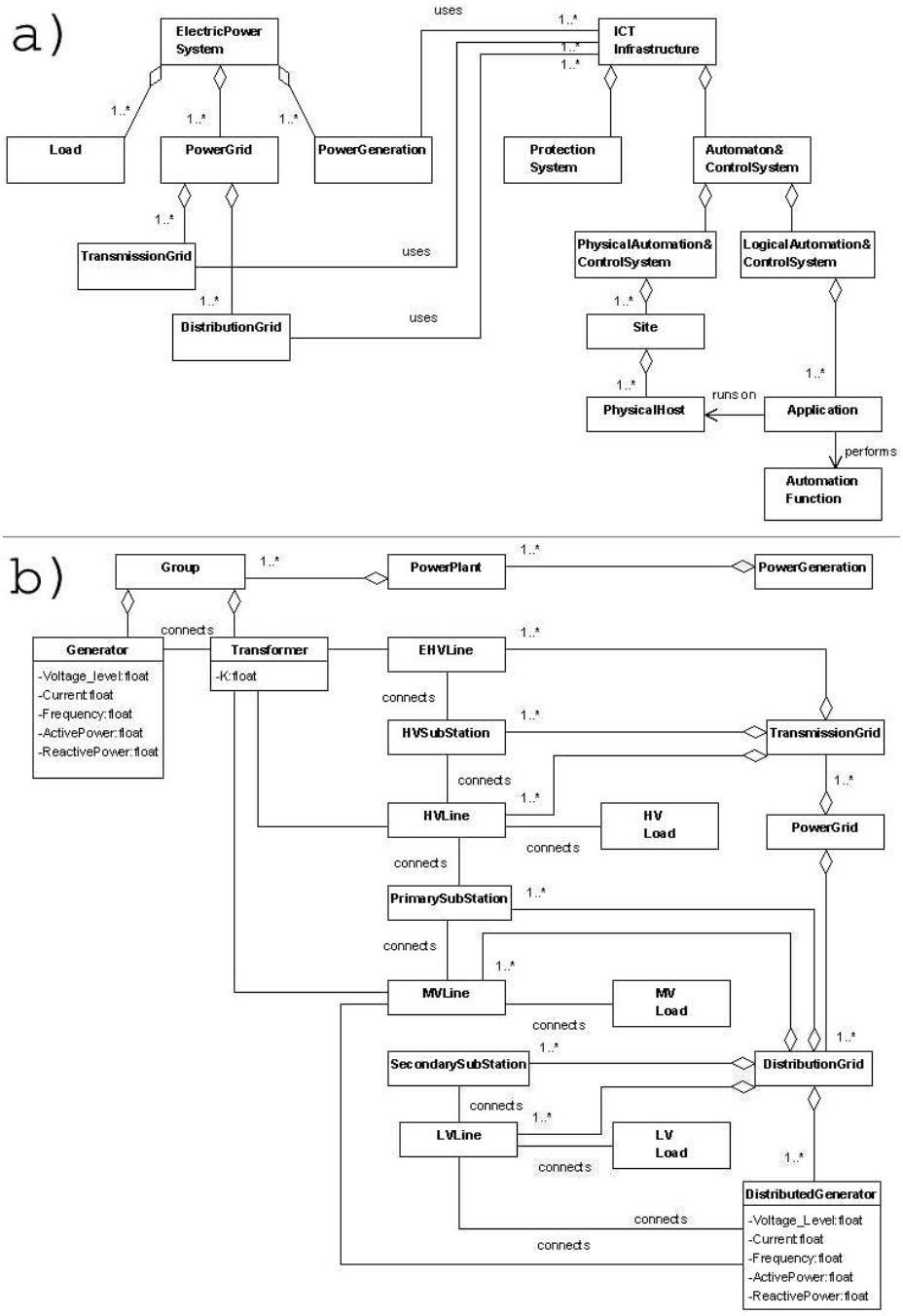

Fig. 1. a) Class diagram of the general EPS architecture **b)** Class diagram of the power generation and of the power grid

– *Load* is the class representing the loads.

The class *ICTInfrastructure* is associated with the classes *PowerGeneration*, *TransmissionGrid* and *DistributionGrid* due to the fact that the control, the management and monitoring of the physical infrastructure is performed by means of the ICT infrastructure. The class *ICTInfrastructure* is the aggregation of the following classes:

– the class *ProtectionSystem* representing the system preserving the safety of the power grid;
– the class *Automation&ControlSystem* representing the system dedicated to the automation and the control of the power grid; such class is the aggregation of these classes:
  • *PhysicalAutomation&ControlSystem* represents the set of the sites realizing the automation and control system; therefore, the class *PhysicalAutomation&ControlSystem* is the aggregation of instances of the class *Site* which is in turn the aggregation of instances of the class *PhysicalHost* representing a generic device connected to the communication network;
  • *LogicalAutomation&ControlSystem* represents the set of software applications performing the automation and control functions; therefore the class *LogicalAutomation&ControlSystem* is the aggregation of instances of the class *Application* representing software applications.

The classes *PhysicalHost* and *Application* are associated because an application runs on a certain physical host. Moreover, the class *Application* is associated with the class *AutomationFunction* because an application performs an automation and control function.

3.2 Power Generation and Power Grid

Fig. 1.b shows the CD representing the power generation together with the power grid. The power generation is represented by the class *PowerGeneration* which is the aggregation of the class *PowerPlant* which represents the power plants for the production of electric power. *PowerGeneration* is already present in the CD in Fig. 1.a.

The class *PowerPlant* is the aggregation of instances of the class *Group* which is in turn the aggregation of the classes *Generator* and *Transformer* representing power generators and transformers respectively. The classes *Generator* and *Transformer* are associated because a generator is connected to a transformer in order to raise the voltage of the produced electric power, to the level used on EHV lines.

The class *PowerGrid* is the aggregation of the class *TransmissionGrid* and of the class *DistributionGrid*. The class *TransmissionGrid* is the aggregation of the following classes:

– the class *EHVLine* is associated with *Transformer* due to the fact that a transformer conveys the electric power from the generator to the to an EHV electric line;

- the class *HVSubStation* is associated with both the class *EHVLine* and with the class *HVLine* because HV substations transform the EHV electric power coming from a EHV electric line, into HV electric power transferred along a HV electric line;
- the class *HVLine* is associated with *HVLoad* in order to represent that a HV electric line is used to connect a HV load to the transmission grid.

The class *DistributionGrid* is the aggregation of the following classes:

- the class *PrimarySubStation* is associated with the class *HVLine* (composing the class *TransmissionGrid*) because a primary substation is connected to the transmission grid by means of a HV electric line. A primary substation is instead connected to the distribution grid by means of a MV line (a primary substation transforms HV electric power into MV electric power); so, the class *PrimarySubStation* is associated also with the class *MVLine*.
- *MVLine* is associated also with *MVLoad* and *DistributedGenerator* because a MV load or a distributed generator is connected to the distribution grid by means of a MV electric line;
- the class *SecondarySubStation* is associated with the classes *MVLine* and *LVLine* because a secondary substation transforms the MV electric power into LV electric power;
- *LVLine* is associated also with the class *LVLoad* and *DistributedGenerator* because a LV load or a distributed generator is connected to the distribution grid by means of a LV electric line.

In the CRUTIAL project, we distinguish between the power generation and the distributed generation [11]: power generation means the production of electric power by means of traditional power plants, while the distributed generation is performed by generators connected to the distribution grid and localized in a distribution area (class *DistributedGenerator* in Fig. 1.b).

3.3 Sites

A site hosts the ICT infrastructures dedicated to the automation, the control and the management of a portion of the power grid. The CD in Fig. 2 represents a site in terms of classes. The main class is *Site* consisting of an aggregation of the class *PhysicalHost*; moreover the class *Site* is associated with itself to represent the fact that automation sites can communicate exchanging information and orders. The class *Site* is already present in the CD in Fig. 1.a.

A site can directly control a substation or a power plant; otherwise a site can control other sites. In order to represent this fact, the class *Site* has been specialized in these classes:

- *SubStationAutomationSite* represents the automation sites controlling a substation in direct way; for this reason, this class is associated with the class *SubStation*.

– *ControlCentreSite* represents the sites controlling other sites on the power grid; the sites of this kind are organized in geographical way, so the class *ControlCentreSite* is specialized in the following classes:
  - *NationalControlCentre* represents the automation sites monitoring the national grid and controlling the regional sites;
  - *RegionalControlCentre* represents the sites monitoring a regional grid and controlling the area (local) sites;
  - *AreaControlCentre* represents the automation sites monitoring and controlling a local area of the power grid; this class is associated with itself to model the possibility for an area control centre to be replaced by another one in case of malfunctioning, or the possibility to realize the redundancy of the area control in normal functioning.
– *PowerPlantAutomationSite* represents the automation sites controlling a power plant in a direct way; for this reason, this class is associated with the class *PowerPlant*.

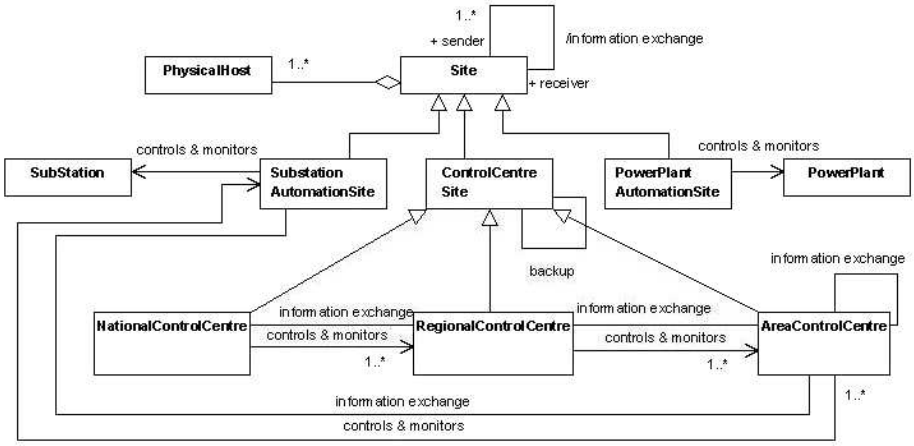

Fig. 2. Class diagram of the site classification

The associations between the class *NationalControlCentre* and the class *RegionalControlCentre* indicate that there is an information exchange between a national control centre and a regional one. The same relations hold between the class *RegionalControlCentre* and the class *AreaControlCentre*. The class *ControlCenterSite* is associated with itself to indicate that a control centre site may be replaced by another one in case of malfunctioning.

3.4 Functions

Activities such as management, monitoring, maintenance and control can be classified as functions. In the CD in Fig. 3, the class *Function* represents the

functions set. A function can be realized in automatic way; therefore the class *Function* is specialized in *AutomationFunction* which is in turn specialized in several classes representing the main automation functionalities: *Protection*, *Management*, *Monitoring*, *Maintenance*, *Regulation*, *Teleoperation* and *Supervision*.

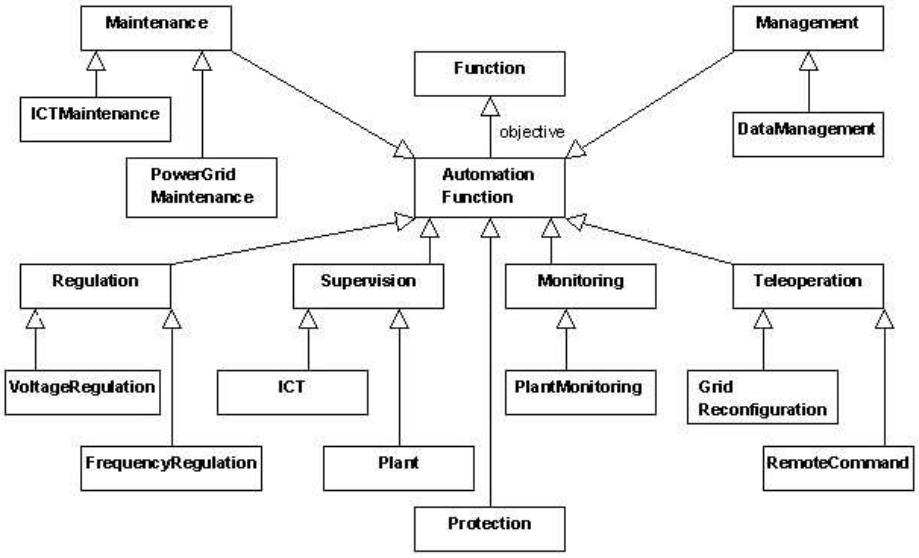

Fig. 3. Class diagram of the functions performed by ICT elements

The class *Regulation* concerning the generic activity of regulation, is specialized in *VoltageRegulation* and *FrequencyRegulation*. The Teleoperation functions are represented by the class *Teleoperation* and they are considered in several critical scenarios [9]. Therefore in section 3.6 we provide a more detailed representation of the Teleoperation.

3.5 ICT elements & Industrial application components

The class *PhysicalHost* represents the generic device connected to the communication network and is already present in the CD in Fig. 1.a. Such class can be specialized in two classes: *WorkStation* (Fig. 4.a) and *Regulation&ControlComponent* (Fig. 4.b).

The ICT elements present in the EPS are represented in the CD in Fig. 4.a where the class *WorkStation* represents generic computers and has several specializations, one for each role of a workstation.

The CD in Fig. 4.b concerns the industrial application components, e.g.: the components dedicated to the regulation and the control of a node of the

power grid. Such components are represented in Fig. 4.b by the class *Regulation&ControlComponent* specialized in *IED*, *MCDTU*, *PQR*, *AVR*, etc.

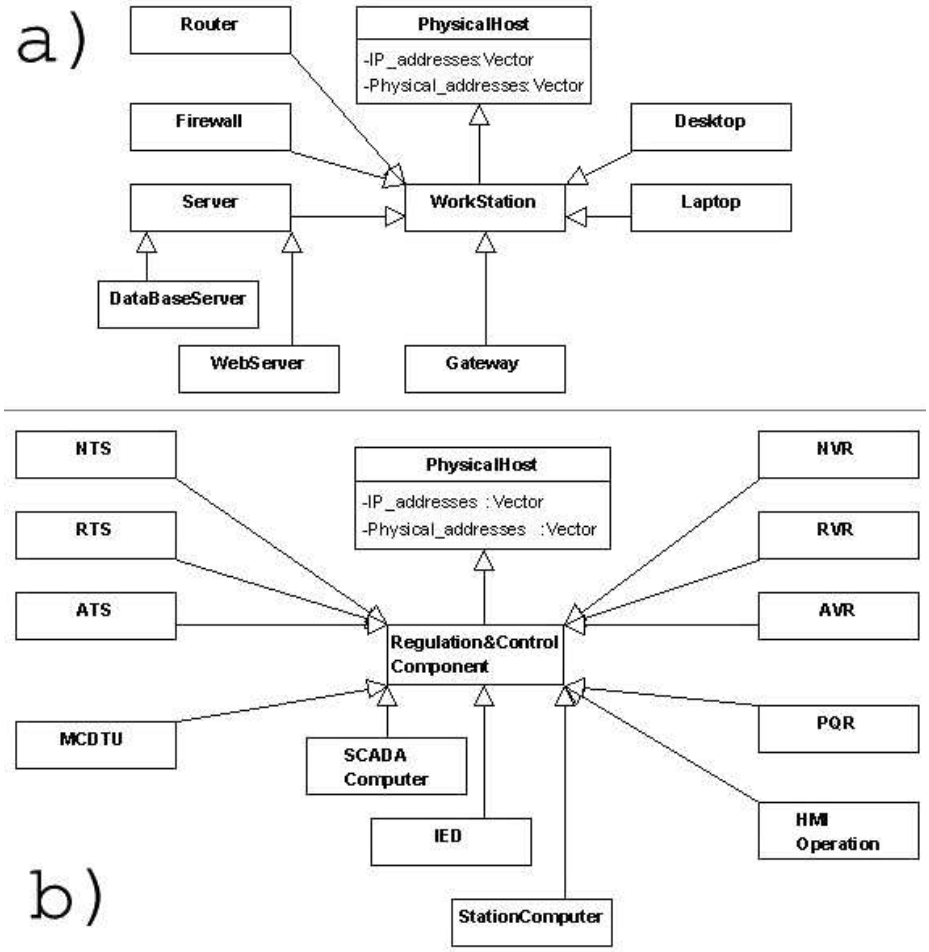

Fig. 4. a) Class diagram of the ICT elements **b)** Class diagram of the industrial application components

3.6 Teleoperation

The structure of the Teleoperation system of the EPS is represented by the CD in Fig. 5 where the main class is *Teleoperation* specialized in *RemoteCommand*. *RemoteCommand* is the aggregation of three classes extending the class *Function*

identifying the functions (*Function* is already present in the CD in Fig. 3); these classes reflect the geographical organization of the Teleoperation; they are: *NationalTeleoperation*, *RegionalTeleoperation* and *AreaTeleoperation*.

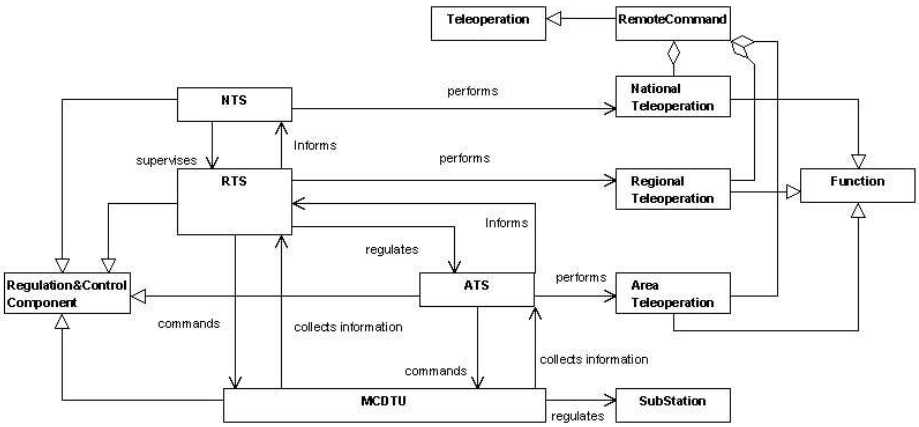

Fig. 5. Class Diagram of the Teleoperation

Besides the description of the Teleoperation in terms of functions, the CD in Fig. 5 indicates also the automation components performing each function. The classes *NTS*, *RTS*, *ATS* and *MCDTU* extend *Regulation&ControlComponent*, already present in the CD in Fig. 4.b.

The class *NTS* is associated with the class *NationalTeleoperation* because *NTS* represents the automation component performing the national Teleoperation; analogously the class *RTS* is associated with the class *RegionalTeleoperation*, the class *ATS* is associated with the class *AreaTeleoperation*. *NTS* and *RTS* are associated since the national Teleoperation can send commands to the regional Teleoperation, while a regional Teleoperation can send information about the state of the corresponding portion of power grid, to the national Teleoperation. Similarly, *RTS* and *ATS* are associated since the regional Teleoperation commands the area Teleoperation, while the regional Teleoperation is informed by the area Teleoperation.

A MCDTU exchanges commands and information also with the regional Teleoperation system, so the class *MCDTU* is associated with the class *RTS*. The class *MCDTU* is associated with the class *ATS* because the area Teleoperation system collects information from the MCDTU automation components; the class *MCDTU* is associated also with the class *SubStation* because the automation components represented by the class *MCDTU* influence the state of the substations.

4 Representing a scenario

Once the project domain has been defined, critical scenarios can be considered. In this section, we provide the partial UML representation of a critical scenario [9] dealing with a case of Teleoperation between an area control centre and a couple of substation automation sites. The Teleoperation activity is performed through the exchange of commands between the area control centre and the substation automation sites. The communication is realized by means of a redundant shared network.

In this scenario, a denial of service attack is performed on the communication channel attempting to reduce the communication bandwidth with the aim of obtaining the delayed or failed delivery of the packets, with consequent partial or complete loss of commands respectively. Some countermeasures, such as firewalling or network traffic monitoring, may detect the attack and recovery from it. More details about this scenario can be found in [9].

In order to represent such scenario, we provide several UML diagrams. In this section, we briefly describe them. First, the object diagram in Fig. 6 indicates all the EPS elements involved in the scenario; they are represented in form of objects, i.e. instances of the classes present in the CDs defining the domain (see Sec. 3 and [9]). This diagram shows also the relations among the objects. In general, the object diagram represents the scope of the scenario and explicits the chosen level of abstraction.

In the scenario under exam, we are interested in the Teleoperation function implemented in a local area, hence in the object diagram in Fig. 6 we include instances of all the classes involved in an area Teleoperation function. We can determine these classes by inspection of the CDs concerning the Teleoperation, starting from the CD in Fig. 5 where we can see that *AreaTeleoperation* is performed by *ATS* sending commands to, and collecting information from *MCDTU*. We can identify the other classes involved in the scenario by inspecting the other CDs where *ATS* and *MCDTU* are present. This procedure is iterated until all the classes relevant to the Teleoperation are identified.

Once we have identified the classes, we have to determine the number of instances of each class involved in the scenario; in our example, we include two instances of *SubStationAutomationSite* and one instance of *AreaControlCentre* connected by two instances of *SharedNetwork* (Fig. 6).

Besides determining the classes involved in the scenario (the scope), we can decide in the object diagram the level of abstraction chosen to represent the scenario. In the diagram in Fig. 6 we show the internal elements of the instances of *AreaControlCenter* and *SubStationAutomationSite*. We could provide the representation of the same scenario with an higher level of abstraction by omitting the internal elements of such instances.

Then, in order to represent the behaviour of a particular element in the scenario, we resort to UML state chart diagrams. This kind of diagram shows the possible states of an element, together with the state transitions. For instance, the states of the *SharedNetwork* in the current scenario, are represented by the diagram in Fig. 7.a, where the possible states are *Idle*, *Normal*, *Delayed* and

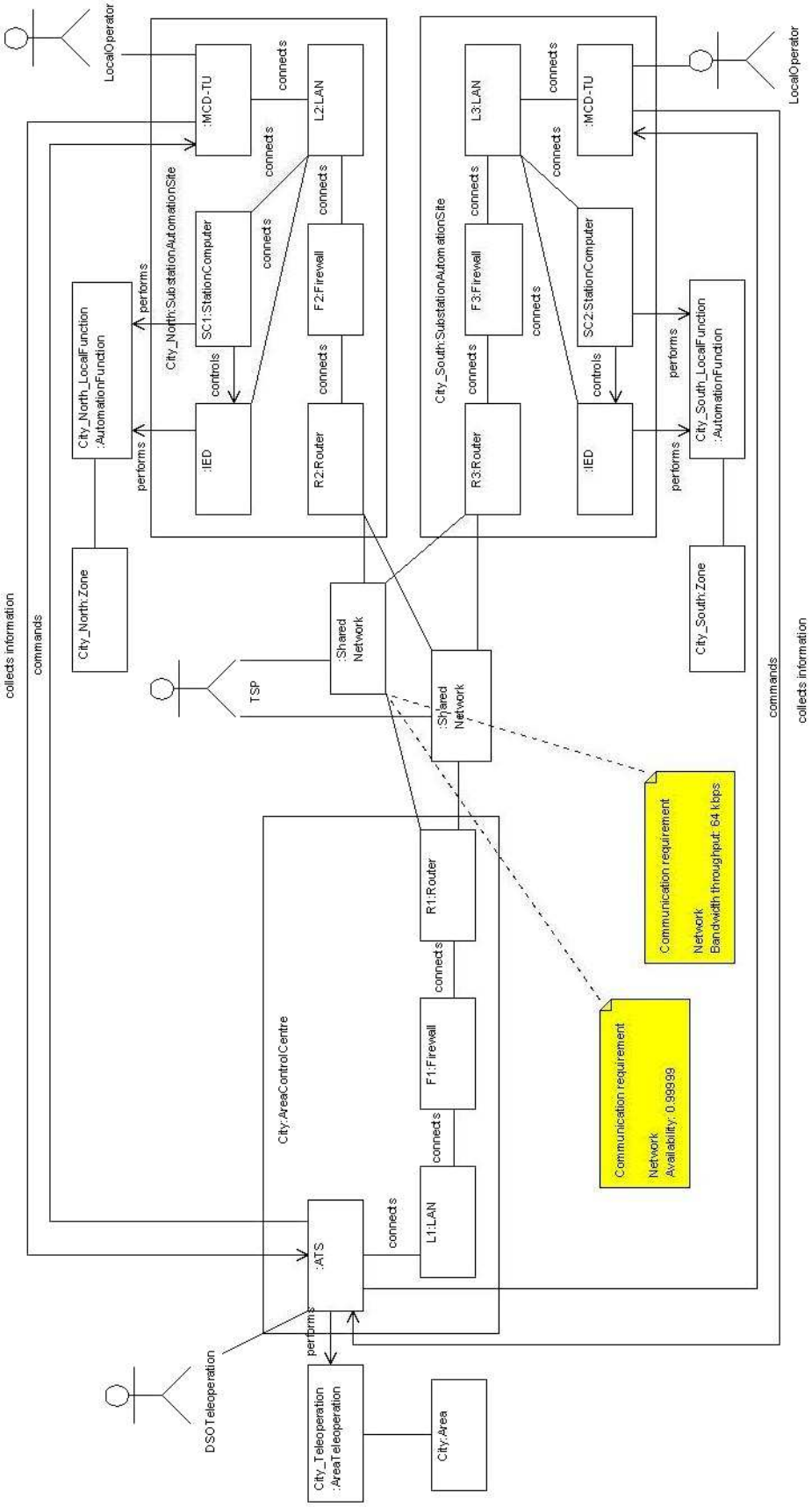

Fig. 6. Object diagram of the scenario

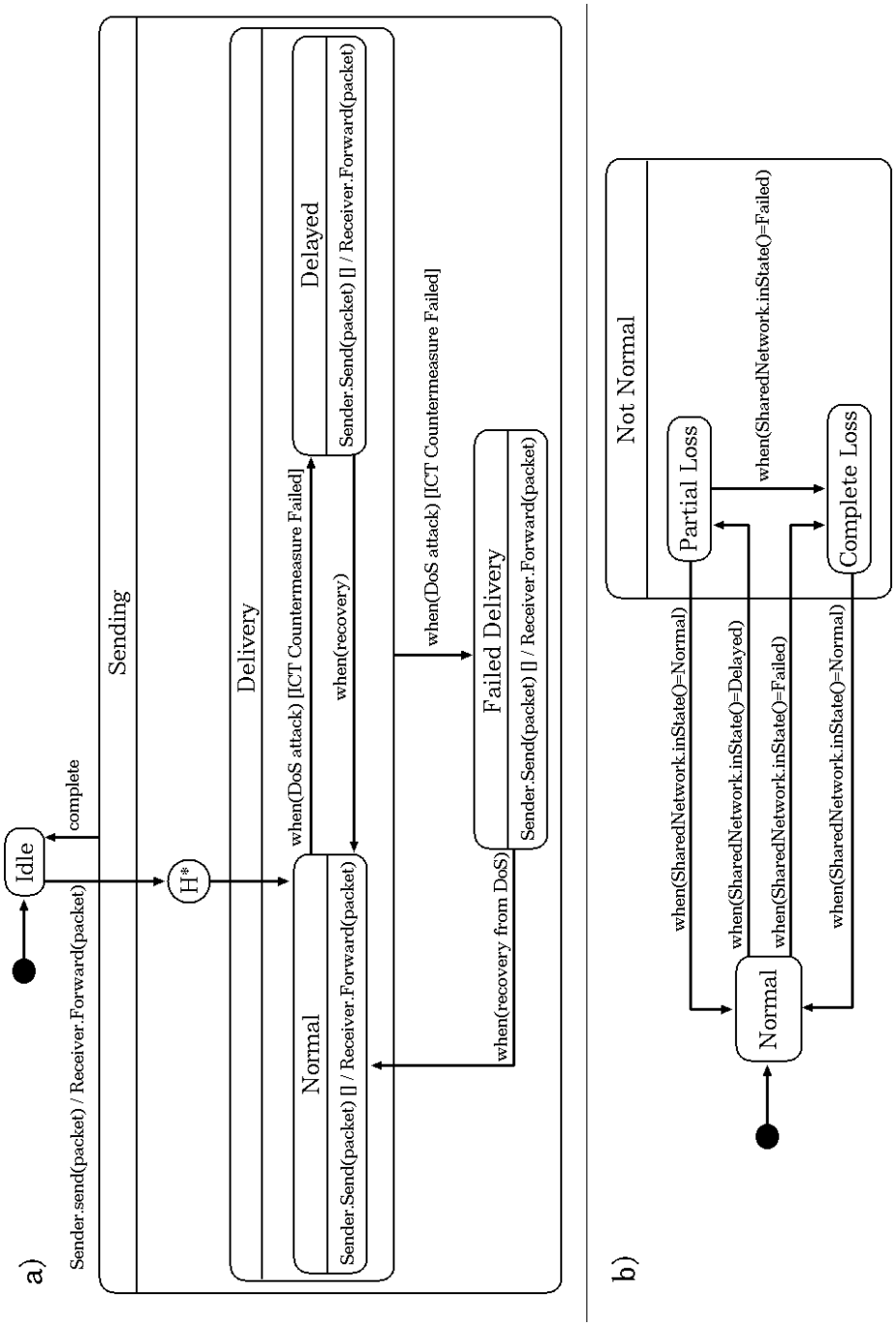

Fig. 7. a) State chart diagram of the *SharedNetwork* in case of denial of service attack **b)** State chart diagram of the Teleoperation function according to the state of the *SharedNetwork*

FailedDelivery. In such diagram, the state transitions are due to the success of the attack or to the success of the countermeasures. The states of the area Teleoperation function are instead represented in the state chart diagram in Fig. 7.b, where the possible states are *Normal*, *PartialLoss* and *CompleteLoss*.

The state transitions in Fig. 7.b are due to the current state of the *Shared-Network* according to the state chart diagram shown in Fig. 7.a. In this way, we represent the dependency between the area Teleoperation function and the network; this is a case of dependency between a function and one of the components participating to its implementation.

5 Future work

In our CDs of the EPS domain, only in a few cases, attributes are specified, while we never resort to methods; therefore we plan to update our CDs in order to increase their detail by specifying more attributes and methods in the classes. Moreover in our CDs, the interdependencies need to be more evident; in this sense, our CDs still need some extensions.

In this paper we showed a possible way to represent a scenario. In [9] we complete the scenario representation by means of UML sequence diagrams. From the scenario representation we should be able to retrieve the information to derive quantitative models, such as Stochastic Petri Nets [10], for the numerical evaluation of the system performance or dependability (as mentioned in Sec. 2). Actually, we remarked that the diagrams exploited for the scenario representation are not enough expressive in order to derive quantitative models in a direct way. Therefore we plan to improve the representation of the scenarios by following two ways: 1) by resorting to the UML profile for *Schedulability, Performance and Time* (SPT) [12]: such profile allows to enrich UML diagrams with temporal and stochastic notations; in this way, an UML diagram can provide additional information useful to the generation of a quantitative model. This approach is already documented in the literature [14]. 2) by means of SysML [13], a rather recent UML extension able to express complex characteristics such as the definition and the organization of the system requirements, the nature of the exchanged data and commands, the definition of test cases, etc.

So far, our attention has been limited to the CRUTIAL project domain. Given such experience, we plan to provide the UML representation of critical infrastructures in general (including the EPS) according to the infrastructure hierarchy and the infrastructure interdependencies defined in [15].

References

1. Pender, T.: UML Bible. Wiley Publishing Inc. (2003).
2. CRUTIAL project web site. http://crutial.cesiricerca.it
3. DeMarco, C. L., Braden, Y.: Threats to Electric Power Grid Security through Hacking of Networked Generation Control. Third International Conference on Critical Infrastructures (CRIS), Alexandria, VA-USA (2006).

4. Dondossola, G., Szanto, J., Masera, M., Nai Fovino, I.: Evaluation of the effects of intentional threats to power substation control systems. International Workshop on Complex Network and Infrastructure Protection (CNIP), Rome, Italy (2006).
5. Object Management Group: UML 2.0 Infrastructure Specification. http://www.uml.org (2002).
6. International Standard IEC 61850-5. Communication network and systems in substations Part 5: Communication requirements for functions and device models, First edition (2003).
7. DEPAUDE project web site, http://www.esat.kuleuven.be/electa/depaude
8. Bernardi, S., Donatelli, D., Dondossola, G.: A class diagram framework for collecting dependability requirements in automation systems. First International Symposium on Leveraging Applications of Formal Methods (ISOLA), Pathos, Cyprus (2004).
9. CRUTIAL deliverable D2. http://crutial.cesiricerca.it (2007).
10. Ajmone-Marsan, M., Balbo, G., Conte, G., Donatelli, S., Franceschinis, G.: Modelling with Generalized Stochastic Petri Nets. J. Wiley Publisher (1995).
11. Ackermann, T., Andersson, G., Sader, L.: Distributed Generation: a definition. Electric Power Systems Research, vol. **57** (2001) 195–204.
12. Object Management Group: UML Profile for Schedulability, Performance, and Time. In OMG document n. ptc/02-03-02 (2002).
13. SysML web site, http://www.sysml.org
14. Bernardi, S., Donatelli, S., Merseguer, J.: From UML Sequence Diagrams and StateCharts to analysable Petri Net models. International Workshop on Software and Performance (WOSP), Rome, Italy, ACM Press (2002), 35–45.
15. Rinaldi, S. M., Peerenboom, J. P., Kelley, T. K.: Identifying, Understanding, and Analyzing Critical Infrastructure Interdependencies. IEEE Control Systems Magazine, vol. **21(6)**, (2001), 11–25.