Real-Time Physical Cybersecurity Massive Open Online Course MOOC – Teacher's Handbook

Authors:
Bahaa Eltahawy, Mike Mekkanen, Tero Vartiainen,
Maria Valliou, Panos Kotsampopoulos, Alexandros Chronis,
Jānis Pekša, Jana Bikovska, Andrejs Romānovs, Rūta Pirta-Dreimane,
Jirapa Kamsamrong, Bjoern Siemers

Cybersecurity Curricula Recommendations for Smart Grids (CC-RSG)
Erasmus+ Strategic Partnership Project
Intellectual Output 3
May 2023
Reviewers: Foivos Palaiogiannis and Irfan Al-Anbagi

Coordinator: University of Vaasa
Partners: OFFIS, Riga Technical University, National Technical University of Athens

This project has received funding from the European Union's Erasmus+ Programme under Grant Agreement № 2020-1-FI01-KA203-066624.

The European Commission's support for the production of this publication does not constitute an endorsement of the contents which reflects the views only of the authors, and the Commission cannot be held responsible for any use which may be made of the information contained therein

Table of Contents

List of Acronyms

ACSI	Abstract Communications Service Interface
AMI	Advanced Metering Infrastructure
APCI	Application Protocol Control Information
ASDU	Application Service Data Unit
BoK	Body of Knowledge
CC-RSG	Cybersecurity Curricula Recommendations in Smart Grids
CEN	the European Committee for Standardization
CI	Critical Infrastructure
CIA	Confidentiality, Integrity and Availability Triad
CPS	Cyber-physical Systems
CPU	Central Processing Unit
CR	Cognitive Radio
CT	Current Transformer
DAM	Database Activity Monitoring
DER	Distributed Energy Resources
DEVS	Discrete Event System Specification
DG	Distributed Generation
DLP	Data Loss Prevention
DMS	Distribution Management System
DOE	Department of Energy
DoS	Denial of Service
DR	Demand Response
DSM	Demand Side Management
DSML	Domain-Specific Modelling Language
DSP	Digital Signal Processor
E2E	End-to-End
EMS	Energy Management System
EPRI	Electric Power Research Institute (USA)
EPS	Electric Power System
ETSI	the European Telecommunications Standards Institute
EU	European Union
EV	Electric Vehicle
FPGA	Field Programmable Gate Array
GHG	Greenhouse Gas
GOOSE	Generic Object-Oriented System Event
GPS	Global Positioning System
GUI	Graphical User Interface
GWAC	Grid-Wise Architecture Council
HIDS	Host Intrusion Detection System
HIL	Hardware in the Loop
HIPS	Host Intrusion Prevention System

HMI	Human Machine Interface
HSRT	High-Speed Real-Time Simulation
HW/SW	Hardware/Software
I/O	Input/Output
ICS	Industrial Control System
ICT	Information and Communication Technology
IDS	Intrusion Detection System
IEC	International Electrotechnical Commission
IED	Intelligent Electronic Device
IEEE	Institute of Electrical and Electronics Engineers
IOT	Internet of Things
IP	Internet Protocol
IPS	Intrusion Prevention System
ISA	International Society of Automation
ISO	International Organization for Standardization
ISP	Internet Service Provider
ISS	Instruction Set Simulator
IT	Information Technology
LET	Logical Execution Time
LN	Logical Node
M&S	Modeling and Simulation
MBD	Model-based Design
MDMS	Meter Data Management Systems
MITM	Man-in-the-Middle
MMS	Manufacturing Message Specification
MOOC	Massive Open Online Course
NBAD	Network Behavior and Anomaly Detection
NIST	National Institute of Standards and Technology
NISTIR	National Institute of Standards and Technology Interagency or Internal Report
OMS	Outage Management System
OT	Operational Technology
P2P	Peer to Peer
PICS	Protocol Implementation Conformance Statement
PMU	Phasor Measurement Unit
PSBD	Progressive Simulation-based Design
PU	Primary Unit
PV	Photovoltaics
QoS	Quality of Service
R&D	Research and Development
RCP	Rapid Control Prototyping
RTHS	Real-time Hybrid Simulation
RTOS	Real-Time Operating Systems
RTU	Remote Terminal Unit

RTW-EC	Real-Time Workshop Embedded Coder
SBT	Simulation-based Training
SCADA	Supervisory Control and Data Acquisition
SCL	Substation Configuration Language
SDR	Software Defined Radio
SET	Social Engineering Toolkit
SF	Scale Factor
SG	Smart Grid
SGAM	Smart Grid Architecture Model (CEN, ETSI)
SGIRM	Smart Grid Interoperability Reference Model
SIEM	Security and Event Management System
SIL	Software in the Loop
SNTP	Simple Network Time Protocol
SSF	Scalable Simulation Framework
SU	Secondary Unit
SV	Sampled Value
T&D	Transmission and Distribution
TCP	Transmission Control Protocol
TDF	Time Dilation Factor
TDL	Time Definition Language
TLS	Transport Layer Security
TOU	Time-of-Use
UML	Unified Modelling Language
V2G	Vehicle to Grid
VE	Virtual Environments
VM	Virtual Machine
VPN	Virtual Private Network
VPP	Virtual Power Plant
VPST	Virtual Power System Testbed
VT	Voltage Transformer
WAMS	Wide Area Monitoring System
WP	Work Package
XML	Extensible Markup Language

Copyright Disclaimer

This report represents an educational course, and hence contains material sourced from various references and sources, whose respective authors hold the rights to the original content. The authors of this report do not claim ownership of the material used within but have utilized it solely for educational purposes and to provide proper attribution to the original creators.

The authors have made every effort to ensure that all sources and references are appropriately cited and acknowledged at the beginning of each chapter/section of this report. All text, paraphrased information, images, graphs, tables, and other elements are identified with the respective sources.

The inclusion of this material does not indicate endorsement or affiliation with the authors or the educational institution with which they are associated. This report is intended for educational purposes only and should not be reproduced or distributed for commercial gain without obtaining explicit permission from the original copyright holders.

By reading and utilizing this report, you acknowledge and agree to respect the rights of the original authors and the intellectual property contained herein.

Executive Summary

In brief, this report builds on the results of previous reports of Cybersecurity Curricula Recommendations in Smart Grids (CC-RSG) project, to provide a Massive Open Online Course (MOOC) for real-time physical cybersecurity systems. The course aims at filling the found gaps in Cyber-physical Systems (CPS), as well as building skills around real-time simulation systems that are used for high-speed simulations, testing of Critical Infrastructure (CI), and evaluating security operations and functions. To design this course, different approaches have been investigated and tested, thus to find the most suitable methods that serve the purpose of creating the course and filling the skill gaps that were highlighted. Accordingly, this course makes use of the following approaches: Flipped learning, Simulations, and Virtual and remote labs. These approaches were carefully chosen based on their capabilities for providing means for interaction, building analytical and problem-solving skills, as well as promoting hands-on skills.

The main contributions of this report are:

1. Providing the theoretical background regarding real time simulation systems
2. Building hands-on skills on using real time simulation
3. Introducing and applying use cases, scenarios, and exercises for practice and training purposes.

In more details, this report is the outcome of Work Package (WP) 3, “Designing, Developing, and Piloting” of the project Cybersecurity Curricula Recommendations in Smart Grids (CC-RSG). The goal of this work package is to develop educational content that matches with the requirements of modern Cyber Physical Systems (CPS) in a way that can fill the gaps identified in previous work packages, WP1 & WP2. Moreover, WP3 acts as a base for generalization that can be used to develop further content related to this field, and also to support the development of the Body of Knowledge (BoK) and implementation roadmap of WP4. In this report, theoretical and foundational knowledge have been taken care of, by furnishing strong grounds and providing the base for knowledge about smart grids, smart grids communication protocols, and cybersecurity in smart grids. In addition, real-time simulation systems, since being identified as one of the niche and most effective tools for experimenting, testing and learning about critical infrastructure and smart grids, has been given special consideration. Accordingly, many concepts around real-time simulation have been presented. Moreover, the course not only focuses on the theoretical knowledge, but it surpasses to cover the hands-on skills required by the industry. Thus, four exercises were built around selected industrial scenarios, and were implemented through a physical Real-Time Simulator system.

The main objective of this course is to provide participants with the knowledge required to embark on the field of CPS, with special emphasis on cybersecurity in smart grids.

Finally, this report acts as the teacher’s handbook of the covered topic. Thus, we tried to gather all information needed to cover the topic of cybersecurity in smart grids in this report. To use this report, teachers/instructors are advised to use the material here to design presentations and handouts, also since the emphasis is on flipped-learning, the design of pre-recorded videos is highly recommended. In this report, we also provided a rich set of pre-reading material, thus also cover different topics and perspectives we could not come across in the main learning material. Lastly, to support the creating of

MOOC, all processes related to learning and knowledge dissemination should be automated. That is, planning, assessment and evaluation. In this way, the delivery of this course will leave room for course providers to focus more on other aspects related to retention, interaction, and communication.

COURSE SYLLABUS

Course Description and Goals

Prerequisites: What should you know before starting this course?

1. This course is designed for both senior university students in the fields of Cybersecurity, Smart Gird, Cyber-physical systems, and related fields. The course is also suitable to professionals in these fields who want to update their skills and learn more about real-time physical cybersecurity.
2. Basic knowledge or previous courses in the fields of smart grid and cybersecurity is a must to have before enrolling into this course.

Learning Outcomes: What should you know after completing this course?

After the course, the participant builds a thorough understanding of cyber physical systems and the role of real-time simulator systems for testing and protection of cyber physical systems against cyber incidents.

During the course, the participant is introduced to fundamentals of cyber physical systems, i.e., models and standards; smart grid infrastructure, standards, and communication protocols, i.e. IEC 61850; NIST cyber security framework; operational security techniques, threats, vulnerabilities, attacks, and security analysis; risk management, standards as IEC 62351, policies and best practices; and real-time simulator systems and tools.

In terms of **management skills**, the participant is introduced to concepts of resilience and service continuity planning, risk management strategies, and integrated security management model.

The participant is also expected to build **social skills** regarding cybersecurity and related matters such as business, legalization, and privacy.

Main outcomes are seen in participants' ability to apply real-time physical cybersecurity concepts in modelling, testing, planning, and threat mitigation.

With regard to **general skills**, the participant develops problem-solving, decision-making, and critical-thinking skills.

Course Content: What topics are taught in this course?

Cyber-physical systems fundamentals, Smart grid communication system, Real-time simulation systems and tools, Cyber security fundamentals, Threats and attacks, Operational security, ISO/IEC 61850 standard, ISO/IEC 62351 standard, NIST cyber security framework, Best practices in the fields of cyber security and smart grid, Resilience and service continuity, Cases and scenarios on cybersecurity, Cases and exercises on a real-time simulator.

How can you deepen your knowledge in this field after completing the course?

This course provides upper-intermediate to advanced knowledge and skills in the fields of cybersecurity in smart grids and the use of real-time simulator, which is required to pursue other advanced courses in the same fields. The course is beneficial for areas such as technical, management and planning, service continuity, risk management and vulnerability assessment.

How to complete the course?

Teaching methods and Time allocation

The course follows the theme of a Massive Open Online Course MOOC to be offered to as many participants as possible regardless of their geographical location, experience, or academic level (as long as meeting with the prerequisites). In this theme, different techniques and approaches are used. Here the approaches of Flipped learning and remote and virtual labs are followed, since they provide the best tools to achieve the goals of this course from building knowledge to experimenting and upgrading hands-on skills.

The course is given as an online course and will consist of the following:

Lectures / Handouts: 40 h	30.75%
Labs, Exercises, and Selected Articles: 50 h	38.50%
Assignments, Practice tasks, and Reports: 40 h	30.75%
Total: 130 hours	100%

135 hours workload

Lectures / Handouts

Time allocated: 40 hours

- Lectures and handouts cover the main knowledge targeted by this course
- Lectures and handouts come in the form of recorded videos **and/or** presentations

Labs, Exercises, and Selected Articles

Time allocated: 50 hours

- Selected articles are distributed, to complete the learning objectives targeted by the course
- Exercises based on study cases and scenarios are provided, to promote analytical and problem-solving skills
- Lab sessions are given remotely or virtually. In these sessions, students connect to a real-time simulator environment to model, plan, test and practice given cases and scenarios.

Assignments, Practice tasks, and Reports

Time allocated: 40 hours

- Assignments and practice tasks are used to check the level at which participants achieve the learning outcomes
- One assignment is given per module
- A short essay report is required to ensure wholeness and understanding of the course's topics, and to reflect on analytical skills.

Assessment methods

- Assignments and practice tasks: **70%**
- Essay report and Documentation: **30%**

Your Role

The responsibility of the participant is to follow the instructions of the course, and to keep track of course content.

Literature

1. Martín, José L. Risco, Saurabh Mittal, and Tuncer Ören, eds. Simulation for Cyber-Physical Systems Engineering: A Cloud-Based Context. Springer Nature, 2020. Taha, Walid M., Abd-Elhamid M. Taha, and Johan Thunberg. Cyber-Physical Systems: A Model-Based Approach. Springer Nature, 2021.
   < https://library.oapen.org/handle/20.500.12657/41754>
2. Hehenberger, Peter, et al. "Design, modelling, simulation and integration of cyber physical systems: Methods and applications." Computers in Industry 82 (2016): 273-289.
3. Liu, Ren, et al. "Analyzing the cyber-physical impact of cyber events on the power grid." IEEE Transactions on Smart Grid 6.5 (2015): 2444-2453.
4. Eidson, John C., et al. "Distributed real-time software for cyber–physical systems." Proceedings of the IEEE 100.1 (2011): 45-59.
5. Poudel, Shiva, Zhen Ni, and Naresh Malla. "Real-time cyber physical system testbed for power system security and control." International Journal of Electrical Power & Energy Systems 90 (2017): 124-133.
6. Faure, Cyril, et al. "Methods for real-time simulation of Cyber-Physical Systems: application to automotive domain." 2011 7th International Wireless Communications and Mobile Computing Conference. IEEE, 2011.
7. Chen, Bo, et al. "Implementing a real-time cyber-physical system test bed in RTDS and OPNET." 2014 North American Power Symposium (NAPS). IEEE, 2014.
8. Chen, Bo, et al. "Implementing attacks for modbus/TCP protocol in a real-time cyber physical system test bed." 2015 IEEE International Workshop Technical Committee on Communications Quality and Reliability (CQR). IEEE, 2015.

9. Vellaithurai, Ceeman B., Saugata S. Biswas, and Anurag K. Srivastava. "Development and application of a real-time test bed for cyber–physical system." IEEE Systems Journal 11.4 (2015): 2192-2203.
10. Brunner, Christoph. "IEC 61850 for power system communication." 2008 IEEE/PES Transmission and Distribution Conference and Exposition. IEEE, 2008.
11. Cleveland, Frances. "Iec tc57 wg15: Iec 62351 security standards for the power system information infrastructure." White Paper (2012).
12. Stine, Kevin M., Kim Quill, and Gregory A. Witte. "Framework for improving critical infrastructure cybersecurity." NIST. (2014).
13. Bian, D., et al. "Real-time co-simulation platform using OPAL-RT and OPNET for analyzing smart grid performance." 2015 IEEE Power & Energy Society General Meeting. IEEE, 2015.
14. Singh, Shiv Kumar, et al. "Development of dynamic test cases in OPAL-RT real-time power system simulator." 2014 Eighteenth National Power Systems Conference (NPSC). IEEE, 2014.
15. Chlela, Martine, et al. "Real-time testing platform for microgrid controllers against false data injection cybersecurity attacks." 2016 IEEE Power and Energy Society General Meeting (PESGM). IEEE, 2016.

User manuals:

16. https://www.opal-rt.com/wp-content/uploads/2016/09/OP7817_User-Manual.pdf
17. https://www.opal-rt.com/wp-content/uploads/2016/07/OP5360-2-User-Manual.pdf
18. https://manualzz.com/doc/7255922/op5600-hil-box-user-manual---opal-rt
19. https://manualzz.com/doc/30208071/op5607-user-guide---opal-rt

PART ONE:

Smart Grids, Cyber Security & Grid Protocols

1. Fundamentals of Smart Grid Systems

Goals	To learn about: 1. Smart Grid infrastructure 2. Cyber-physical systems
Pre-reading material	1. Blumsack, Seth, and Alisha Fernandez. "Ready or not, here comes the smart grid!." Energy 37.1 (2012): 61-68. 2. Fang, Xi, et al. "Smart grid—The new and improved power grid: A survey." IEEE communications surveys & tutorials 14.4 (2011): 944-980. 3. Yu, Xinghuo, and Yusheng Xue. "Smart grids: A cyber–physical systems perspective." Proceedings of the IEEE 104.5 (2016): 1058-1070. 4. Shafiullah, G. M., et al. "Smart grid for a sustainable future." Smart Grid and Renewable Energy 4.1 (2013): 23-34. 5. Kenneth C.. Budka, Jayant G.. Deshpande, and Marina Thottan. Communication Networks for Smart Grids: Making Smart Grid Real. Springer Verlag, 2014.
Main material resources	Smart Grids Infrastructure Technology and Solutions, Stuart Borlase Borlase, Stuart, ed. Smart grids: infrastructure, technology, and solutions. CRC press, 2017.
Hours assigned	
Assignment criteria	1. Pre-lecture MCQ evaluation set 2. Post-lecture group discussion followed by a short essay
Done by	University of OFFIS / University of Oldenburg

1.1. What is a smart grid?

The term "smart grid" has been widely used in the energy industry, and different stakeholders have varying interpretations of the meaning. Depending on the perspective of utilities, vendors, consultants, academics, or consumers, the definition of a smart grid can differ. However, it is apparent that the concept represents a significant change for the whole electricity domain including generation, transmission, distribution, and delivering electricity to end users. A modern smart grid includes sustainable concepts that leverage proven, cleaner, and cost-effective technologies available today or in ongoing development. By incorporating these technologies, the smart grid improves energy systems effectively and economically, thus achieving sustainability

The debate over whether or not the electric grid can be considered "dumb" has been ongoing. In fact, the traditional grid is operated by complex software programs with a proprietary solution, and automation routines, and protected by microprocessor-based relays. While it is true that some parts of the world have made significant improvements in modernizing their electric infrastructure but the overall evolution of the grid has been slow. This could be from slow economic growth, passive regulation, and more. Despite these obstacles, there is a growing need for more advanced capabilities and greater integration to address energy demand, security, and environmental challenges. As a result, a need for a smarter grid is gaining momentum toward more efficient and sustainable approaches to generating, delivering, and using energy.

The electricity infrastructure is more capital-intensive, and developments in telecommunications processing and IT devices have far developed beyond electricity network modernization. Therefore, a smarter grid should focus on providing more services to consumers than merely reducing billing costs. The goal is not to do things differently from current practices but to do them smartly by developing and introducing more advanced capabilities focusing on the system value.

This can be done by sharing communications infrastructures, filling in product gaps, and leveraging existing technologies to a much greater extent. By driving towards a higher level of integration, the smart grid can achieve synergies across enterprise applications. Ultimately, a smart grid represents a sustainable approach to the energy transition that can meet the challenges of the present and the future.

1.1.1. What constitutes a smart grid?

Most smart grid solutions today focus on responses to metering requirements and use a pilot program to showcase a technology under the smart grid banner. In the broader sense, smart grid is not an off-the-shelf product which can be installed and operated the next day. Instead, it is an integrated set of technologies for cost savings in capital expenditures, operation and maintenance costs by providing customer and societal benefits. Figure 1.1 shows a typical example of a smart grid.

As illustrated in Figure 1.1, a smart grid is a concept for integrating electrical and communications domains with advanced process automation and information technologies to upgrade the existing electrical networks. It represents a sustainable perspective change of utilities, politicians, customers and other participating industries, on the way how to deliver better and cleaner electricity and provide new services. This concept involves integrating Information and Communication Technologies (ICT) to change the way for control and operations of the utilities towards real-time capabilities. The set of solutions that will provide these benefits is vast. At the same time, all utilities and their consumers may not easily benefit from significant smart grid development since new solutions are required only if needed and if operational strategies and benefits are prominent. Moreover, the growth of global, regional, and national scales will serve as the cornerstones for investments in smart grid infrastructure and which accelerate a high need for integrating communications and information technologies. One of the main drivers is national policy (and state government policy) including supportive incentives to speed up smart grid development.

The concept of a smart grid has many definitions and interpretations, which depend on the specific country and region and the various industry stakeholders' drivers and desirable outcomes and benefits. A preferred view of the smart grid may not be what it is but what it does and how it benefits utilities, consumers, the environment, and the economy. The European Technology Platform (comprising European stakeholders and the surrounding research community) defines "a Smart Grid as an electricity network that can intelligently integrate the actions of all users connected to it i.e., generators, consumers and those that do both, to efficiently deliver sustainable, economical and secure electricity supply." The U.S. Department of Energy (DOE) views the smart grid as a futuristic grid where every customer and node is monitored and controlled through a fully automated power delivery network. This network will ensure a two-way flow of information and electricity across power plants, appliances, and all points in between. Furthermore, the U.S. Electric Power Research Institute (EPRI) defines Smart Grid as a modernization that optimizes the operation of interconnected elements including, central and distributed generators, industrial users, distribution systems, building automation systems, energy storage installations, and end-use consumers and their appliances.

By stakeholder-driven definitions, smart grid should refer to the entire power grid from generation, through Transmission and Distribution (T&D) infrastructure throughout consumers. Smart grid essentially aims to modernize the existing grid to be more intelligent. It builds on the existing infrastructure and provides a wide range of applications to leverage benefits across applications and remove the major barriers of silos of organizational processes.

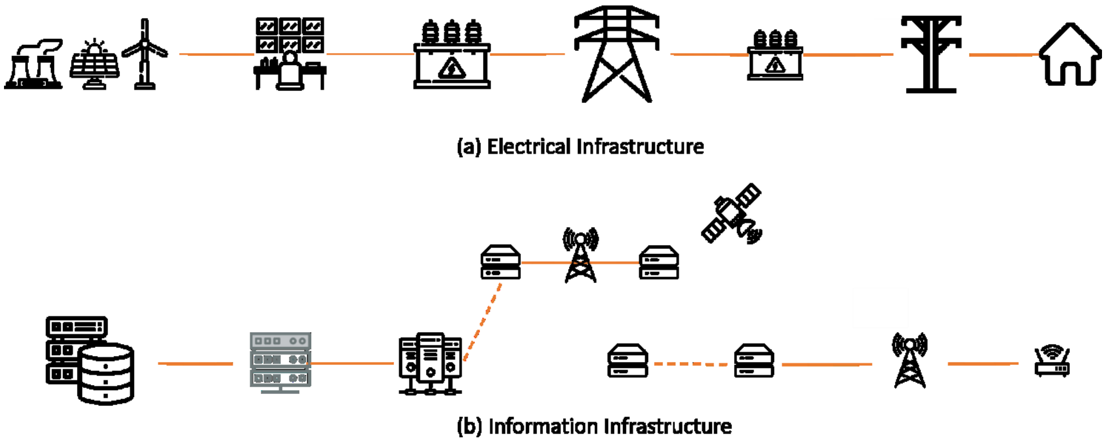


Figure 1.1. A smart gird consists of both electrical and information infrastructure

1.1.2. Smart Grid Drivers

A vision from several stakeholders should be aligned to fully modernize today's grid. A number of key factors that can drive smart grid development are:

- Policy and Legislative Drivers
  - Electric market rules that create comparability and monetize benefits
  - Electricity pricing and access to enable smart grid options
  - State regulations to allow smart grid deferral of capital and operating costs
  - Compatible Federal and state policies to enable full integration of smart grid benefits
- Economic Competitiveness
  - Creating new businesses and new business models and adding "green" jobs
  - Technology regionalization
  - Alleviate the challenge of a drain of technical resources in an aging workforce
- Energy Reliability and Security
  - Improve reliability through decreased outage duration and frequency
  - Reduce labor costs, such as manual meter reading and field maintenance, etc.
  - Reduce nonlabor costs, such as the use of field service vehicles, insurance, damage, etc.
  - Reduce T&D system delivery losses through improved system planning and asset management
  - Protect revenues with improved billing accuracy, prevention and detection of theft and fraud
  - Provide new sources of revenue with consumer programs, such as energy management
  - Defer capital expenditures as a result of increased grid efficiencies and reduced generation requirements
  - Fulfill national security objectives
  - Improve wholesale market efficiency
- Customer Empowerment
  - Respond to consumer demand for sustainable energy resources
  - Respond to customers increasing demand for uninterruptible power
  - Empower customers s that they have more control over their own energy usage with minimal compromise in their lifestyle
  - Facilitate performance-based rate behavior
- Environmental Sustainability
  - Response to governmental mandates
  - Support the addition of renewable and Distributed Generation (DG) to the grid

Many of these drivers however are country and region specific and differ according to unique governmental, economic, societal, and technical characteristics

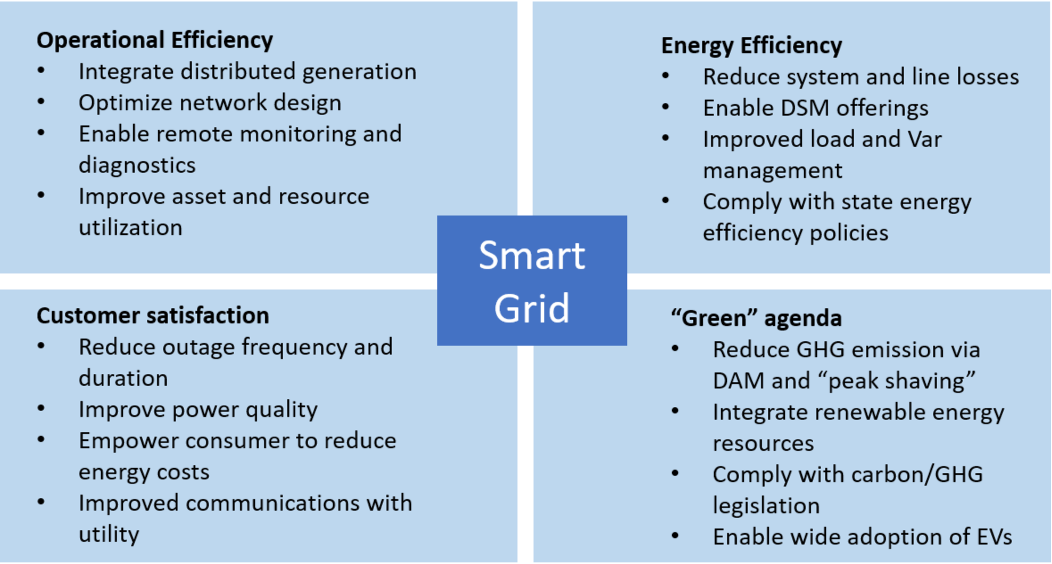
1.1.3. Smart Grid Benefits

The smart grid provides enterprise-wide solutions that deliver far-reaching benefits for both utilities and their end customers than being a simple business case. Utilities that adopt smart grid technologies can reap significant benefits in reduced capital and operating costs, improved power quality, increased customer satisfaction, and a positive environmental impact. In Figure 1.2 some of the smart grid benefits are presented.

Environmental, health, and other social benefits of the smart grid can be realized, if the grid is designed to capture them. A well-designed smart grid can deliver significant additional benefits, which can give return of investment quickly. Consumers will benefit from reduced bills and able to manage their electricity usage according to the price. For consumers, a smart grid will provide real-time information and pricing, which has the effect of conserving energy 5 to 15%. Another benefit will be in cutting peak demand and expanding demand response, which provides greater control. Organizations will benefit from new opportunities to provide energy services—from storage at substations to behind-the-meter "energy applications". Communities will enjoy greater cleaner energy, as they rely increasingly on distributed energy resources in their own backyards.

Moreover, the smart grid has the potential to drastically reduce costly damage to the environment and public health - while increasing energy independence and security and creating new industries and jobs - by enabling:

- Increased reliance on clean, renewable energy, including DG resources
- Vastly improved efficiency of electricity production, transportation, and use, including the ability to shift demand to lower impact times and supply resources.
- Decarbonization of the transport sector.
- Reduced water impacts—wind, solar Photovoltaics (PVs), and demand side resources use very little or no water for power generation.



Operational Efficiency
- Integrate distributed generation
- Optimize network design
- Enable remote monitoring and diagnostics
- Improve asset and resource utilization

Energy Efficiency
- Reduce system and line losses
- Enable DSM offerings
- Improved load and Var management
- Comply with state energy efficiency policies

Smart Grid

Customer satisfaction
- Reduce outage frequency and duration
- Improve power quality
- Empower consumer to reduce energy costs
- Improved communications with utility

"Green" agenda
- Reduce GHG emission via DAM and "peak shaving"
- Integrate renewable energy resources
- Comply with carbon/GHG legislation
- Enable wide adoption of EVs

Figure 1.2. Smart grid benefits

1.1.4. Smart Grid Challenges

Though the transition to smart grids brings many benefits, it also comes associated with own challenges, for instance utility and regulatory challenges as stated below.

- Utility Industry challenges:
    - Generation and energy resource mix changes
    - Energy storage
    - Consumer demand management
    - Transmission expansion
    - New demands
    - New technical opportunities
        - Managing an increasing number of operating contingencies that differ from "system as design" expectations (e.g., in response to wind and solar variability)
        - Facilitating the introduction of intermittent renewable and distributed energy resources with limited controllability and dispatchability
        - Mitigating power quality issues (voltage and frequency variations) that cannot be readily addressed by conventional solutions
        - Integrating highly distributed, advanced control and operations logic into system operations
        - Developing sufficiently fast response capabilities for quickly developing disturbances
        - Operating systems reliably despite increasing volatility of generation and demand patterns,
        - given increasing wholesale market demand elasticity
        - Increasing the adaptability of advanced protection schemes to rapidly changing operational behavior (due to the intermittent nature of renewable and DG resources)
- Regulatory challenges
    - Policies
    - Governance
    - Security and privacy challenges

1.1.5. Smart Grid Architecture Model

In Figure 1.3, a typical smart grid model is shown. One of the major differences between the traditional grid and its recent smart version is that the smart grid is directly connected to markets, transmission, distribution, service providers, and consumers. This allows automated monitoring, control, and operation throughout the electricity supply chain.

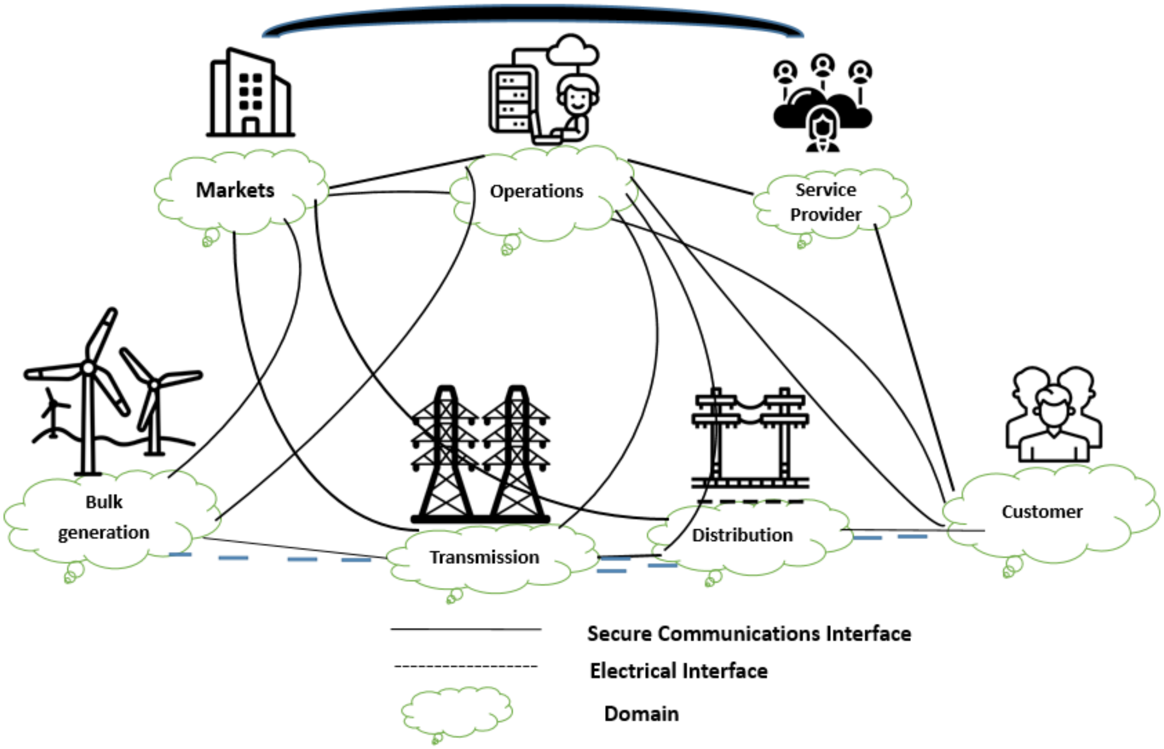


Figure 1.3. Smart grid reference model

1.2. Smart grid technologies

Figure 1.4 illustrates the steps to transform a conventional grid into a smart grid.

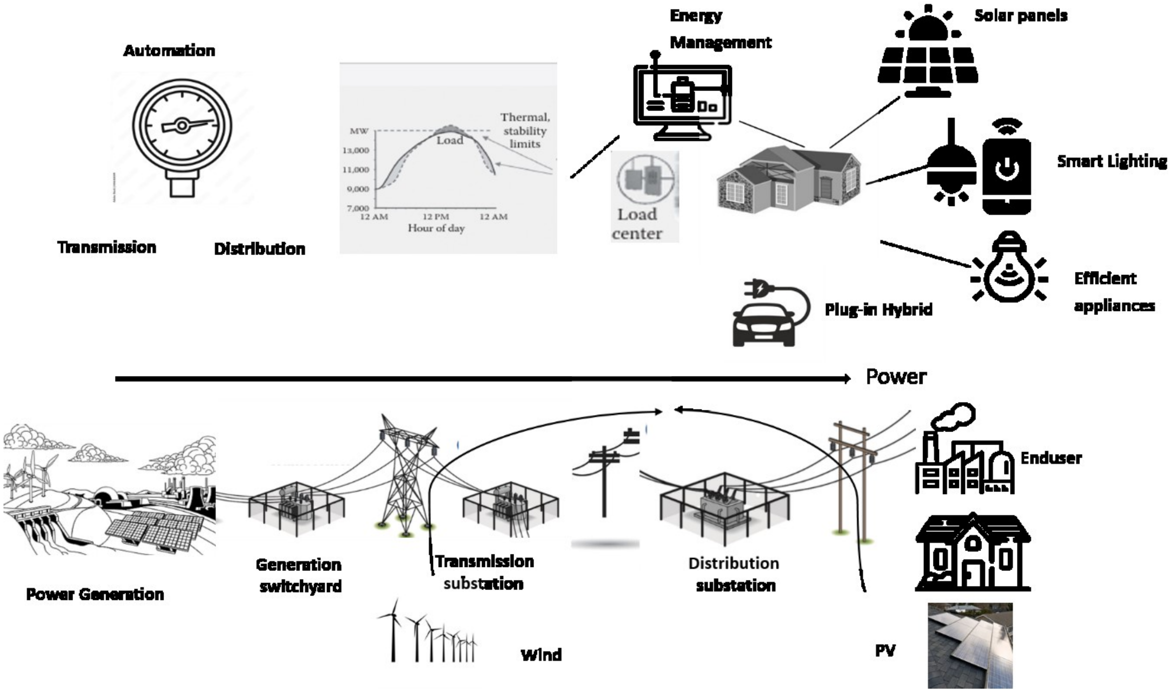


Figure 1.4. Transformation of a conventional grid to smart grid

1.2.1. Smart Grid Technology Framework

In Figure 1.5, smart grid technology framework is shown. Depending on the scenario and the components of the grid, different technologies and concepts are to be utilized. For instance, energy storage technologies are seen across generation, transmission, distribution, industrial and commercial units. In contrast, smart appliances are only seen across residential units but it could deliver services to the energy supplier e.g., via demand response and direct load control.

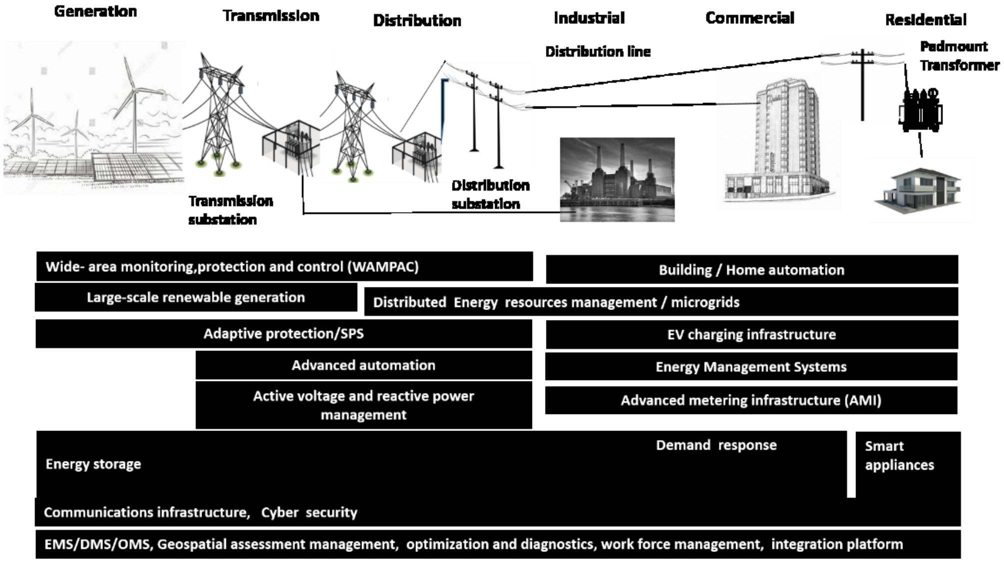
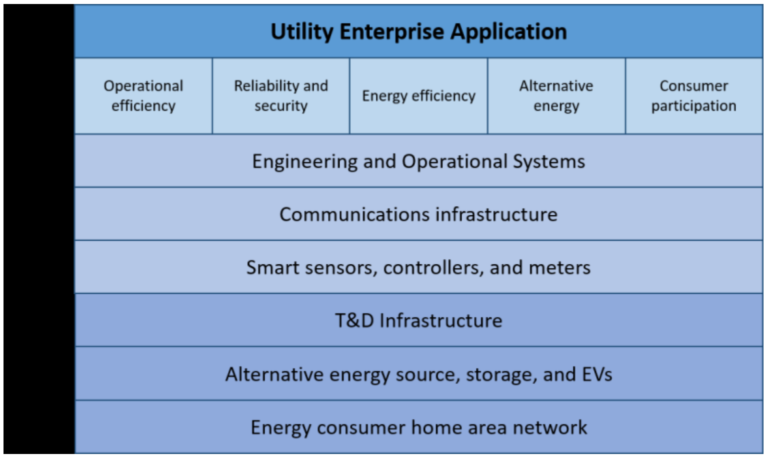

Generation
Transmission
Distribution
Industrial
Commercial
Residential
Distribution line
Padmount Transformer
Transmission substation
Distribution substation
Wide- area monitoring,protection and control (WAMPAC)
Building / Home automation
Large-scale renewable generation
Distributed Energy resources management / microgrids
Adaptive protection/SPS
EV charging infrastructure
Advanced automation
Energy Management Systems
Active voltage and reactive power management
Advanced metering infrastructure (AMI)
Energy storage
Demand response
Smart appliances
Communications infrastructure, Cyber security
EMS/DMS/OMS, Geospatial assessment management, optimization and diagnostics, work force management, integration platform

Figure 1.5. Smart grid technology framework

Figure 1.6 shows different infrastructures of smart gird. For example, conventional electrical infrastructure exists with its energy sources and T&D. However, new components are also added, e.g., storage infrastructure and Electric Vehicles (EV). Smart grid solutions offer many opportunities, ranging from operational efficiency, energy efficiency, to reliability and security, as well as consumer participation. Moreover, smart grids also allow for the continuous development of utility enterprise applications depending on the business models.



Utility Enterprise Application

Operational efficiency | Reliability and security | Energy efficiency | Alternative energy | Consumer participation

Engineering and Operational Systems

Communications infrastructure

Smart sensors, controllers, and meters

T&D Infrastructure

Alternative energy source, storage, and EVs

Energy consumer home area network

Figure 1.6. Smart grid layered infrastructure

1.2.2. Smart Grid Technology Functionalities and Capabilities

In Table 1.1, smart grid technology functionalities and capabilities of specific infrastructure components are described.

Table 1.1. Smart grid functionalities and capabilities

Component	Functionalities and capabilities	Description
Infrastructure	Communication and security	Underlying communications to support real-time operational and non-operational smart technology performance
	Embedded EVs, large-scale renewable generation, and Distributed Energy Resources (DER)	Integration of high penetration of EVs, large-scale renewable generation, and DERs can lead to situations in which the distribution network evolves from a "passive" system to one that actively responds to the various dynamics of the electric grid
Metering	Remote consumer price signals	The function that provides Time-of-Use (TOU) pricing information
	Granular energy consumption data/ information	Function with the ability to collect, store, and report customer energy consumption data/information for any required time intervals or near real-time
	Identify outage location and extent remotely	Metering function capable of sending signal when meter goes out and identifying itself after power restoration
	Remote connection, disconnection, reconnection	Function capable of remotely controlling "on" and "off" smart asset
	Remote configuration	Function capable of being remotely configured for functionality changes and firmware and software updates
	Optimize retailer cash flow	The ability of a retail energy service provider to manage its revenues through more effective cash collection and debt management
Grid	Embedded sensing, automation, protection, and control	Wide area system monitoring and advanced system analytics: a real-time, PMU-based grid monitoring system combined with advanced analytics consisting of intelligent fault and outage detection. PMU-based state estimation enables real-time dynamic and static system stability analysis, risk and margin evaluation, power system optimization, special protection schemes arming, etc., therefore providing planners/ system operators and engineering with capabilities to

		effectively predict possible severe grid disturbances leading to major power system outages and blackouts
	Advanced system operation	The modern grid relies on fully or semiautomated grid operation with a certain level of human intervention provided by the control centers. Advanced system operation tools are comprised of dynamic security assessment and Wide Area Monitoring System (WAMS) and control capabilities
	Advanced system management	Advanced asset management enables two-key smart grid capabilities. Optimum equipment performance leads to effective asset utilization. Maintenance efficiency of network components, attained by implementing condition- and performance-based maintenance
	Advanced system planning	Smart grid system planning considering real-time system impacts from large integration of renewable energy resources, high penetration of distributed generation, and chargeable and dischargeable EVs
	Intentional islanding (microgrids) and aggregated load and generation management, Virtual Power Plant (VPP)	Intentional islanding and/or grid-parallel operation of electric subsystem. Allows for optimum, multiple load/generation balancing to enable reliable and cost-effective operation.
Home/Building	Aggregated Demand Response (DR)	Aggregation of demand to reduce peak load and help balance the system more efficiently.
	EMS	Ability to control in-home appliances, distributed generation, and EVs to provide optimum energy consumption.

1.2.3. Smart Grid Resources, Storage, and Other Components

Smart energy resources and other means for renewable generation are, but not limited to:

- Solar PV
- Solar Thermal
- Wind
- Biomass and Biogas
- Geothermal

  - Wave Power
  - Hydro
  - Fuel Cells
  - Tidal Power
  - Combined Heat and Power

- Energy Storage means are, but not limited to:
  - Battery Energy Storage
  - Superconducting Magnetic Energy Storage
  - Flywheel Energy Storage
  - Compressed Air Energy Storage
  - Ultracapacitors
  - Pumped Hydro
  - Thermal Energy Storage

Finally, electric vehicles and similar applications play a vital role within the smart grid. Currently, EV technologies include, but not limited to:

  - Battery Electric Vehicles
  - Hybrid Electric Vehicles
  - Plug-in Hybrid Electric Vehicles
  - Vehicle to Grid (V2G)

1.2.4. Microgrids

A microgrid is an integrated energy system consisting of interconnected loads and DERs that can operate either connected to the grid or in an intentional island mode. The objective is to ensure local energy reliability, security, and efficiency. Microgrids exist in practice in utilities where they can support electricity supply when needed. Therefore, the microgrid is an effective alternative solution for the reinforcement of transmission lines. The drivers, benefits, and technologies of a microgrid are listed below.

- Microgrid drivers and resilience objectives:
  - Environmental incentives
  - Cost-effective access to electricity
  - Reliability
  - Energy security
  - Energy efficiency
  - Renewable energy implementation
  - Progress in energy storage technologies

- Microgrid benefits:
  - Economic: Load consumer benefits, microgeneration benefits, and network spending reduction
  - Environmental: Greenhouse gas reduction

    - Technical: Peak load shaving, reliability enhancement, voltage regulation, and energy loss reduction

- Technologies:
    - Sensors
    - Switches
    - Power electronics
    - Energy storage
    - Generators
    - Protection equipment
    - Metering
    - Local controllers
    - Real-time monitoring systems
    - The master microgrid management system
    - Interfaces with other systems

1.2.5. Smart Substations

An electrical substation is the central point of an electricity generation, transmission, and distribution system, where voltage is converted from high to low and vice versa using transformers. Electricity is transmitted from power plants to customers via many substations. There are different substations such as transmission substations, distribution substations, collector substations, and switching substations. The general functions of a substation include:

- Voltage transformation
- Connection point for transmission and distribution power lines
- Switchyard for electrical transmission and/or distribution system configuration
- Monitoring point for control center
- Protection of power lines and apparatus
- Communication with other substations and regional control center

The state-of-the-art technologies that operate in a smart substation are as follows:

- Intelligent Electronic Devices (IEDs) are microprocessor-based devices that have the capability to exchange data and control signals with another device such as IED, electronic meter, controller, Supervisory Control and Data Acquisition (SCADA), etc. over communication protocols. The IEDs are widely used in substations for different purposes such as protection, monitoring, control, and data acquisition functions in the electric network. In some cases, they are separately used to achieve individual functions, such as differential protection, distance protection, overcurrent protection, and monitoring. Moreover, there are also multifunctional IEDs that can perform several functionalities via user-interfacing functions on one hardware-software platform. The main advantages of multifunctional IEDs are that they are fully standardized communication protocols compatible and compact in size with various functions in one design. This allows a reduction in the size of the overall system and improvement of efficiency and robustness as well as providing extensible solutions. Therefore, the IEDs are the major component in the substation.

Furthermore, IED technology can help utilities improve reliability, gain operational efficiencies, and enable asset management programs including predictive maintenance, life extensions, and improved planning.

- Sensors provide the functionality to collect data from power equipment at the substation such as transformers, circuit breakers, and power lines. However, new sensors do not only allow data collection, but they also allow monitoring and control to be implemented simultaneously.

- SCADA is a widely used technology for providing control and management solutions in a variety of industries. SCADA systems must meet certain standards in the electric power business. An electric utility SCADA system's primary function is to collect real-time data from field devices, control the field equipment, and transfer the information to operating personnel.

- A Remote Terminal Unit (RTU) is used in particular for collecting analog and status telemetry data from field equipment and representing commands to the master station via the communication interface. Real-time monitoring and control of substations and feeders are generally in the range of 1–5 s. SCADA Systems are installed typically at a centralized location and include SCADA software, operator Graphical User Interface (GUI), engineering applications, historical software, and other components. Recent trends in SCADA include providing increased situational awareness through improved GUIs, data and information visualization, intelligent alarm processing, the utilization of thin clients and web-based clients, improved integration with other engineering and business systems, and enhanced security features.

Figure 1.7 below shows modern substations in the smart grid architecture

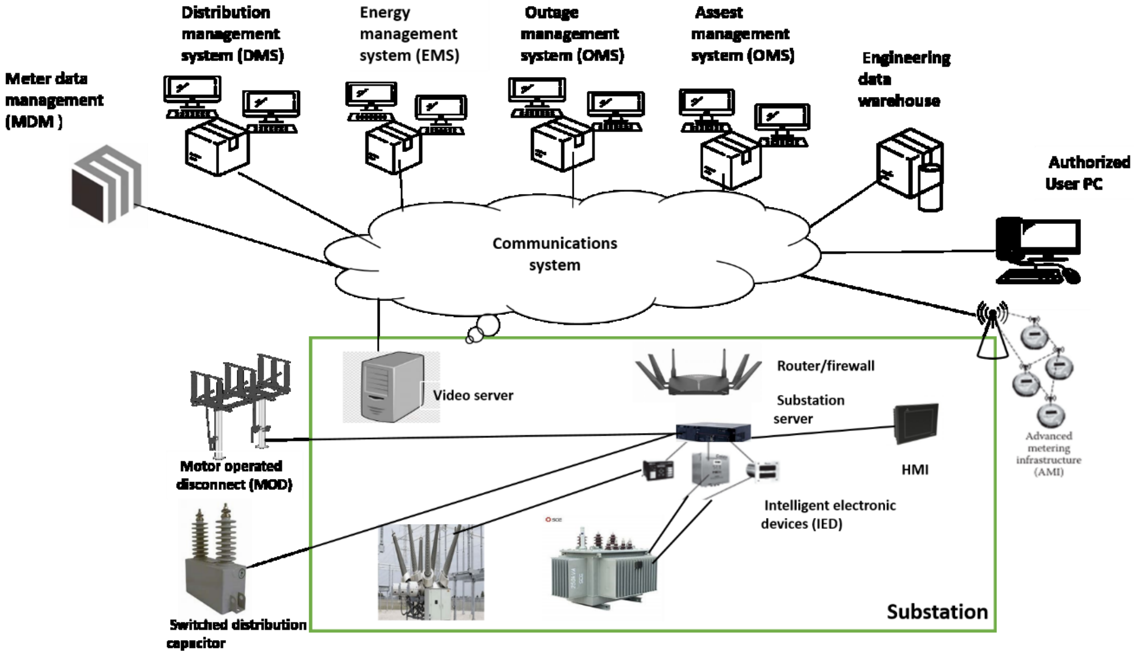

Distribution management system (DMS)
Energy management system (EMS)
Outage management system (OMS)
Assest management system (OMS)
Engineering data warehouse
Meter data management (MDM)
Authorized User PC
Communications system
Video server
Router/firewall
Substation server
HMI
Advanced metering infrastructure (AMI)
Motor operated disconnect (MOD)
Intelligent electronic devices (IED)
Switched distribution capacitor
Substation

Figure 1.7. The architecture of modern substations in the smart grid

1.2.6. Interoperability and IEC 61850

IEC 61850 is a standard for communication protocols for IEDs at substations. The standard specifies protocol-independent and standardized information models for various application domains in combination with abstract communications services, a standardized mapping to communications protocols, a supporting engineering process, and testing specifications. This standard allows standardized communication between IEDs located within electric utility facilities, such as power plants, substations, and feeders but also outside of these facilities such as wind farms, electric vehicles, storage systems, and meters. The standard also includes requirements for database configuration, object definition, file processing, and IED self-description methods. These requirements also provide modular integration or called “plug and play” capabilities. With IEC 61850, utilities will benefit from cost reductions in system design, substation wiring, redundant equipment, IED integration, configuration, testing, and commissioning. Additional cost savings will also be gained in training and system maintenance.

IEC 61850 has been identified as a cornerstone technology for field device communications and general device object data modelling. IEC 61850 Part 6 specifies a configuration language for systems based on industry standards. Peer-to-Peer (P2P) communication techniques such as the Generic Object-Oriented System Event (GOOSE) would reduce wiring between IEDs. P2P communication, along with Sampled Values (SVs) from sensors, will reduce the usage of wire throughout the substation, resulting in significant cost savings, more compact designs, and advanced and more flexible automation. The IEC 61850-based communications system can manage all of the data accessible at the process and station levels using high-speed Ethernet. The concepts and solutions offered by IEC 61850 are founded on three pillars:

- Interoperability: The ability of IED from one or several manufacturers to exchange information and use that information for their own functions
- Free configuration: The standard supports different philosophies and allows a free allocation of functions, for example, it will work equally well for centralized (RTU based) or decentralized (substation control system based) configurations
- Long-term stability: The standard shall be future proof, that is, it must be able to follow the progress in communications technology as well as evolving system requirements.

These pillars are achieved by creating a degree of abstraction that enables for the development of basically any solution using any configuration that is interoperable and stable in the long run.

IEC 61850 is divided into parts in which the information model of the substation equipment is specified. Models for primary devices such as circuit breakers and instrument transformers such as Current Transformers (CT) and Voltage Transformers (VT) are included in these information models. They also include models for secondary functions such as protection, control, measurement, metering, monitoring, and synchro phasors. It follows a layered approach, and in order to have access to the information contained in the information models, the standard defines protocol-independent, abstract communications services. These are described in part 7-2, such that the information models are coupled with communications services suited to the functionality making use of the models. This concept, which is independent of any communications protocol, is known as the Abstract Communications Service Interface (ACSI).

- Read and write data
- Control
- Reporting
- GOOSE
- SV transmission

In Figure 1.8, an abstract of IEC 61850 protocol is shown, with layered separation of applications and communication.

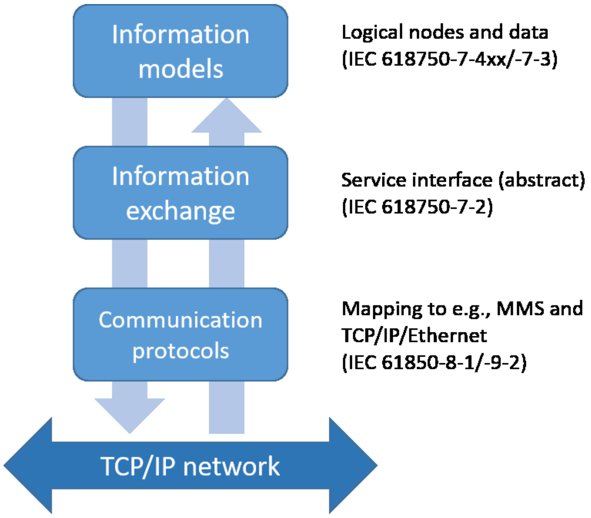


Figure 1.8. Separation of application and communications in IEC 61850

1.3. Challenges and success factors

Smart grid opens opportunities as well as challenges because of the interdependency of multi-domains which require multidisciplinary know-how skills. Utilities usually improve their services by deploying new technologies which increases the investment cost. From the private’s benefit point of view, new investment is economical if the benefit is greater than the investment cost considering its life cycle. However, this is not the case if the companies could not deal with risks for uncertain benefits in long term.

Policy and regulatory framework could support businesses by having a vision of the system value and economic scale i.e. societal benefit. Designing policy instruments should consider the whole system's benefit i.e. increasing society's welfare which can enhance win-win scenarios for all stakeholders. For example, an incentive to support green technology investment could accelerate low-carbon and grid modernization that could benefit the end-users. Policymakers should also monitor and modify policy instruments when the system value is fading away.

1.3.1. Challenge 1: Utility Organizational and Business Process Transformation

Organization’s mindset is a key to success for cohesive business transformation. There are many departments with different backgrounds and areas of work within the organization. Therefore, the transformation of businesses necessitates coordination and cooperation across organization boundaries to achieve the long–term vision. Figure 1.9 shows an example of relevant components for management changes toward business transformation. The first phase focuses on understanding the current and future business models, motivations, and visions. A road map and an action plan should be developed to define the tasks and responsibilities. In the first stage, communication and discussion among stakeholders are common to identify suitable approaches for implementation. Later on, lessons learned from the first phase can be modified and improved with corrective actions and continuous improvement to achieve the long-term vision.

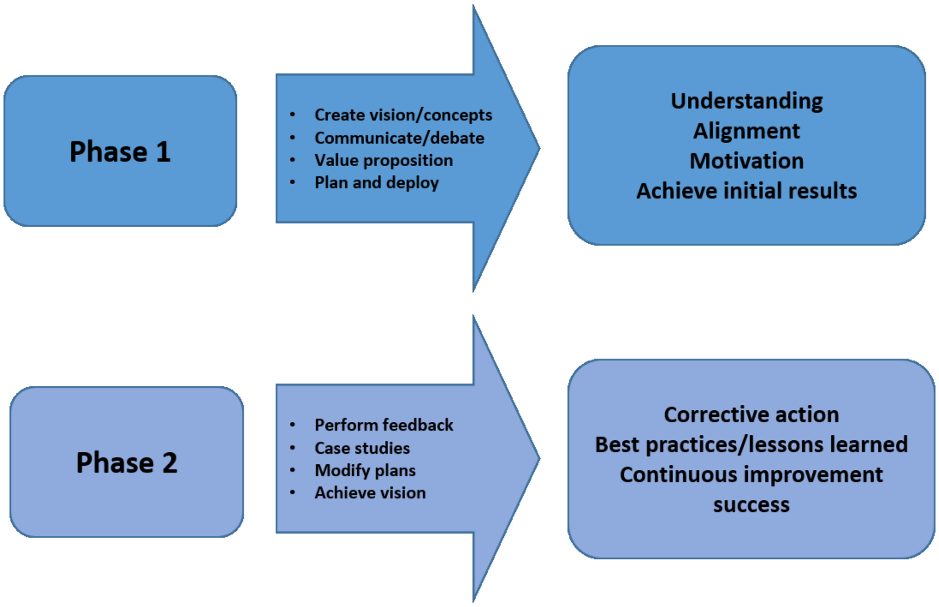


Figure 1.9. Components of managing change

1.3.2. Challenge 2: Convergence of Operations Technology and Information Technology

In the past, Operational Technology (OT) and Information Technology (IT) were separated which required more human supervision. The development of technologies has accelerated the convergence of OT and IT which can perform tasks automatically with less human supervision. This IT/OT convergence could be seen in every sector, especially for the proactive businesses that deployed innovative solutions. Organizational management is changed from an individual to a single overall perspective. This is because there is an interdependency across departments and their responsibilities. The management processes will make use of the technological benefits of building a network for communication and monitoring processes as real-time and two-way communications rather than top-down management approaches. This new management process in the organization mimics the smart grid control and management processes which enable effective monitoring and controls.

1.3.3. Challenge 3: Integrated System Approach

An integrated strategy is required for the smart grid development to the design and operation of generating, transmission, and distribution systems. Smart grid implementation necessitates a robust and flexible communication infrastructure. The smart grid's communication demands more requirements than the typical grid. Real-time data and information transmission necessitate minimal latency with robust redundancy in communication architectures. Back-office data processing and storage require communication support for distributed databases and processing facilities which must also support the system integrity protection to ensure security and dependability of operation and control.

1.3.4. Challenge 4: Smart Grid Integrative Approach

A smart gird is an integration of many components and applications from different fields and disciplines. Accordingly, the smart grid inherits all challenges coming from such an integration. In Figure 1.10, smart grid engineering with all integrative components is shown.

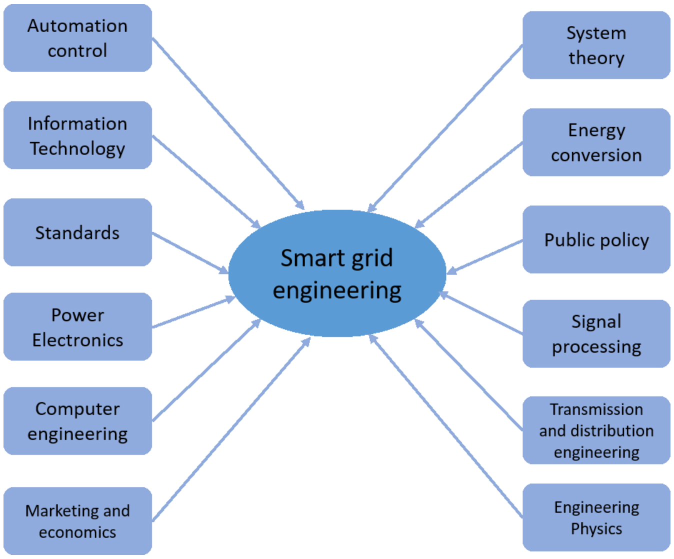


Figure 1.10. Smart grid integrative approach

1.3.5. Challenge 5: Cybersecurity

Traditional electricity grid is being modernized by integrating ICT technologies for real-time communication and control. Risks could be arise caused by a failure of device, unauthorized access, misconfigured components, human error, and etc. Therefore, an organization should implement the monitoring and countermeasures to tackle with the risks including vulnerabilities caused by attackers.
A communication network without monitoring and counter-measures could be at risk. An organization should implement cybersecurity practices to monitor, control, protect, and counter-act to the vulnerabilities. Security controls must be specified across each security domain which should be agreed-upon risk assessments considering company policies and any government regulation requirements. Furthermore, restrictions associated with existing legacy systems must be carried out in a way that does

not threaten corporate security. Another consideration is the possibility of extending an Internet Protocol (IP)-based network to the meter, which poses cyber vulnerabilities. The system architecture's structure should be well examined with additional innovative solutions such as using distributed intelligence to supervise a single point of failure.

1.3.6. Challenge 6: Data Privacy

Smart grid enables real-time exchange including sensitive data collected by e.g., Advanced Metering Infrastructure (AMI) which electricity consumption profile could be interpreted to consumer's lifestyle e.g. when they are at home. Consumer concern of their data privacy in which data is processed and who use it. Data privacy protection law can increase confidence for the consumer that their data are used only in within their consent. Company and organization must comply with the regulatory in practice to build the trust to consumer.

1.3.7. Success Factors

Seamless smart grid deployment does not only require the best technology but also other factors such as the organization's perspective, regulatory framework, and also consumer engagement. Preparedness for the organization's transformation is essential in the motivation, design, implementation, and monitoring.

From the business perspective, the investment is economical when the benefit is greater than the investment cost. One example could be seen in smart meter deployment where electricity suppliers foresee its benefit to reducing operational and maintenance costs by providing automated reading data and billing. This private value motivates businesses to invest in new technologies. The policy can play an active role to increase system value by developing a regulatory framework and providing incentives to increase societal welfare.

Demand Response (DR) can accelerate consumer behavior change in energy consumption with an incentive or dynamic pricing mechanism using smart meter technology. Additional direct load control under the DR program enables automated control by energy companies or aggregator, however, it could raise privacy concern that leads consumer to opt out of the program. A proactive regulatory framework and business model can ensure the sustainability of system value and consumer engagement.
Table 1.2 shows the challenges and key success factors for smart grid development.

Table 1.2. Challenges and success factors of smart grids

Dimension	Challenges	Key success factors
Policy factors	• Passive regulatory framework and instruments • Focusing only one dimension	• Proactive policy focusing on system value • Support technology and innovation • Clear roadmap for achieving its vision • Enhance interoperability between vendors e.g. standards

Societal factors	• Lack of awareness in organizations and consumer	• Increasing awareness and mindset of relevant stakeholders in the systems including end-users
Economic factors	• Absences and insignificant of incentives	• Providing and monitoring appropriate policy instrument mechanism to support and accelerate green technologies adoption and cost recovery • Considering system value rather than private value
Educational factors	• Outdated education curricular Insufficient innovative funding for Research and Development (R&D) projects	• Update curricular and working closely with industry for the skills needed • Providing budget with priority of R&D topics in the current challenges • Increase industry's expertise and skills

Other success factors include:

- Technology investment and innovation
- Consumer engagement and empowerment
- Vendor partnership and collaboration
- Standards development, compatibility, coordination, and acceleration
- Policy and regulations
- Industry expertise and skills
- Knowledge and education

1.4. Global Initiatives on Smart Grids in Europe

1.4.1. Lead Organizations

Organizations of the projects are split into the following categories:

- Energy companies (e.g., EDF energy)
- Distribution system operators (e.g., Enel Distribution)
- Transmission system operators
- Service providers (manufacturers, aggregators, retailers, IT companies, etc.)
- Universities, research centers, and public organizations

1.4.2. Project Category/Technology

Approximately 27% of the projects implemented in the smart meters; these projects involve the installation of about 40 million devices for a total investment of around €3 billion. In France and Finland, main budget contributed to smart meter projects. In France, the demonstration project "Pilot Linky" accounts for about 75% of the total budget, while in Finland the smart meters roll-out project by Fortum accounts for over 80% of the whole budget.

1.4.3. Project Examples

Integrated system projects represent about 34% of total projects and about 15% of the total budget. Most of the technologies are known, but their integration is a new challenge. A system of the system should be considered rather than an individual technology and application.

1.4.4. Country-specific Drivers and Benefits

Projects are not distributed evenly across Europe. The majority of initiatives are concentrated in a few countries: Denmark, Germany, Spain, and the United Kingdom which account for almost half of all projects. Denmark leads in R&D and demonstration projects which clearly shows a high penetration of renewables and distributed generation as well as upgrade its energy infrastructure in a country.

1.4.5. Scale

There are only a few outstanding countries for investing in smart grid development. The different pace at which smart grids are deployed across Europe could make trade and cooperation across national borders more difficult and jeopardize the achievement of the European Union's (EU) energy policy goals. The majority of projects are concentrated in the R&D and demonstration phases. Only 7 to 10% of initiatives have reached the deployment stage. R&D and demonstration projects account for lower fraction of the total budget, which these initiatives are small to medium in size.

1.5. Future of smart grid

Some smart grid business is expected to be further developed and implemented, particularly for the following:

- Systems operations (visualization, virtualization, and situational awareness)
- Physical and cybersecurity advances
- Data analytics
- Advanced metering (in states/provinces/countries with regulatory foresight and where the business case and customer education requirements have been made)
- Advanced communications networks with greater bandwidth and lower latency – serving as enablers for several smart grid components (including phasor measurement implementations at transmission substations; distribution network automation; and metering data acquisition)
- Utility and enterprise systems integration activities

1.5.1. Market Drivers and Enablers

Market uptake is expected to increase depending on vision and advantages. Each electrical market has its own market rule in which regulatory framework and commercial driver could enhance the smart grid development over time.

1.5.2. Technical Innovation

Real-time communication is essential for integrating DERs which require observable and controllable capabilities. Increasing of DERs motivates the management and control innovations such as forecasting and controlling variable renewable energy such as wind and solar resources. In addition, power quality control is becoming a challenge with variable renewable energies which are dependent on the weather condition. Integrating distributed advanced and automated control can enhance DERs integration and maintain power system security. In addition, power system planning should consider a proactive defense plan in the event of an emergency that could minimize blackout severity. Innovative solutions with intelligent algorithms and modular configuration can increase resilient power systems that could react the change in operational behavior, external factors, as well as the development of technology.

1.5.3. Policy and Regulatory Priorities and The Role of Electricity Markets

Smart grid technology adoption is dependent on the available regulatory frameworks. Policymakers should consider the increasing need for ICT for improving the reliability, security, and efficiency of the electric grid. Cybersecurity must be considered to ensure secure control and operation. Policy and regulatory framework can accelerate smart grid development and competitiveness. The interoperability of different smart grid applications requires agreement and/or standards as a way to regulate and control solutions from different vendors in a standardized manner. This also increases freedom of choice for the customers as well.

Moreover, smart technology can induce consumer engagement in participating in smart grid development such as an active consumers who can manage their electricity consumption or prosumer who can generate and consume electricity. Internet of Things (IoT) and smart electric appliances can also contribute to demand response from the end user via demand-side management programs e.g. direct load control. Electric vehicle adoption is increasing in low-voltage and medium-voltage electricity networks which could pose power quality concerns and increase business opportunities. The regulatory framework for integrating and controlling new smart grid applications should be up to date with the technology development status in the market. Increasing fair competitiveness is essential for small entrants by lowing market barriers to increase smart grid businesses in the market.

2. Cybersecurity and Operational Security in Smart Grid systems

Goals	To learn about: 1. Cybersecurity fundamentals 2. Operational security 3. Smart grid security
Pre-reading material	1. El Mrabet, Zakaria, et al. "Cyber-security in smart grid: Survey and challenges." Computers & Electrical Engineering 67 (2018): 469-482.Smart grid – The new and improved power grid: A survey 2. Gunduz, Muhammed Zekeriya, and Resul Das. "Cyber-security on smart grid: Threats and potential solutions." Computer networks 169 (2020): 107094. 3. Yaacoub, Jean-Paul A., et al. "Cyber-physical systems security: Limitations, issues and future trends." Microprocessors and Microsystems 77 (2020): 103201. 4. Mishra, Sakshi, et al. "Microgrid resilience: A holistic approach for assessing threats, identifying vulnerabilities, and designing corresponding mitigation strategies." Applied Energy 264 (2020): 114726.
Main material resources	Knapp, Eric D., and Raj Samani. Applied cyber security and the smart grid: implementing security controls into the modern power infrastructure. Newnes, 2013.
Hours assigned	
Assignment criteria	1. Pre-lecture MCQ evaluation set 2. Post-lecture group discussion followed by a short essay
Done by	National Technical University of Athens

2.1. Hacking the Grid

A common way to classify cyber-attacks against a power grid is to use the intent of the attacker. In most cases the intents include the denial of service (financial extortion might take place, or the grid might be sabotaged), the theft of information in order to gain an advantage (e.g. profit or reconnaissance) and others. In a similar way, cyber-attacks can be prevented, or their effects can be mitigated using the same classification i.e. by targeting the intent.

2.1.1. Theft of information

The information that can be reached through access into a power grid is very diverse and includes very valuable parts. This valuable information includes personal information of the consumers that is used mainly for billing purposes and/or customer support and the load consumption profile of the household from which an experienced operator can extract information such as the number of persons in a household, their daily habits etc. Also, the control systems (i.e. SCADA) can be used to provide an outline of the grid itself and help the intruders identify vulnerable attack points of the system. This information can also be used to influence the energy trading (e.g. by extracting the information about available generators or planned expansions).

2.1.2. Denial of service

Denial of Service (DoS) is one of the most common attacks, although it can be challenging to accomplish in today’s networks where modern operating systems and high bandwidth are used. Smart grids and SCADA systems were designed with the intent to not have publicly accessible points which is why no emphasis has been given to modernizing them. As a result, most of these systems are older systems that might be unpatched and outdated in combination with a large attack surface comparable to that of the Internet.

An added drawback is that unlike the internet no universal security measures are implemented such as those that Internet Service Providers (ISP) use for the Internet. This could pose particular risks to substation automation systems, where distributed denial-of-service attacks are perpetrated inbound over long-range wireless frequencies, or even through the advanced metering infrastructure.

The choice of communication medium that is used (cellular, satellite, radio) can also be exploited in the orchestration of a DoS attack. For example, a war-dialing attack in which a large list of telephone numbers is scanned automatically, can exhaust the bandwidth of the cellular service of metering infrastructure and be used as Denial of service-attack. Other examples of DoS attack specific to the Smart Grid include the spoofing of the Global Positioning System (GPS) time source in order to bring a line out of phase and trip protection circuits, or by faking an outage in order to make the EMS system to route power around a service area.

2.1.3. Manipulation of Service

During manipulation of service the service appears available to the users/operators but its operation has been altered or eliminated. An example can be spoofing GPS time synchronization where the readings of the phasors become desynchronized or even the Stuxnet attack which was rather complex.

2.1.4. Identifying a target

Cyber-attacks begin very often with identifying a target, and Smart Grids not only include a large number of targets themselves but also a large number of systems that can be used as beachheads in order to reach another target. Targets include Advanced Metering Infrastructure, Customer Relationship Management, Transmission and Distribution systems, SCADA, billing systems, EMS, Communication system etc. In order to exploit a target, the attacker can either scan the transmission and distribution infrastructure or use the enumeration functions of the automation system.

2.1.5. Vulnerability

When the target has been selected, the next step is to penetrate the key systems, exploiting the vulnerabilities of the SCADA protocols. The vulnerabilities can be device specific, protocol specific or can be more general and they can also be scaled up to cause even further harm. Most of the time they become known using reverse engineering and then they are shared with other people. Although many of these vulnerabilities are device specific, the continuous analysis of the systems creates a growing number of these vulnerabilities to become publicly available sometimes along with the method that is used to exploit it.

2.1.6. Attack Tools

- General tools include: Nmap for network scanning, the Social Engineering Toolkit (SET) for social engineering attacks, Nessus, Nexpose, and OpenVAS.
    - Frameworks for penetration testing and exploitation are also used, as Metasploit, which includes several auxiliary modules for scanning and enumerating the systems used by generation, transmission, and distribution SCADA, making it particularly relevant to Smart Grid cyber security.
- In addition to general tools, there are tools that are specific tools, as OPTIGUARD meter assessment toolkit, which can be used maliciously to target smart meters.

2.1.7. Attack Methods

- Common attack methods include:
    - Man-in-the-middle attack (MITM), where the attacker inserts himself between communicating devices and snoops the traffic between them.
    - Replay attacks, by capturing packets and replay them to inject a desired process command into the system.

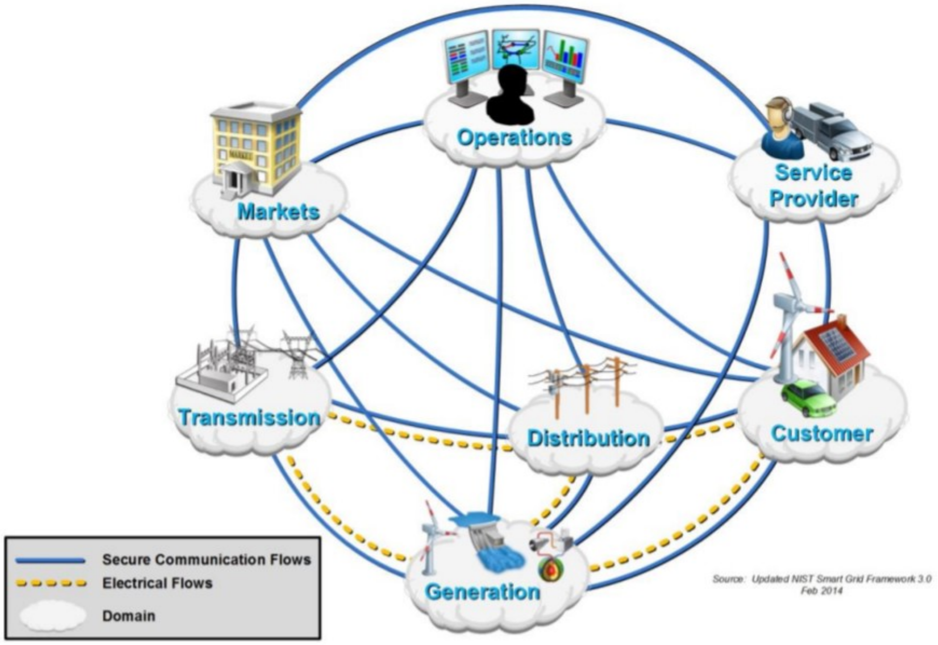
- Popping the Human Machine Interface (HMI) to obtain unauthorized command and control of a system, by manipulating controls via the console interface. This also can be done by remote access to the console.
- Blended attacks that involve more than single exploits against a single vulnerability
- Manipulating Phasor Measurement Units (PMU) to cause cascading outage.
- Attacking generation facilities from the grid, which is a sort of insider attack

2.2. Security models for SCADA, Industrial Control Systems (ICS), and Smart Grid

Several security models exist, and they cover different needs. In order to choose a model, the operator needs to first identify the assets and understand their function and interconnection in order to prioritize them, adding in their assessment the concepts of trust and criticality. After this procedure is finalized then the operator can effectively segment the Smart Grid into component enclaves, zones, domains, groups levels etc.

2.2.1. NISTIR 7628 Smart Grid Security Architecture

- In this model, smart grid is divided into actors and domains. Here is how secure communication could be done. In Figure 2.1, actors and domains of smart grid are shown. Also, in Figure 2.2, the logical reference model of these actors and domains is shown, with specific details within each domain.



Source: Updated NIST Smart Grid Framework 3.0 Feb 2014

Figure 2.1. High level view of the actors within each of the smart grid domains

This framework is used as a guide in order to identify the security requirements of the system. The steps to achieve this include defining the logical interface (Meter Data Management Systems (MDMS)), defining the logical interface categories (interface between back-office systems under common management authority) and choose the most important aspect to protect between Confidentiality, Integrity and Availability (CIA). The framework provides guidance that covers the following topics:

- Session lock
- Permitted actions
- User identification and authentication
- Security function isolation
- Denial of service protection
- Resource priority
- Communication integrity
- Confidentiality of information at rest
- Software and information integrity

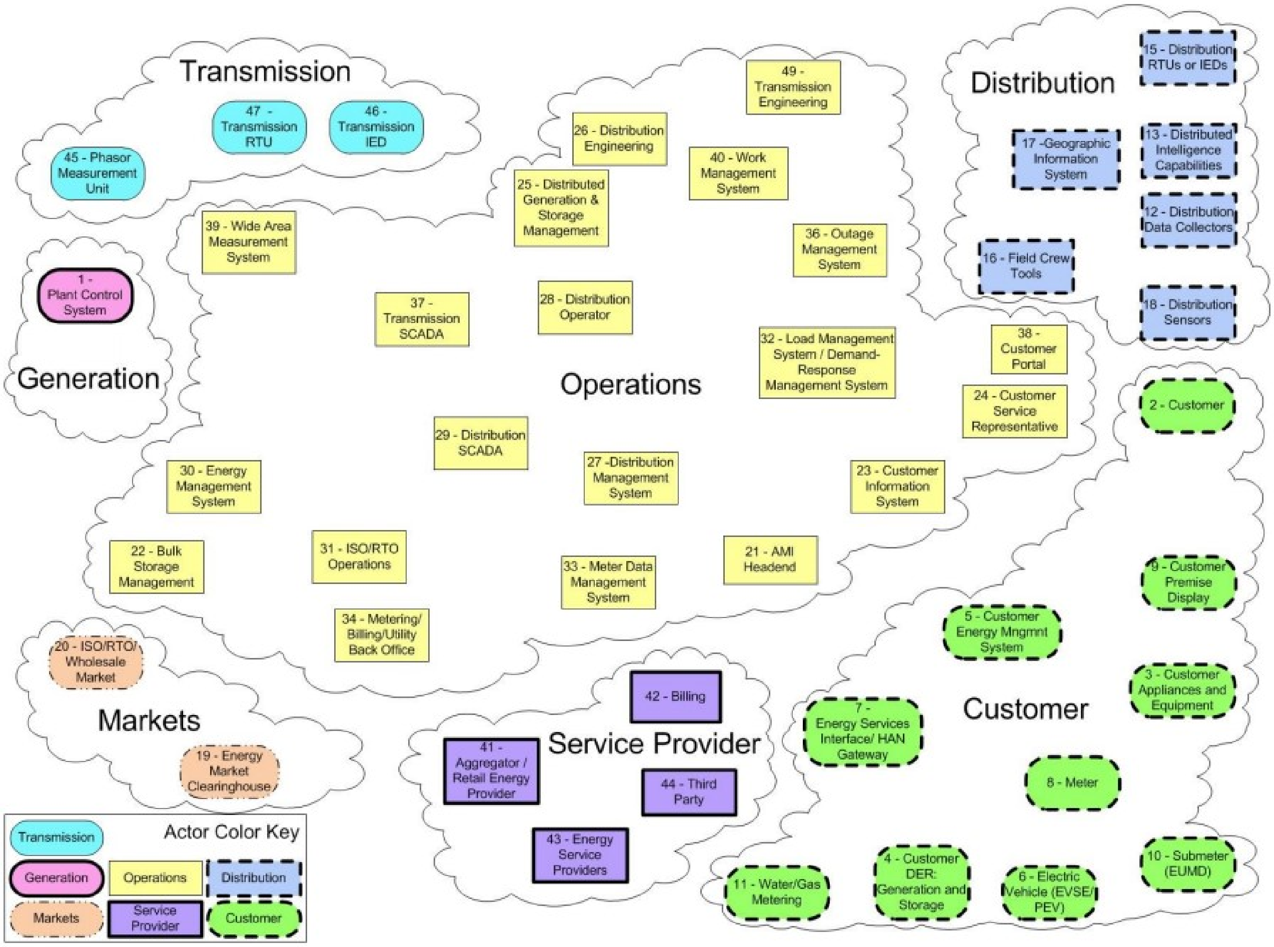

Transmission
47 - Transmission RTU
46 - Transmission IED
45 - Phasor Measurement Unit
Generation
1 - Plant Control System
Operations
49 - Transmission Engineering
26 - Distribution Engineering
40 - Work Management System
25 - Distributed Generation & Storage Management
39 - Wide Area Measurement System
36 - Outage Management System
37 - Transmission SCADA
28 - Distribution Operator
32 - Load Management System / Demand-Response Management System
38 - Customer Portal
24 - Customer Service Representative
29 - Distribution SCADA
27 -Distribution Management System
30 - Energy Management System
23 - Customer Information System
22 - Bulk Storage Management
31 - ISO/RTO Operations
33 - Meter Data Management System
21 - AMI Headend
34 - Metering/ Billing/Utility Back Office
Distribution
15 - Distribution RTUs or IEDs
17 -Geographic Information System
13 - Distributed Intelligence Capabilities
12 - Distribution Data Collectors
16 - Field Crew Tools
18 - Distribution Sensors
Customer
2 - Customer
9 - Customer Premise Display
5 - Customer Energy Mngmnt System
3 - Customer Appliances and Equipment
7 - Energy Services Interface/ HAN Gateway
8 - Meter
10 - Submeter (EUMD)
11 - Water/Gas Metering
4 - Customer DER: Generation and Storage
6 - Electric Vehicle (EVSE/ PEV)
Markets
20 - ISO/RTO/ Wholesale Market
19 - Energy Market Clearinghouse
Service Provider
42 - Billing
41 - Aggregator / Retail Energy Provider
44 - Third Party
43 - Energy Service Providers
Actor Color Key
Transmission
Generation
Operations
Distribution
Markets
Service Provider
Customer

Figure2.2. Logical reference model of smart grid actors and domains

2.2.2. EU M/490 and the SGCG reference architecture for the smart grid

Standardization Mandate M/490 is a mandate of the European Standardization Organizations (ESOs) to support European Smart Grid deployment and requires the provision of "a technical reference architecture, which will represent the functional information data, flows between the main domains and integrate several systems and subsystems architectures." Based on that mandate, SGAM framework was developed.

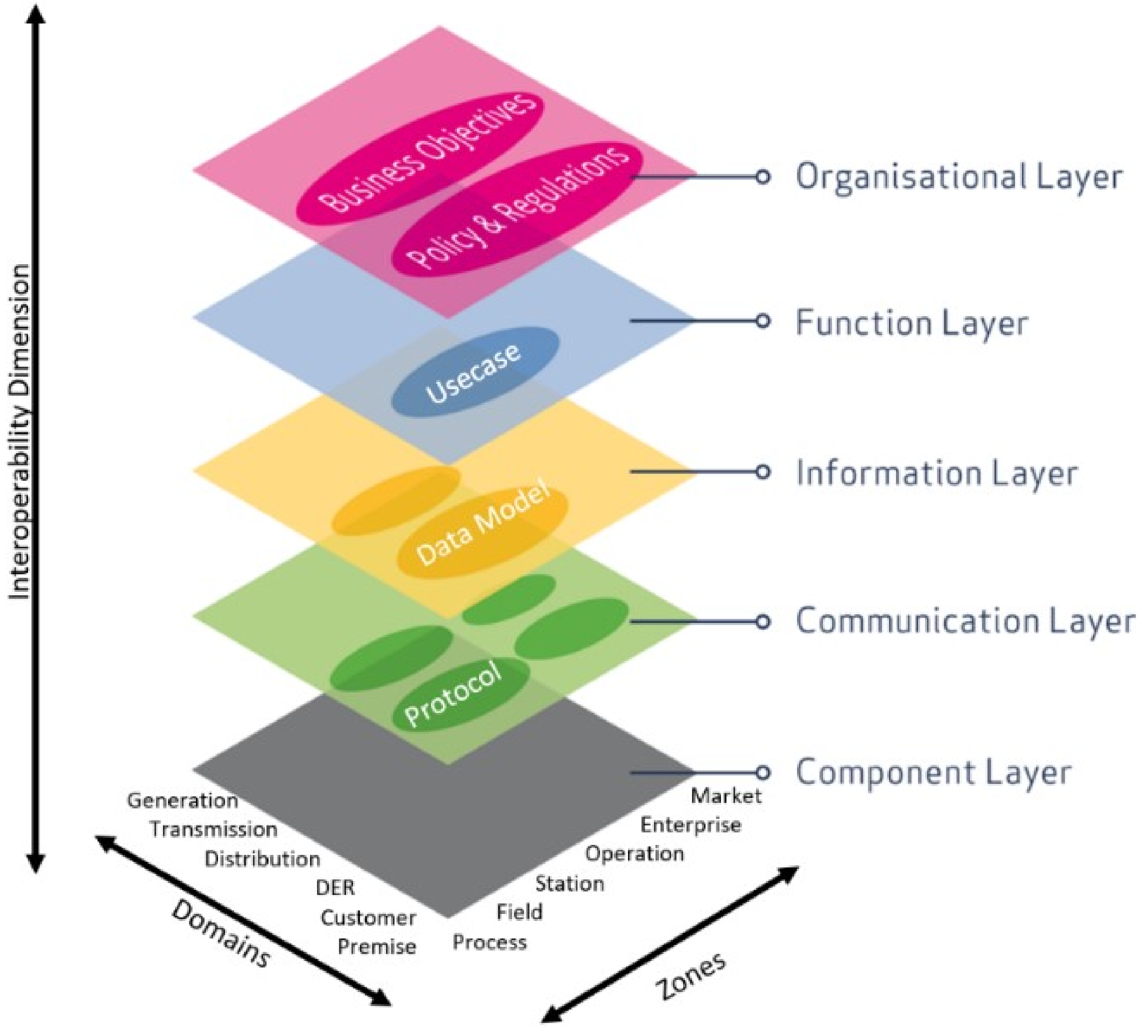


Figure 2.3. Smart Grids architecture model (SGAM) framework

SGAM – which is shown in Figure 2.3 - contains five layers (Business, Function, Information, Communication, and Component layers) and "allow[s] for a representation of interoperability viewpoints in a technology neutral manner, both for current implementation of the electrical grid as well as future implementations of the Smart Grid." This model builds on NISTIR 7628 by enabling for three-dimensional representation of the interoperability with the use of the five layers. It is also important to mention that it defines a new domain about DERs and it incorporates the Grid-Wise Architecture Council (GWAC) interoperability categories (organizational, informational, and technical).

2.2.3. IEEE 2030-2011 Smart Grid Power System, communication technology, and information technology reference

The IEEE "Guide for Smart Grid Interoperability of Energy Technology and Information Technology Operation with the Electric Power System (EPS), End-use Applications, and Loads", as shown in Figure 2.4, again models the Smart Grid, the various actors and influencers, and the interconnection between systems.

IEEE 2030 Smart Grid Interoperability Reference Model (SGIRM) differs from the other standards in the way that its intent is not the security itself but rather is to provide to all different stakeholders common interoperability criteria for the different perspectives (power system, communications, and IT).

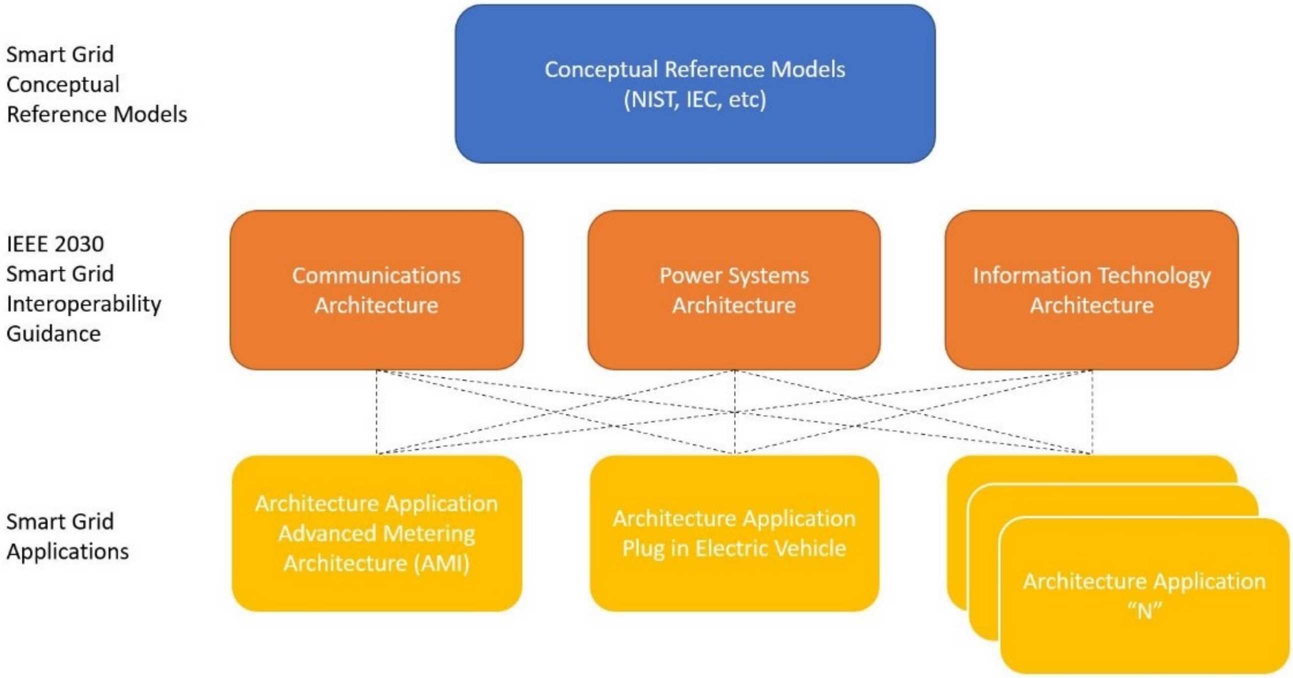

Smart Grid Conceptual Reference Models
Conceptual Reference Models (NIST, IEC, etc)
IEEE 2030 Smart Grid Interoperability Guidance
Communications Architecture
Power Systems Architecture
Information Technology Architecture
Smart Grid Applications
Architecture Application Advanced Metering Architecture (AMI)
Architecture Application Plug in Electric Vehicle
Architecture Application "N"

Figure 2.4. IEEE 2030-2011 Smart Grid Interoperability Reference Model (SGIRM)

2.2.4. ISA-62443, zones and conduits and Smart Grids

The extensive use of industrial automation in Smart Grids makes the ISA SP99 security recommendations highly relevant. The standard is shown in Figure 2.5 and it focuses on the separation of the protected systems based on the physical and logical location. The standard uses the "zone and conduit" model (Figure 2.6) in order to identify the "zones" i.e. the systems that work together by necessity or function and the "conduits" i.e. the separations between the zones. This standard is generally used complimentary to other standards such as IEC 61850 to add more functions.

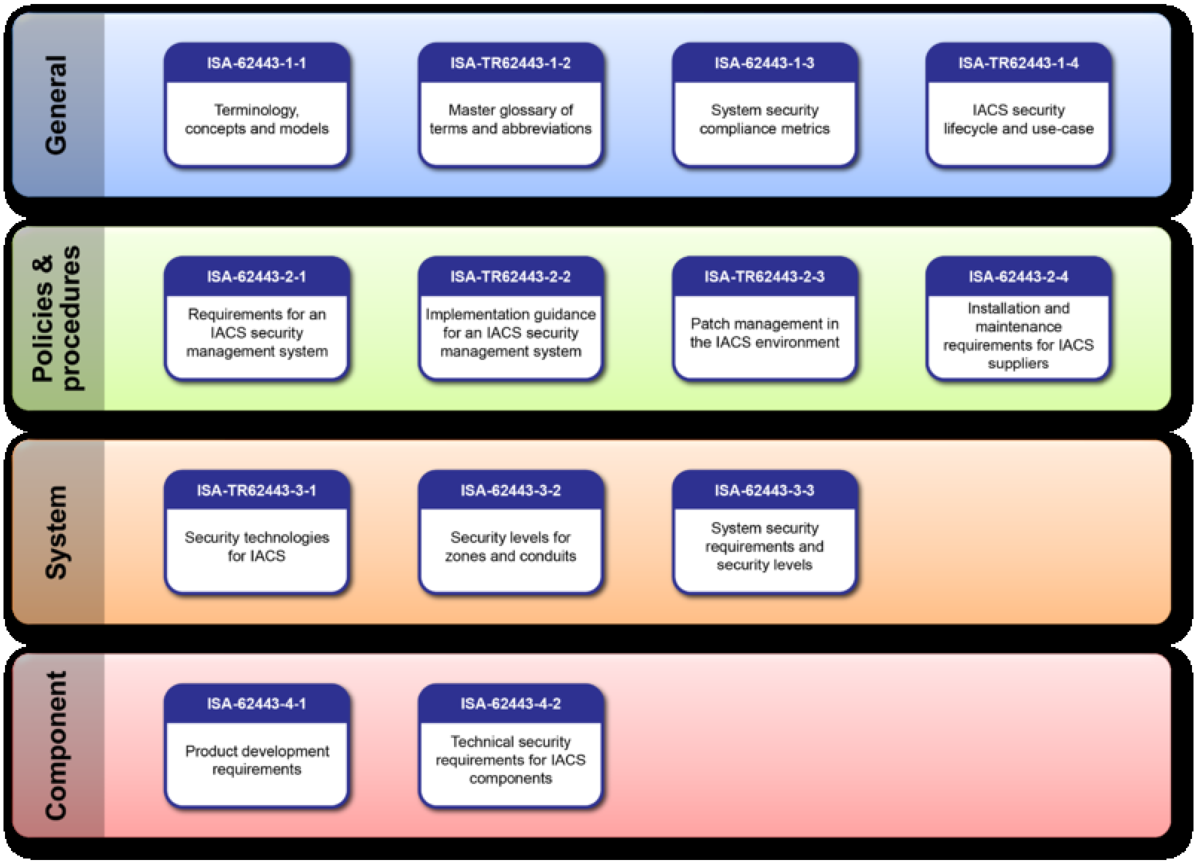

General
ISA-62443-1-1
Terminology, concepts and models
ISA-TR62443-1-2
Master glossary of terms and abbreviations
ISA-62443-1-3
System security compliance metrics
ISA-TR62443-1-4
IACS security lifecycle and use-case
Policies & procedures
ISA-62443-2-1
Requirements for an IACS security management system
ISA-TR62443-2-2
Implementation guidance for an IACS security management system
ISA-TR62443-2-3
Patch management in the IACS environment
ISA-62443-2-4
Installation and maintenance requirements for IACS suppliers
System
ISA-TR62443-3-1
Security technologies for IACS
ISA-62443-3-2
Security levels for zones and conduits
ISA-62443-3-3
System security requirements and security levels
Component
ISA-62443-4-1
Product development requirements
ISA-62443-4-2
Technical security requirements for IACS components

Figure 2.5. ISA/IEC 62443 standard

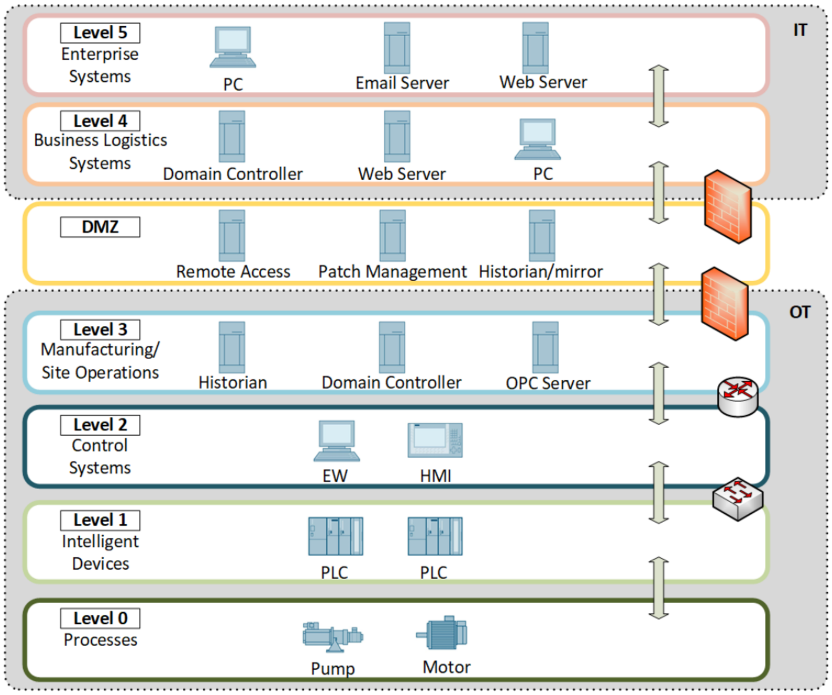
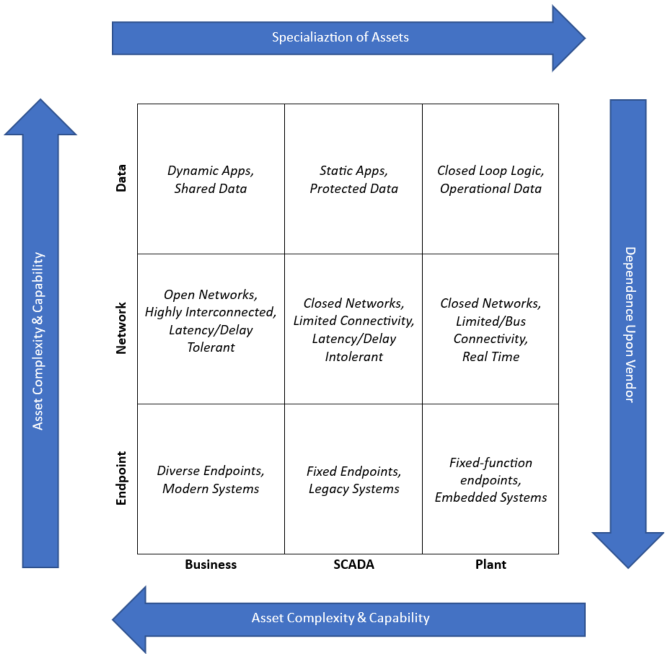


Figure 2.6. ISA-62443 model, zones and conduits

2.2.5. The McAfee Security model for critical infrastructure cyber security

McAfee model – shown in Figure 2.7 – specifies the level of security to be adopted based on the specialization of assets and their complexity. For instance, in case of SCADA, different measures will be adopted based on whether the asset is an endpoint, network, or data. Etc.



	Business	SCADA	Plant
Data	Dynamic Apps, Shared Data	Static Apps, Protected Data	Closed Loop Logic, Operational Data
Network	Open Networks, Highly Interconnected, Latency/Delay Tolerant	Closed Networks, Limited Connectivity, Latency/Delay Intolerant	Closed Networks, Limited/Bus Connectivity, Real Time
Endpoint	Diverse Endpoints, Modern Systems	Fixed Endpoints, Legacy Systems	Fixed-function endpoints, Embedded Systems

Figure 2.7. McAfee security model for critical infrastructure

For example, when applying McAfee model on a smart grid, only the components required depending on the application unit, are used. The result then becomes similar to the one in Figure 2.8.

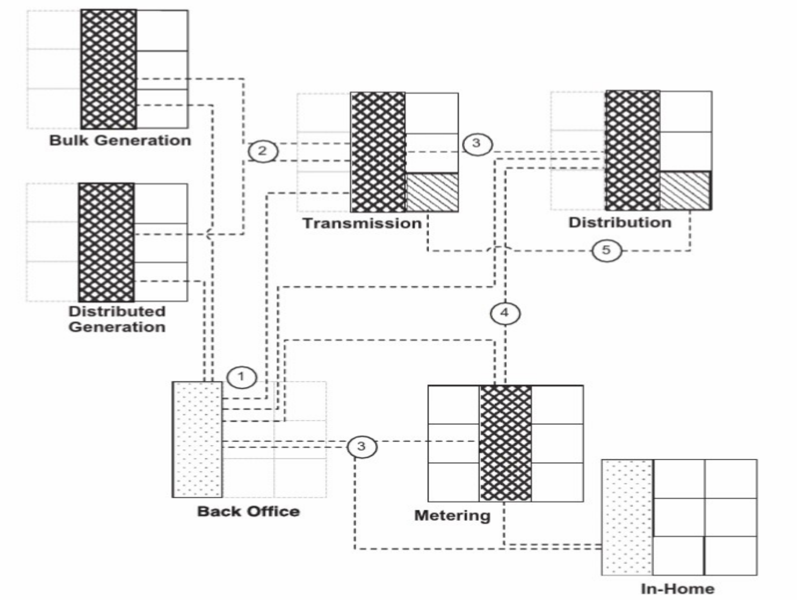


Figure 2.8. Applying McAfee model on smart grid's different units

A simplified smart grid reference model for cybersecurity includes different field zones, different control zones and an energy service zone that is separate from the back-office services zone. All previous zones are connected to the enterprise zones via conduits, and they can follow different security measures as needed. This is shown in Figure 2.9 below.

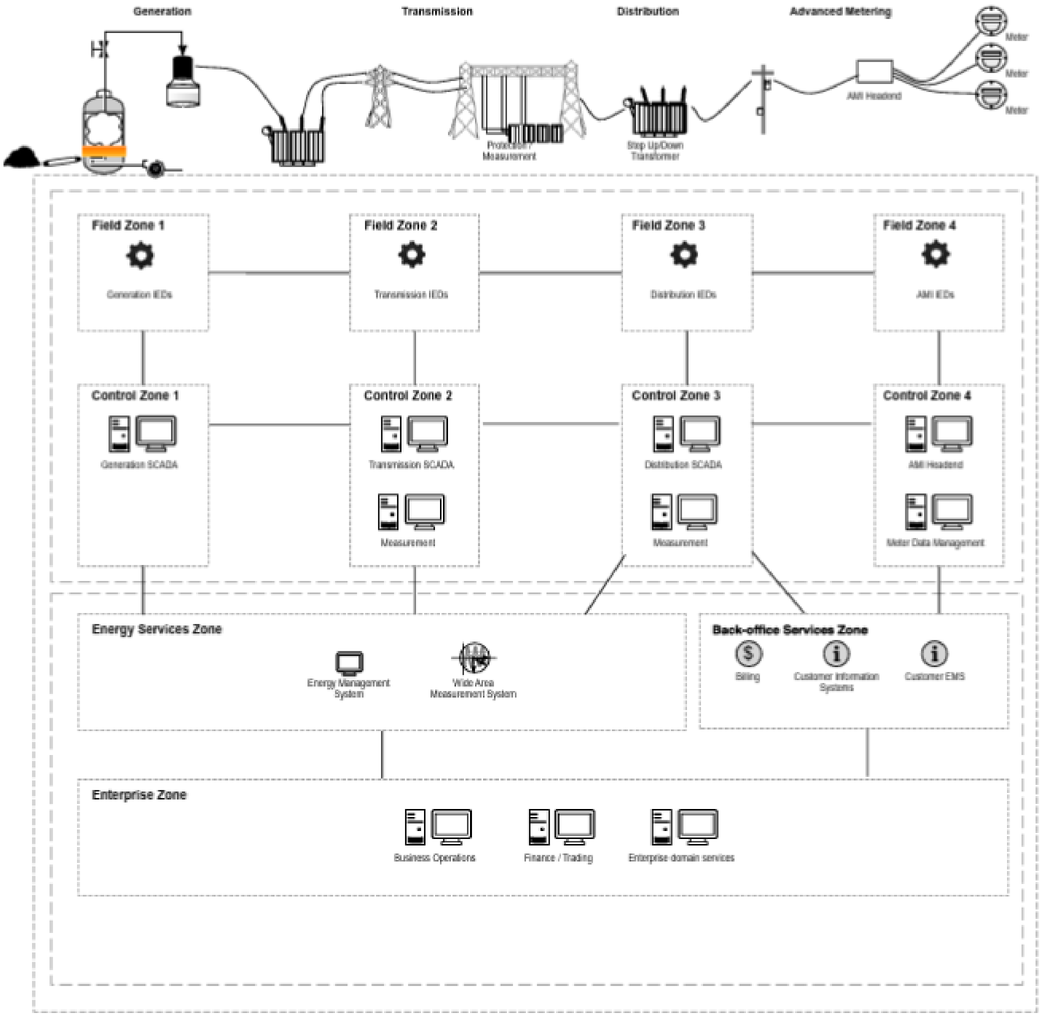


Figure 2.9. A simplified smart grid reference model with different zones and conduits

2.3. Securing the Smart Grid

The different products and technologies can be used together, perhaps by covering different areas of the Smart Grid in order to minimize the risk of a successful cyber-attack. In this way, by having multiple layers of protection, the system can remain resilient even if some of the defenses fail creating a fault tolerant security environment.

2.3.1. Implementing security control within Smart Grid endpoints

- Access control/data access control.

- Anti-virus.
- Application whitelisting or dynamic whitelisting.
- Change control or configuration control.
- Database security.
- Endpoint encryption.
- Host Data Loss Prevention (DLP).
- Host firewall.
- Host Intrusion Detection Systems/Host Intrusion Prevention Systems (HIDS/ HIPS).
- System hardening.

2.3.2. Field zone protection

Field devices (such as IEDs) are usually designed to be low cost and low power and as a result little resources are available to implement security countermeasures (memory, Central Processing Unit (CPU) etc). Although these devices can pose a serious threat to the system, operators frequently do not pay enough attention into securing them and the simplest way of security is most often used.

2.3.3. Control zone protection

- Application whitelisting
- Anti-virus
- Change control, change management or configuration management systems
- HIDS and HIPS
- DLP
- Event logging

2.3.4. Service zone protection and back-office systems

Abovementioned control zone protection mechanisms apply

2.3.5. Establishing strong boundaries and zone separation

Only one physical connection - the ISA-99 “conduit” - is expected to exist between two separate systems that contains the necessary communication paths (“wires”). When designed this way, it is easier to demarcate both ends of the conduit with the needed security gateways (Virtual Private Network (VPN), firewall, Intrusion Detection System (IDS), Intrusion Prevention System (IPS), etc.). In Smart Grids the amount of the wires and conduits can be very difficult to manage so any excess connectivity should be avoided. Encryption and authentication are highly needed, although care is needed in order to preserve the functionality of the monitoring and situational awareness tools that might be affected by encryption. Transport Layer Security (TLS) can add overhead and lead to issues with real-time communication. “Risk vs Visibility” method is proposed to adequately cover the need for strong network security where all traffic and endpoints are visible and depending on the criticality of the connection, strong network VPN or security gateway solutions can be adopted.

2.3.6. Compensating controls

Compensating controls of network security include firewalls, IDSs, IPSs, network access control systems etc. Firewalls should be used to minimize unauthorized network connections. Of course, some unauthorized connections will remain, so in order to prevent and mitigate their effects several other supplementary technologies can be used such as the following:

- Industrial protocol filters monitor industrial protocols such as Modbus, DNP3, 61850, and others, and filter traffic based upon the protocol being used
- Intrusion detection systems to perform deep packet inspection (DPI) on network traffic
- Intrusion prevention systems to be deployed inline and can actively block traffic by dropping the offending packet, or by resetting the Transmission Control Protocol (TCP) session.
- Application content inspection systems to perform a hybrid function
- TLS

2.3.7. Advanced network monitoring

More advanced network monitoring is realized for example by use of Network Behavior and Anomaly Detection (NBAD) tools which are devices whose main function is to analyze, detect and block suspicious traffic similarly to an IPS. Another approach is to use a Security and Event Management System (SIEM) that can also detect anomalies. Complementary to the aspect of network monitoring, it is also very useful to use a network forensics tool (e.g. Netwitness Investigator or Solera's Networks' DeepSee products) whose function is the storage and preservation of the network traffic in order for it to be used as forensic evidence in the event of an attack. Lastly, for the sensitive information both at rest and in motion, DLP techniques are applicable. DLP prevents the loss or theft of data across the network by detecting specific data that has been tagged or flagged in some way.

2.3.8. Protecting data and applications within the smart grid

Data protection in Smart grids requires the operators to have a very broad and yet detailed knowledge of the system.

The required information is the following:

- Being aware of all the data and applications that are being used within the Smart Grid, including:
  - Where automation logic resides, what it controls, and how.
  - Where measurements are being taken, and how those measurements are being used.
  - Where management systems— including SCADA, EMS, and other - systems—reside, what they manage, and how.
  - What business applications are being used, how they utilize or depend upon grid operations or measurement data, and how they obtain that data.
- Being aware of where repositories of data reside, and how they are stored (i.e. a database).
- Being able to collect that information in a format that is relevant to digital cyber security, even if the data spans multiple domains or zones.

- Being able to analyze and assess that data in a meaningful manner, to detect indications of cyber risk and threat.
- Being able to articulate that analysis back to the many stakeholders involved in Smart Grid operations.

Data and application protection systems include SIEM, Network DLP (both mentioned in section 2.3.7) and also Database Activity Monitoring (DAM), which is a system that monitors activity to and from databases, either via in-line inspection of network traffic, or via a host-based agent that monitors database activity locally.

2.3.9. Situational awareness

“Situational Awareness” refers to a process of perception, decision and action that enables the assessment of and reaction to a situation. Perception is formed when the appropriate information is aggregated usually by the SIEM and then informed decisions can be taken. SIEM consoles provide a way for the operator to make a manual assessment of the situation in an environment where the available information is also automatically processed to some extent. SIEM consoles can correlate the current situation with known threat patterns and indicate rather complex threats and also keep track of the statistical deviation between the current state of the system and the historical baseline of the system. Another function is to validate information that is being used by the system with external sources and check that the data conforms with defined criteria. The risk is calculated for the different assets, users and applications and an alert is triggered when the risk is high.

For effective situational awareness, a minimum of the following should be monitored:

- All endpoint activity of the servers, gateways, controllers, field devices, etc.
- All network activity between any and all of these devices, obtained from the network infrastructure itself and/or from network probes.
- All data produced by and/or utilized within the grid, especially readings from measurement devices, device status information, protection status, phasor data, etc.

Also, to complete the protection scheme, monitoring should be applied to all zones and conduits.

3. IEC 61850 and IEC 62351

Goals	To learn about: 1. IEC 61850 Grid communication standard 2. IEC 62351 Security standard for IEC 61850
Pre-reading material	1. Mackiewicz, Ralph E. "Overview of IEC 61850 and Benefits." 2006 IEEE Power Engineering Society General Meeting. IEEE, 2006. 2. Brunner, Christoph. "IEC 61850 for power system communication." 2008 IEEE/PES Transmission and Distribution Conference and Exposition. IEEE, 2008. 3. Cleveland, Frances. "Iec tc57 wg15: Iec 62351 security standards for the power system information infrastructure." White Paper (2012). 4. Hussain, SM Suhail, Taha Selim Ustun, and Akhtar Kalam. "A review of IEC 62351 security mechanisms for IEC 61850 message exchanges." IEEE Transactions on Industrial Informatics 16.9 (2019): 5643-5654.
Main material resources	IEC 61850 Communication Protocol Manual Relion Protection and Control. 650 Series IEC 61850 Communication Protocol Manual, Revision: Product version: 1.1; ABB AB Substation Automation Products SE-721 59: Västerås, Sweden, 2011. Available online https://www.naic.edu/~phil/hardware/sitePower/evd4/1MRK511242-UEN_-_en_Communication_protocol_manual__IEC_61850__650_series__IEC.pdf
Hours assigned	
Assignment criteria	1. Pre-lecture MCQ evaluation set 2. Post-lecture group discussion followed by a short essay
Done by	National Technical University of Athens

3.1. Introduction to IEC 61850

IEC61850 is a standard that was created for the communication of the different components of power substations (e.g. IEDs). Later, it was expanded to be more generic, e.g. IEC61850-7-420 describes the modeling of DERs like photovoltaics and diesel generators. The main feature of the standard is that it can be mapped to different communication protocols (mappings exists for GOOSE, SV and Manufacturing Message Specification (MMS) protocols) giving a unified structure to the messages and enabling the interoperability between the devices with the vision to allow communication between vendor-agnostic devices. It is used for the protection, control, measurements and monitoring functions of the station and enables high-speed substation applications, station wide interlocking and other functions. A latency of 4 ms is designated, by this protocol, satisfying the strict timing requirements of the power grid. In Figure 3.1, communication between the different IEDs is shown.

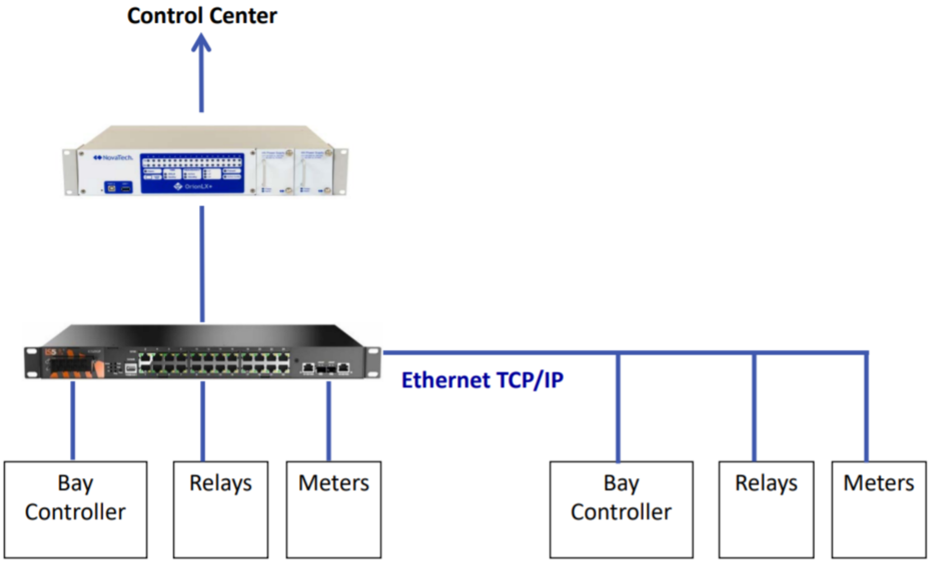


Figure 3.1. Communication between IEDs

- IEC 61850 provides the means for interoperability between the different IEDs by standardizing:
  - Designs
  - Descriptive language
  - Services
  - Configuration
  - Substation information
  - Devices services
  - Naming convention
  - Faults records
  - Logs
  - Tests

- IEC 61850 features include:
  - Data modelling

  - Reporting schemes
  - Fast transfer of events
  - Reduction of hard-wired connections
  - Setting groups
  - Sampled data transfer
  - Supporting various commands
  - Substation Configuration Language (SCL) standardization
  - Data storage

IEC61850 has been designed in order to minimize the design efforts and required maintenance and to simplify the installation tasks and the expansion of the system by making it easy to add new equipment with minimum configuration required. The communication in the network layer is implemented through the Generic Object-Oriented Substation Events (GOOSE and SV protocols).

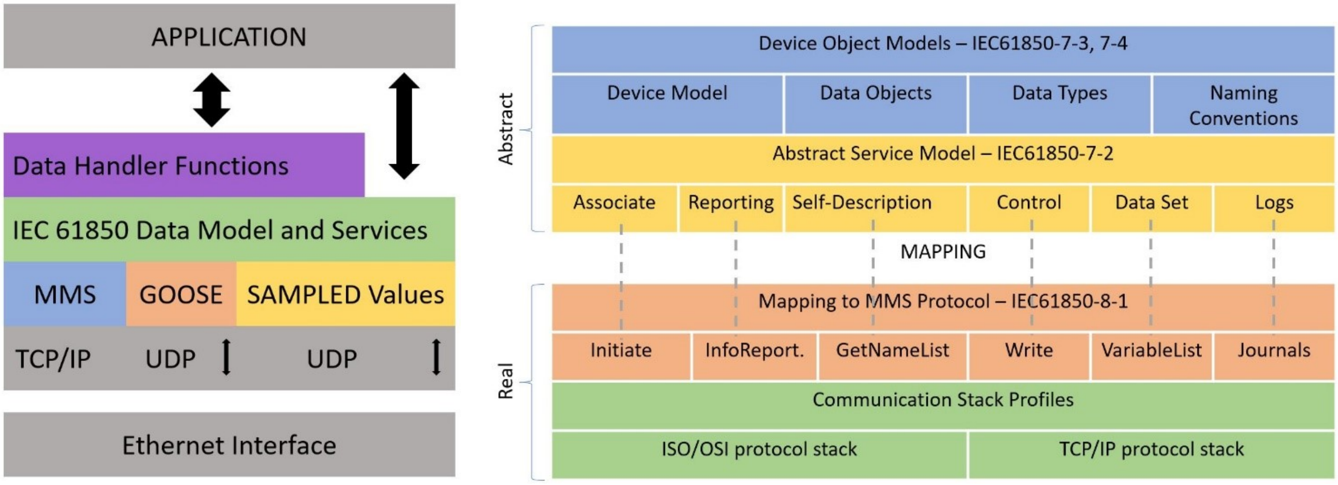


Figure 3.2. IEC 61850 standard stack

3.2. Substation Configuration Language

IEC 61850 defines the SCL which is an Extensible Markup Language (XML)-based configuration language used to support the exchange of database configuration data of substation devices between different tools, which may come from different manufacturers. SCL files include information regarding representation of modeled data and communication services, as well as representations of substation device entities based on logical devices and logical nodes.

- Types of SCL files:
  - SSD: System Specification Description, that describes the entire system based on the one-line diagram of the substation and the required Logical Nodes (LNs)
  - SCD: Substation Configuration Description, that describes a single substation with all IEDs included and their communication configuration data.
  - ICD: IED Capability Description, that describes complete capabilities supported by an IED (e.g. GOOSE support)

- CID: Configured IED Description, that describes configuration for a specific IED

- SCL file contains five sections:
  - Header
  - Substation section: organization of the IEDs within the substation
  - Communication section: establishment of the communication between the IEDs
  - IED section: description of the logical node types included in the respective IED configuration, the data sets and the control blocks. The data sets and the control blocks are logically defined as part of the logical nodes (see IEC 61850–7–2 clause 9)
  - Data type template section: content description of each logical node type to all tools and users (clients) of the information. Each IED and vendor may have their own logical node type definitions included in the data type template section together with all other logical node types based on the standard.

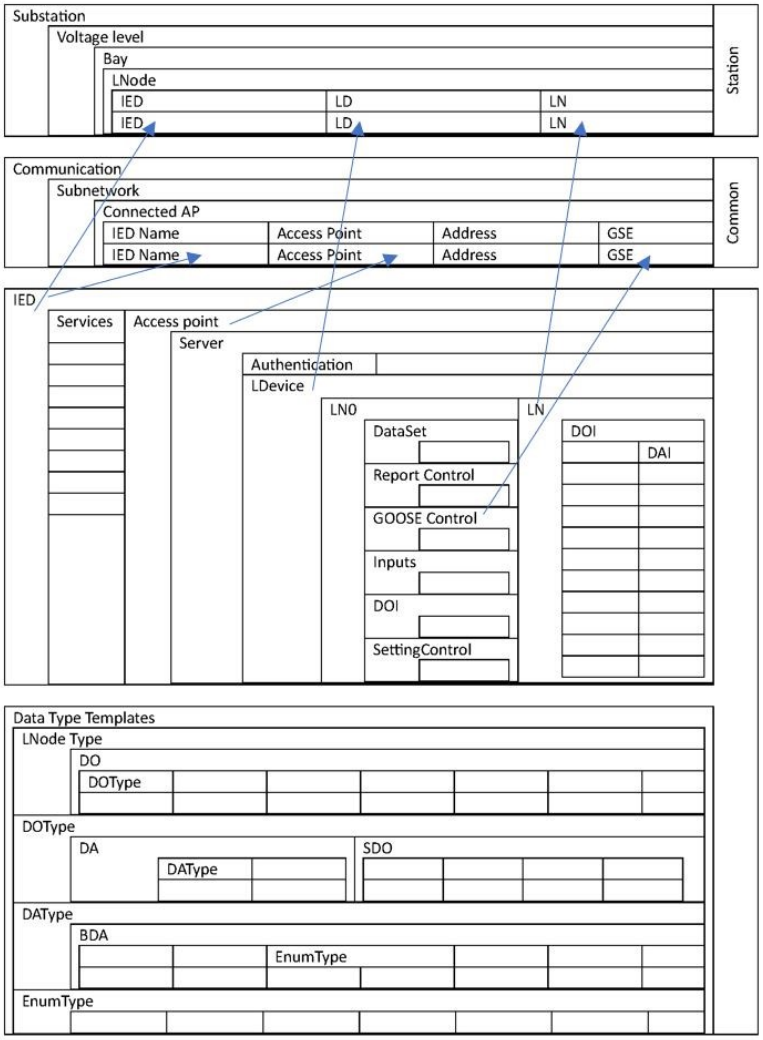


Figure 3.3. Connection between the SCL sections

Figure 3.3 shows the connection between the different SCL sections. The arrows show the link between the different sections given when an IED is integrated in the substation structure and/or in the communication structure. All needed logical nodes of an IED are linked to the substation section by the SC tool.

3.2.1. The Substation Section

As mentioned above, the substation description in IEC 61850–6 clause 9 describes the organization of the primary equipment (IEDs) and the logical nodes used and their relation to the primary equipment.

3.2.2. The Communication Section

The communication section describes how information is routed between the IEDs and contains the following parts:

- Subnetworks
- IEDs connected to different subnetworks
- Access points per IED to subnetworks
- Address
- IP address of LAN network (is exceptionally part of the address elements)
- Link to GoCB message in transmission direction (extended during signal engineering and routing)

3.2.3. The IED Section

The IED section describes the complete IED and may include the data type template part even when it is separated in its own section. The IED's ICD files include the description of the logical nodes, their data type templates and the used or supported services. The structure of the IED section follows the definitions made in the IEC 61850 standard.

Two basic IED types are used in system configuration:

- Station level IEDs: located on the station level and are identified as client IEDs when they read or write information from or to the bay IEDs. These logical nodes have no data objects. They are only used to link the report control blocks (BRCBs) from the server IEDs.
- Bay level IEDs: located on the bay level and are identified as server IEDs when they read or write information vertically. When GOOSE messages are received, the bay level IED also has the client role.

3.2.4. IEC 61850 Information Structure

In Figure 3.4, the information structure of IEC 61850 is shown. In this figure, logical devices and logical nodes are the main concern.

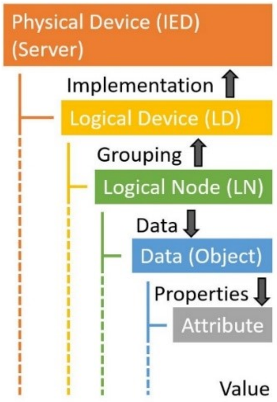


Figure 3.4. IEC 61850 information structure

- Logical Devices: Each device in a substation is a logical device. Each logical device should have a unique name
- Logical Nodes: Each logical device can support different functions, which are specified and categorized as logical nodes.

In Table 3.1, different logical groups and their names are shown.

Table 3.1. IEC 61850 logical groups

Logical Group	Name
L	System LN
P	Protection
R	Protection related
C	Control
G	Generic
I	Interfacing and archiving
A	Automatic control
M	Metering
S	Sensor and monitoring
X	Switchgear
T	Instrument transformers
Y	Power transformers
Z	Further power system equipment

3.3. Communication profiles

IEC 61850 is created by design to be independent from existing communication media and message transmission concept. The abstract model is then mapped to specific protocol stack based on Ethernet as the medium, TCP/IP and MMS. Part 7-2 of the standard defines the ACSI. The mapping of ACSI to the MMS for all aspects of services and Ethernet usage is specified in part 8-1 of IEC 61850. Device manufacturers that want to support IEC61850 have to make sure that their respective product conforms to the requirements and definitions given in the standard. In order for other partners to be able to check

what they can expect and what they have to support, a device's profile is defined in the Protocol Implementation Conformance Statement (PICS) document. The PICS document contains in a table-based form the conformance of a product or product family to ACSI.
The specific protocols for which a mapping from IEC61850 exists include:

- GOOSE
- SV
- TimeSync using the Simple Network Time Protocol (SNTP)
- MMS protocol suite with the T-profile TCP/IP

MMS is an international standard dealing with messaging systems for transferring real-time process data and supervisory control information, similar to standard DNP3.
In Figure 3.5, the relationship between IEC 61850 and other protocols is shown.

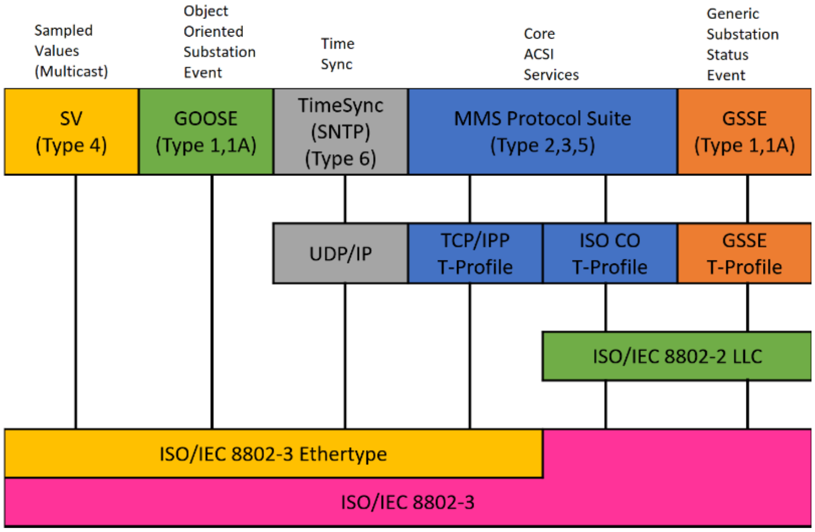


Figure 3.5. IEC 61850 and other communication protocols

3.4. Data sets and control blocks

IEC 61850 has defined data sets and report control blocks for signal transmission in monitoring direction. Data sets are also used for GOOSE messages in horizontal direction.

3.4.1. Data set

A Dataset is simply a group of Data Objects that are transmitted together. They can be arbitrary types and organization of the data. Datasets are used by the reporting & logging services, GOOSE, MMS and Sampled Values. Figure 3.6 shows the position of Datasets inside a logical node.

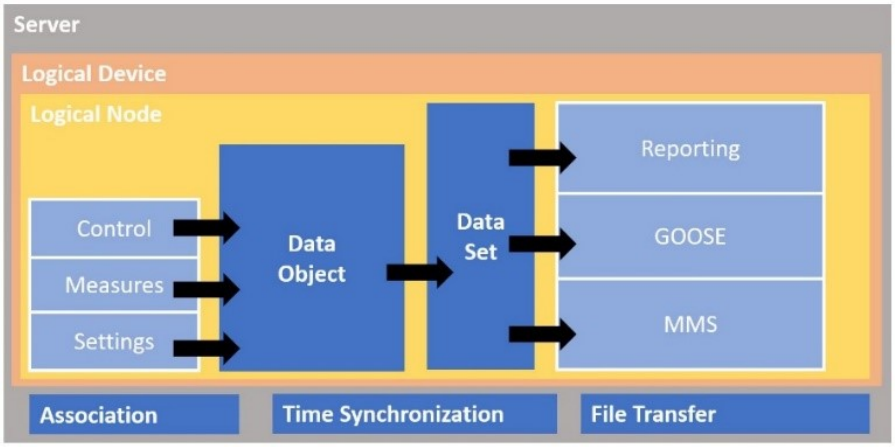


Figure 3.6. IEC 61850 data sets

3.4.2. Report Control Blocks RCB

IEC61850 enables the server to send data (reports) based on preconfigured events without the explicit request of the client using the Report Control Blocks. They are categorized in Buffered (BRCB) and unbuffered (URCB) with the distinction being that buffered reports will be stored for a limited amount of time in case the client is not connected at the time the report is generated. Triggers include the change of value or quality of a variable in the monitored data set.

3.4.3. GOOSE Control Blocks GoCB

GOOSE protocol is a communication model (defined in IEC61850) that enables very fast and reliable transfer of data between IEDs at bay level (horizontal direction) over Ethernet in a multicast way (one sender multiple receivers). GOOSE messages can request a return message in which way a loop is closed between the sender and the receiver that ensures that both operate safely. This implementation uses a specific scheme of re-transmission to achieve the appropriate level of reliability and it is suitable for transferring of status, controls and measured values between peers (P2P messaging). When a GOOSE server generates a SendGOOSEMessage request, the current data set values are encoded in a GOOSE message and transmitted on the multicast association. The event that causes the server to invoke a SendGOOSE service is a local application issue as defined in IEC 61850-7-2. Each update may generate a message in order to minimize throughput time. GOOSE control block specifies the properties and behavior of the GOOSE message and is an integral part of the message.

3.5. Overview of IEC 62351

In Smart grids the priority is given to the integrity of the measures and due to its reliability and speed, IEC 61850 is the used standard. However, the standard does not cover the cyber security aspect of the substation communication. These issues are addressed by its complementary standard, IEC 62351, which acts as an enabler. IEC62351 also covers, IEC60870-5 (DNP3), IEC60870-6, IEC 61970 and IEC 61968 (CIM) protocols.

3.5.1. IEC 62351

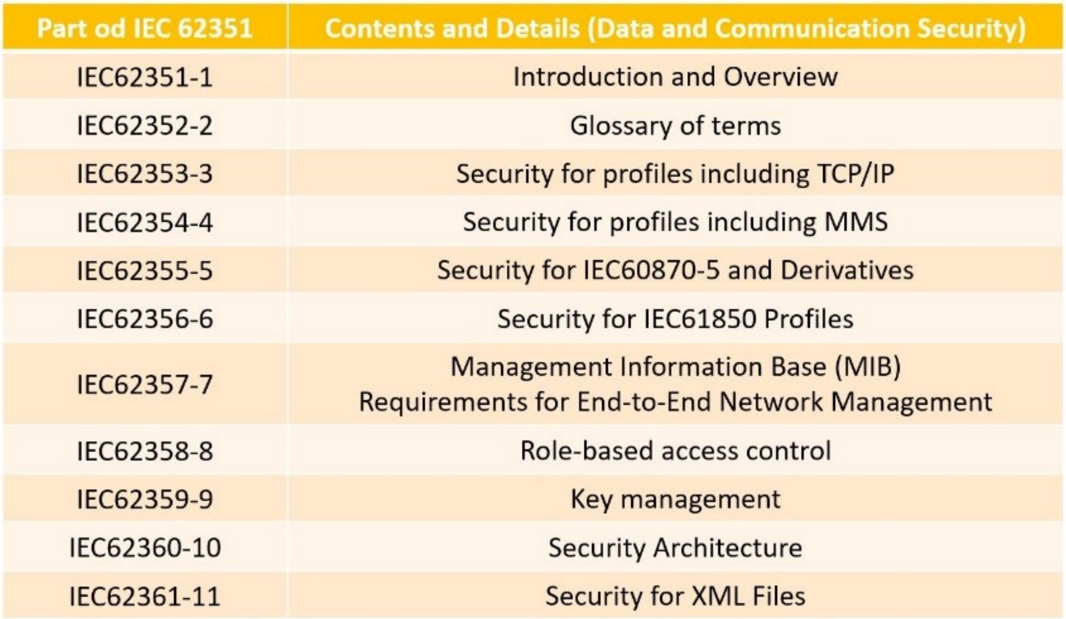
IEC62351 is used in environments where layered security is in place and specifies security requirements both at the transport and application layer providing more protection than VPNs (only transport level secured). It also extends the integrity and authentication both at the handshake phase and the data transfer phase and can be used in systems that use other protocol stacks other than the OSI. The protocol includes a section dedicated to key management, it provides data transfer encryption at the application layer and it provides security end-to-end (E2E) with zero or more intermediate entities.

3.5.2. Security features and integration with IEC 61850

- IEC 62351 Security parameters:
    - Authentication: The property that the claimed identity of an entity is correct.
    - Authorization: The process of giving someone permission to do or have something.
    - Integrity: The property that information has not been altered in an unauthorized manner.
    - Non-repudiation: The property that involvement in an action cannot be denied.
    - Confidentiality: The property that information is not made available or disclosed to unauthorized individuals, entities or processes.

3.5.3. Parts and content of IEC 62351

Figure 3.7 shows the content of the IEC 62351 parts. Also, Figure 3.8 shows the mapping between IEC 62351 and the other IEC TC57 standards (Power Systems Management and Associated Information Exchange)



Part od IEC 62351	Contents and Details (Data and Communication Security)
IEC62351-1	Introduction and Overview
IEC62352-2	Glossary of terms
IEC62353-3	Security for profiles including TCP/IP
IEC62354-4	Security for profiles including MMS
IEC62355-5	Security for IEC60870-5 and Derivatives
IEC62356-6	Security for IEC61850 Profiles
IEC62357-7	Management Information Base (MIB) Requirements for End-to-End Network Management
IEC62358-8	Role-based access control
IEC62359-9	Key management
IEC62360-10	Security Architecture
IEC62361-11	Security for XML Files

Figure 3.7. Contents of IEC 62351 parts

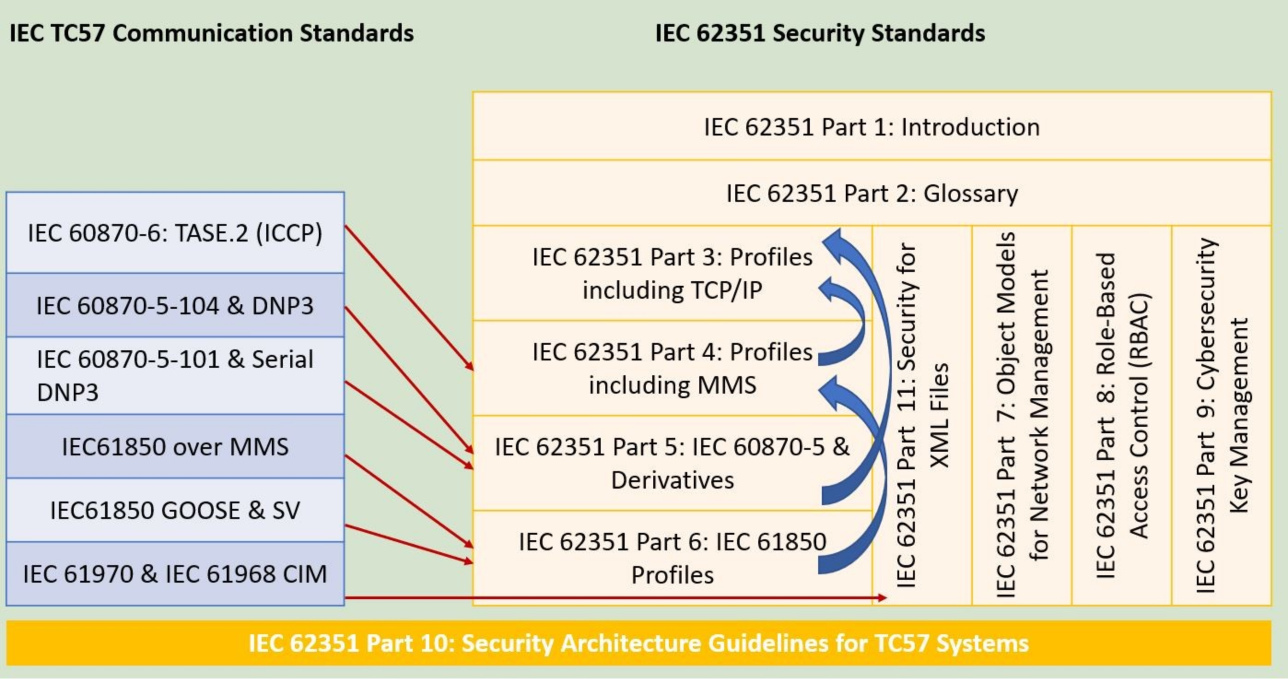

IEC TC57 Communication Standards
IEC 62351 Security Standards
IEC 62351 Part 1: Introduction
IEC 62351 Part 2: Glossary
IEC 60870-6: TASE.2 (ICCP)
IEC 60870-5-104 & DNP3
IEC 60870-5-101 & Serial DNP3
IEC61850 over MMS
IEC61850 GOOSE & SV
IEC 61970 & IEC 61968 CIM
IEC 62351 Part 3: Profiles including TCP/IP
IEC 62351 Part 4: Profiles including MMS
IEC 62351 Part 5: IEC 60870-5 & Derivatives
IEC 62351 Part 6: IEC 61850 Profiles
IEC 62351 Part 11: Security for XML Files
IEC 62351 Part 7: Object Models for Network Management
IEC 62351 Part 8: Role-Based Access Control (RBAC)
IEC 62351 Part 9: Cybersecurity Key Management
IEC 62351 Part 10: Security Architecture Guidelines for TC57 Systems

Figure 3.7. Mapping between IEC 62351 and other IEC TC57 protocols

PART TWO:

Real-Time Simulation, The Theory

Fundamentals of Real-Time Simulation systems.

Goals	To learn about: 1. Real time simulation systems
Pre-reading material	1. Mikkili, Suresh, Anup Kumar Panda, and Jayanthi Prattipati. "Review of real-time simulator and the steps involved for implementation of a model from MATLAB/SIMULINK to real-time." Journal of The Institution of Engineers (India): Series B 96.2 (2015): 179-196. 2. Chen, Bo, et al. "Implementing attacks for modbus/TCP protocol in a real-time cyber physical system test bed." 2015 IEEE International Workshop Technical Committee on Communications Quality and Reliability (CQR). IEEE, 2015. 3. Sun, Chih-Che, Adam Hahn, and Chen-Ching Liu. "Cyber security of a power grid: State-of-the-art." International Journal of Electrical Power & Energy Systems 99 (2018): 45-56.
Main material resources	Popovici, Katalin, and Pieter J. Mosterman, eds. Real-time simulation technologies: Principles, methodologies, and applications. CRC Press, 2017.
Hours assigned	
Assignment criteria	1. Pre-lecture MCQ evaluation set 2. Post-lecture group discussion followed by a short essay
Done by	Chapter 4: Riga Technical University Chapters 5, 6 and 7: University of Vaasa

4. Simulation Fundamentals

4.1. Real-Time Simulation Using Hybrid Models

4.1.1. Discrete and continuous models

Developing a simulation of a natural system involves developing a conceptual model of the simulated system, which may be based on a set of rules, mathematical equations or another method to determine the state of the simulated system and the way it changes over time.

Historically, two types of models were distinguished for both real-time and non-real-time simulations, discrete models and continuous models. Therefore, two types of simulation could be derived: discrete simulation and continuous simulation. As a result, the terms discrete system and continuous system were used to characterize the actual systems that were being simulated, even though it is often the model to represent it.

- Traffic flow as an example:
    - Traffic studies often use discrete models in which every vehicle is represented, and events such as arriving at an intersection or at a line of stationary traffic on changing a traffic signal cause changes in the state of the system.
    - Traffic flow studies on the highway, where traffic volumes can be much higher, however, are often represented using a continuous model in which traffic is treated as a fluid flowing along the highway, with traffic flow speeds as model variables.

Discrete and continuous modelling are two different approaches to modelling dynamic systems. In both types of models, the system is characterized by a state of the system that changes over time. It is the nature of these changes that distinguishes the two approaches. A hybrid model is one that includes elements of both discrete and continuous models. The discrete model assumes that the state of the system changes at certain discrete points in time and remains unchanged between these points. These changes in system state occur instantaneously as a result of the event that causes them. In the discrete model, time moves from event to event with a corresponding update of the system state at each time point of the event.

In a continuous model, the state of the system is assumed to change in a continuous fashion as defined by the model differential equations, which relate the instantaneous rates of change of system variables to the current state of the system. In practice, the advance of time in the simulation is quasi-continuous. In effect, even continuous simulations execute discretely.

4.1.2. Discrete modelling

A discrete model is one in which the state of the system is assumed to change at specific instances. Very often, the discrete model is based on queuing models. In a typical queuing model, entities such as customers or parts arrive at service points representing operators or service units, which process them in turn. The waiting entities form queues, and both the arrival times of new entities and the service times of the servers are often generated from statistical distributions using random number generators. Changes in the state of the system caused by queue arrivals or departures, or completion of service, are referred to as events, and the time at which the event occurs is the event time.

A simulation based on a discrete model defines the initial state of the system and a queue of future events with time intervals of events. The simulation then advances to the first of these even, ts and the system state is modified accordingly. These changes can result in changes to the entries in the event queue, including the identification of additional events, and the event queue is modified accordingly. Once the current state is fully established, the simulation moves on to the next event and the process is repeated. This repetitive sequence continues until the simulation fulfils some final condition.

Three main approaches have been developed for simulations that use these models, known as the activity-based, event-based, and process-based methods:

- Activity-based simulation is used to describe simulations in which time advances in small steps with checks for changes at each step. The term is also being used for Discrete Event System Specification (DEVS) simulations, where only components that are potentially active are evaluated. The original kind of activity-based simulation is inefficient and suitable only for simple applications.
- Event-based simulation, in which time advances from event to event in a single software read, has been the basis of many popular discrete simulation languages.
- As parallel computing options increase, process-based simulation using parallel processors and multiple software threads has become the more popular approach. In this approach, the simulation is divided into processes that can be run in parallel.

There are two distinct types of discrete models, the queuing models and models that are used to represent systems that can be described by means of difference equations or z-transforms. These digital system models include digital electronic circuits and systems and sampled-data and digital control systems in which the discrete subsystem is updated at typically equally spaced time intervals. This type of discrete model consists of a set of difference equations that define the next stage of the system in terms of its current and past states and its external inputs. If the model is defined, in its entirety or in part, by means of z-transforms, it is usually a straightforward task to convert the z-transform model into a different equation form.

In a discrete simulation, time management is based on knowledge of the times of future events, which are either known a priori or are maintained in a prioritized event queue with priorities that are determined by the time ordering of the events in the queue. It is not necessary to know the time of all events at the start of a simulation, but in many cases, as long as the time of the next event is known, time management becomes fairly straightforward. The simulation will begin with a specification of the initial state of the system, and some future events can be determined using the random number generators to generate event times for the various elements in the system. Time is then advanced to the next event time, and appropriate changes are made to the state of the system. A queue length may be increased or decreased upon arrival in the queue or by the completion of service. Simulation continues alternately between updating the system state as a result of an event and advancing the time to the time of the next event. The time of the final event is determined by the completion condition set by the programmer.

There are many software products aimed at supporting discrete-event simulation, for example, General Purpose Simulation Systems, Simscript, SIMAN/Arena, Simula, and Dymola/Modelica. Simulation of digital electronics, digital control, and sampled-data systems can be accomplished using digital hardware design software such as VHDL or Verilog, which combines a specification of the system with a simulator to test the design.

4.1.3. Continuous modelling

A continuous model assumes that the state of the system changes continuously, without instantaneous changes in the values of the system states or their derivatives. Continuous models are usually described using a combination of differential and algebraic equations. This may be expressed as first-order differential equations, each defining a state variable of the system and additional equations defining auxiliary variables. The independent variable is time, thus forming a mathematical model of a continuously evolving dynamic system. Some models may contain additional independent variables, such as spatial dimensions.

A simple example is the simulation of temperature changes along a thin metal rod as it is heated. The state variable is temperature, and it varies with both time and displacement along the rod. The mathematical model is a partial differential equation with two independent variables, time and displacement. Simulation using a continuous model involves the initialization of the values of system states followed by the calculation of the initial values of the other algebraic variables. Variable-step routines typically include an estimation of the error generated in the current step and a strategy to change the step size to satisfy a user-supplied error tolerance. Such variable-step routines are not normally suitable for real-time simulation because their computation times vary from step to step.

Continuous simulations are based on solutions of ordinary or partial differential equations for which time is a continuously changing variable with no instantaneous changes from one value of time to the next, in contrast to the behavior of a discrete simulation in which time jumps from one event time to the next. Computers are programmed to produce approximate solutions of the mathematical model using a numerical approximation in which time does advance in small discrete steps. The numerical algorithms depend on approximations, the use of Taylor series, and Euler integration methods. For instance:

$$y' = \frac{dy}{dt} = f(y, x, u)$$
$$x = g(x, y, y', u)$$

, where *y* is a vector of states,
y' is a vector of state derivatives,
x is a vector of auxiliary variables (algebraic variables),
u is a vector of inputs, and
f and g are arbitrary but well-behaved functions.

To calculate the next step in terms of the current state and past states, time is assumed to advance,

$$\mathrm{y(t+h)\ =\ f(y(t), x(t). u(t))},$$

which can be expressed in terms of the Taylor series:

$$y(t+h) = y(t) + h\,y'(t) + (\frac{h^2}{2!})y''(t) + (\frac{h^3}{3!})y'''(t) + \cdots$$

The simple Euler integration method to solve this is

$$\mathrm{y(t+h)} = \mathrm{y(t)} + h\,y'(t)$$

Figure 4.1 shows this solution using Euler integration and S transform.

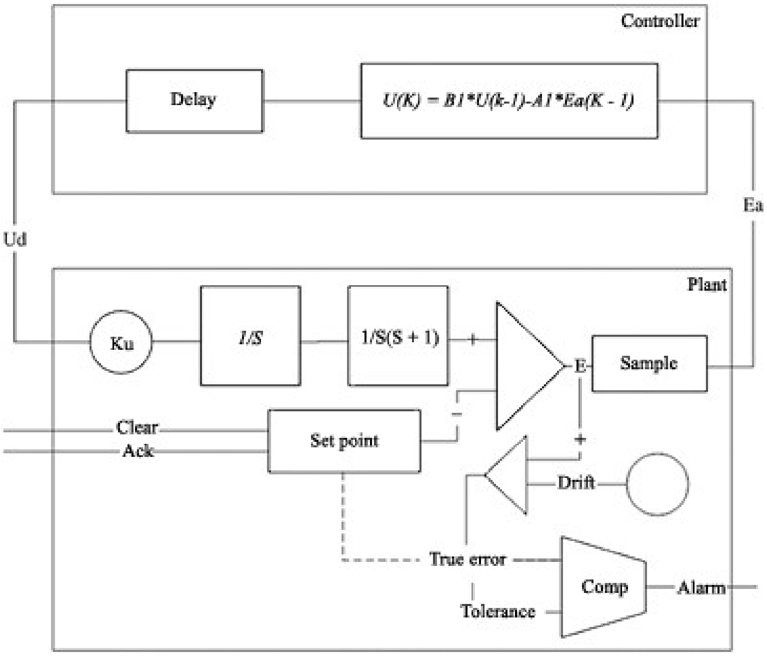


Figure 4.1. Numerical algorithm of continuous modelling using Euler integration and S transform

This process can be repeated until the final time is reached. The previous solution is the simplest form, and it only considers the first two terms in the Taylor series and is known as a first-order method. More complex methods are available that match the Taylor series to the terms in y" (second-order methods), y"' (third-order methods), and so on.

Order and step size are the two major factors that affect the accuracy of numerical integration. Generally speaking, errors decrease with decreasing step size and increasing order. Modern variable-step routines often use an approach in which estimated solutions of order n and n − 1 are generated at each step, and the error estimate is based on the difference between the two. Some methods spread the calculation over more than one step and are called multistep methods. For example, the value of the system state at the next step may be determined in terms of the system state at the current and previous step times.

Other algorithms may require data from the past three or four steps. These methods are often cost-effective (in terms of the amount of computation necessary to achieve a given accuracy), but they do require special procedures to start them since past data is not available at the start of the simulation.

These methods can also cause problems with hybrid and real-time simulations. Typical software used for continuous simulation are MATLAB and Simulink

4.1.4. Hybrid models

A hybrid model is one that combines the characteristics of discrete and continuous models. A hybrid system can be viewed in some cases as a discrete model containing one or more states that change in a continuous fashion between events and in others as a continuous model in which the states change continuously for most of the time but in which one or more of the states or their derivatives can occasionally change instantaneously. In other cases, the discrete and continuous processes may be balanced in an integrated model that is equally distributed between discrete and continuous components.

Regardless of the proportion between discrete and continuous elements in a model, time management for hybrid simulations requires synchronization between the processing of events and the continuous advance of time for the continuous elements. In some cases, a period of continuous simulation can be triggered by a discrete event. In this case, a discrete event may trigger a period of continuous simulation. In this case, the time management control will switch from discrete to continuous mode, with the duration of the time steps determined by the integration algorithm. The simulation will remain in continuous mode until either the continuous simulation terminates or an event occurs in the discrete part of the model that impacts the continuous part. This can cause the continuous simulation to terminate, or it can cause a change of input or even a change to the continuous mathematical model in some way. Conversely, changes in the state of the continuous system can generate new discrete events.

Most of the simulation software that is used to simulate discrete or continuous events can also be used to simulate hybrid simulations. This is done by means of combining programming languages into the simulation algorithm and thus moving between continuous and discrete simulations when some conditions are met/triggered.

Logical programming, *If* and *When* statements and loops are mostly used here. An example is as follows:

```
SUBMODEL TFLOP(LOGICAL:flag:=LOGICAL:trigger);
INITIAL flag:=false;
DYNAMIC when trigger then flag:=not flag;
end_when;
END TFLOP;
```

in which the TFLOP model's flag output is initiated as false and continues like this until the input trigger changes and the hen flag changes. It is seen then that triggers a discrete while the model is continuous.

One of the examples that can be simulated using hybrid models is electric current changes due to a capacitor/coil when the circuit switch is triggered or conversely when the value of the current triggers the circuit switch.

4.1.5. Real-time hybrid simulation

Real-time Hybrid Simulation (RTHS) is a simulation technique that combines physical testing and numerical simulation in real-time. It is used to study the dynamic behavior of complex systems that involve both physical and computational components. This can be hardware, software, or a human in the loop. In all cases, it is necessary for the simulation to be synchronous with a real-time clock to ensure the correct timing of the interactions between the simulation and the external agent.

4.1.6. High-Speed Real-Time Hybrid Simulation HSRT

Problems arise in applications involving short intervals between repeated discrete events, such as in real-time simulations of power electronic systems. This is true, for instance, of real-time simulations of modern power-electronic systems. Electric power can be delivered in many different forms. The most obvious distinction is between alternating current and direct current. Real-time simulations that require frame times of less than about 10–20 µs require special processing systems because conventional real-time simulation systems are not capable of reliably maintaining such short frames. Digital Signal Processors (DSPs) and Field-Programmable Gate Arrays (FPGAs) have been used for this purpose. The key is that the processor that executes the critical simulation code is not vulnerable to interruption by the operating system. A dedicated core or cores in a multicore processor could also be employed in this manner.

Most applications that require HSRT simulation involve components with lower bandwidth dynamics that do not require the short frame-time that HSRT methods are designed to deliver. In these cases, it is better to design a multi-rate simulation in which the system is separated into segments that are simulated using different frame rates. Segments that do not depend on HSRT techniques can be implemented on conventional processors.

Multi-rate simulation has often been used to accelerate the execution of large simulations. The idea is that it is not necessary to update the dynamics of the slower parts of a system at the same rates as are required for the faster parts. This is particularly suitable for HSRT applications for which it is possible to execute the high-speed segments on the special high-speed processors and the slower segments on more conventional computer systems. Issues still exist, e.g., step sizes for the different segments must be chosen with special care; otherwise, multi-rate simulations can become inaccurate or unstable.

4.2. Formalized Approach for the Design of Real-Time Distributed Computer Systems

The need for efficient system design is called on to reduce the design cost and design life cycle. Traditional design approaches cannot meet the requirement of large-scale and complex real-time distributed computer systems because traditional nonformalized design approaches are error-prone, costly, and difficult to validate. Formalized system design approaches have been widely used in many areas for more efficient design of real-time distributed computer systems.

Design approaches fall into three categories: dataflow-oriented, communication-oriented, and object-oriented, as follows. A data flow-oriented design approach basically uses structured analysis/ structured design as the fundamental design method, which is based on the functional decomposition of the system. The concurrency-related issues are not directly handled in the design phase, which results in inefficient design for distributed real-time-based computer systems. Communication-oriented design methods are developed to overcome this problem. However, because of the focus on concurrency naturally, the functional aspect of the system becomes a secondary concern, which results in the limitation of the communication-oriented approach to the design of communication-intensive applications with simple data structures and computations within the system. This approach has limitations related to architectural modelling and prototyping. An object-orient design approach can overcome the shortcomings of both dataflow-oriented and communication-oriented approaches. Most of today's real-time distributed computer systems are built upon object-oriented concepts and programming languages since it naturally fits the modern concept of system designs. Object-oriented design can take full advantage of an object-oriented methodology to use a formal design specification language and tools in one unified domain.

Unified Modelling Language (UML), as an object-oriented formal method, has been widely adopted for designing computer systems, including real-time distributed systems. Discrete Event System Specifications (DEVS,) also as an object-oriented formal system specification language, is playing an increasingly important role in aiding the design of complex real-time computer systems, including real-time embedded systems, distributed real-time P2P systems, distributed Virtual Environments (VEs), and many more.

4.2.1. DEVS as a design formalization tool

For DEVS, the key features are:

- System design models can be directly simulated in one integrative environment that includes model specifications, simulators, and experimental frames (under which models are evaluated using appropriate simulators).
- The design model components are reusable through a model repository due to the separation of models from simulators.
- The model and simulator structure are natively hierarchical, which makes it easy to map them onto complex computer systems.

Figures 4.2 and 4.3 show the general concepts of DEVS.

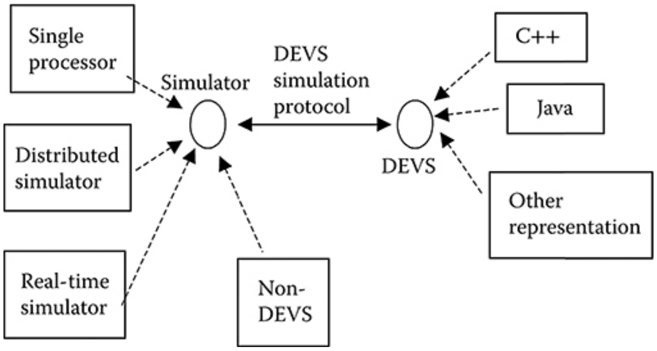


Figure 4.2. DEVS simulation concept

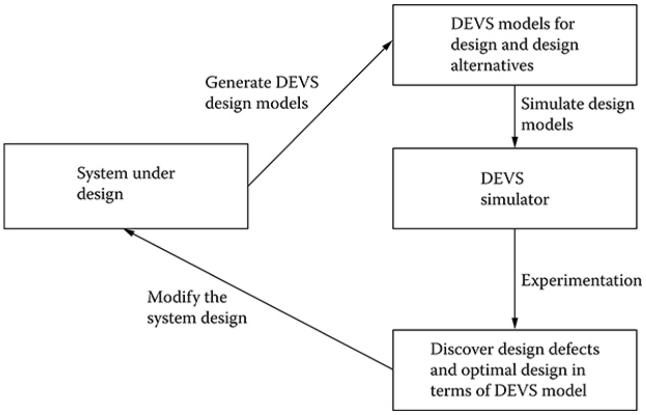

DEVS models for design and design alternatives
Generate DEVS design models
Simulate design models
System under design
DEVS simulator
Experimentation
Modify the system design
Discover design defects and optimal design in terms of DEVS model

Figure 4.3. Simulating with DEVS

DEVS has many benefits with its unique features, including fast hierarchical model construction/reconstruction, model reusability, and ease of model validation.

The flexible Modelling and Simulation (M&S) tools and environments that DEVS enables can help in discovering optimal system design quickly as well as in validating existing designs. Distributed DEVS can aid the design of distributed real-time systems in a more accurate and effective method as it can better represent physical systems, thus more accuracy in identifying the design defects and capturing the key design parameters that must be effectively manipulated.

4.2.2. Modelling with UML and its Real-Time profiles

Modelling languages should meet the following requirements for a comprehensive methodology of real-time systems:

- Consistency: checking for models expressed in different notations, design iterations, or different views of the same system, thus allowing grouping of requirements, structure, behavior, and analysis in a single, integrated system model.
- Traceability: requirements should be mappable to a precise specification of the system and from there to implementation, while the mapping should be kept current during the system evolution. Traceability also applies to models at different levels of abstraction.
- Realizability: since models represent partial specifications and different views on the system, realizability checks whether the models allow a system to be constructed such that all or some of the requirements are fulfilled.
- Distribution and integration: models should be capable of expressing concurrency and synchronization. Additionally, modelling should support overarching system specification, addressing the integration requirements as well as concerns that cut across the individual components, such as resource optimization across the integrated system.
- Interdisciplinary domains: since embedded systems design involves multiple domains, e.g., mechanical, electronics, and software, also since system components are often designed at different stages by different teams and using different tools and languages, modelling should ease integration and trade-off analysis, accordingly reduce the need for disruptive feedback iteration cycles.

- Non-functional properties: to allow specifying non-functional properties associated with behaviors, refinement relationships, and deployment models, e.g., performance, reliability, and power consumption.
- Resource models: a specification should support the modelling of platforms and resources, as well as the allocation and optimization of resources to meet the functional and non-functional requirements.
- Timing: a modelling notation should express timing requirements in various temporal models:
    - causal models, which are concerned only with the order of activities;
    - synchronous models, which use the concept of simultaneity of events at discrete time instants;
    - real-time scheduled models, which take physical durations and the timing of activities as influenced by the CPU speed, scheduler, and utilization into account;
    - logical time models, which consider that activities take a fixed logical amount of time, assuming that the platform can execute all activities to meet their constraints.
- Heterogeneous models of computation and communication: a modelling specification should support continuous behaviors, discrete event-based or time-based behaviors, or combinations thereof.

4.3. Modelling with UML

The Unified Modelling Language is a general-purpose modelling language widely used in academia and industry. It is a family of graphical notations underpinned by a single metamodel. UML can be used at different levels of the development process, in particular for requirement modelling and functional design, resulting in specification for behavior, structure, and Quality of Service (QoS) properties.

There are several UML-compliant modelling tools that support code generation to C/C++, Java, Ada, and different Real-Time Operating Systems (RTOSs). UML can be tailored for various domains or different target platforms. UML supports modelling structural, i.e., static, and behavioral, i.e., dynamic, views of a system. It provides 14 types of diagrams; the structural view includes class and state machine diagrams and the behavior view includes sequence and state machine diagrams.

All graphical notations in UML are backed by a single metamodel:

- The notation is the graphical syntax of the language.
- The metamodel defines the concepts of the language—the abstract syntax.
- The UML metamodel defines the language elements and the relationship between them in the different UML graphical notations.

The meta-layer hierarchy for any language generally has three layers:

- the metamodel, or the language specification, which defines how model elements in a model are instantiated
- the model
- objects of the model.

This layered structure can be applied recursively, such that the same layer that is a model instantiated from a metamodel can be seen as a metamodel of another model at the next lower level of instantiation. Figure 4.4 represents a schematic diagram of UML language and its components. After identifying requirements, architecture and design models show how the system is structured and how its internal entities behave to achieve system goals. All diagrams in Figure 4.4 can be used at this stage. Generally, a type of structural or behavioral diagram can be used both for logical and technical architecture. The models relevant to the logical architecture focus on capabilities and their mapping to logical entities. The models relevant to the technical architecture focus on deployment entities. A platform model adds details such as middleware, operating system, network, and resources. The technical architecture then expresses the mapping between the logical architecture and the platform.

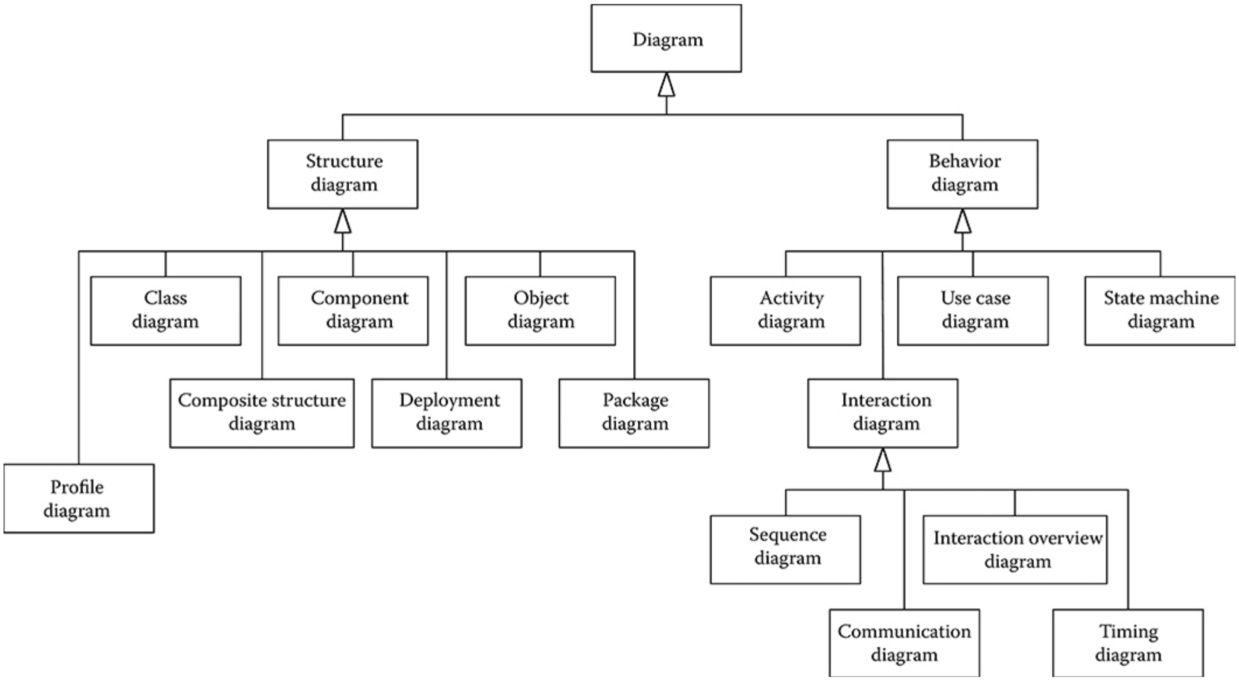


Figure 4.4. UML language architecture and components

4.3.1. UML example

BART is the commuter rail train system which automatically controls over 50 trains, most of them consisting of 10 cars. Tracks are unidirectional, and sections of the track network are shared by trains of different lines. A track is partitioned into track segments, which may be bounded by gates. A gate can be viewed as a traffic light, establishing the right-of-way where tracks join at switches. An illustration of the example is shown in Figure 4.5. The Advanced Automatic Train Control (AATC) system controls the train movement for BART and optimizes train speeds and the spacing between the trains to increase throughput on the congested parts of the network while constantly ensuring train safety. The system is controlled automatically. Onboard operators have limited responsibility, and they signal the system when the platforms are clear, so a train can depart a station and they can operate the trains manually when a problem arises.

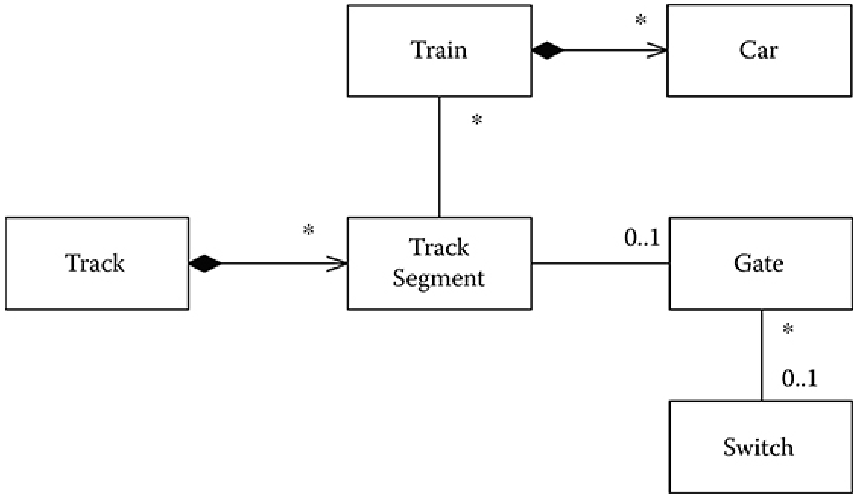


Figure 4.5. BART case illustration

Use case diagrams are useful in identifying the system boundaries and the external actors that interact with the system. Typically, in UML, actors are human actors that use an application, but in embedded systems, actors can be external physical resources such as devices and sensors. Actors represent logical roles, so a physical resource could play several roles in UML models. Figure 4.6 shows the use case diagram of the AATC.

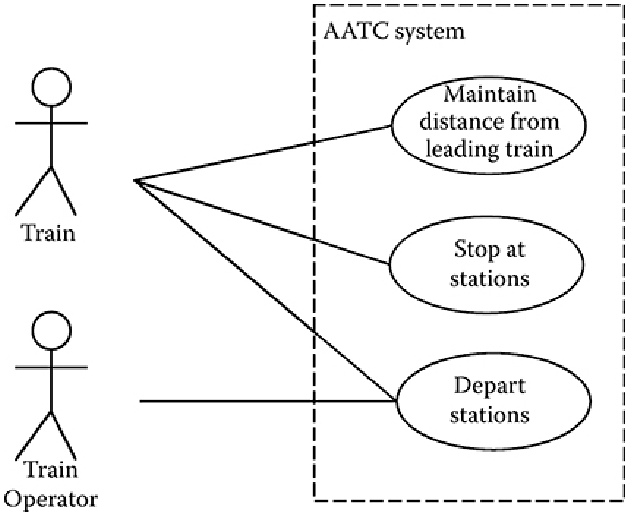


Figure 4.6. Use a case diagram of the AATC system

AATC consists of computers at train stations, a radio communications network that links the stations with the trains, and two controllers on board at the front and back of the train. A track is not a loop. Thus, at the end of the line, the front and back controllers exchange roles, and the train moves in the other direction. Each station controls a local part of the track network. Stations communicate with the neighboring stations. The system operates in half a second cycles. In each cycle, the station control computer receives train information, computes commands for all trains under its control, and forwards these commands to the train controllers. Figures 4.7 and 4.8 show commands issuing and checking train status diagrams.

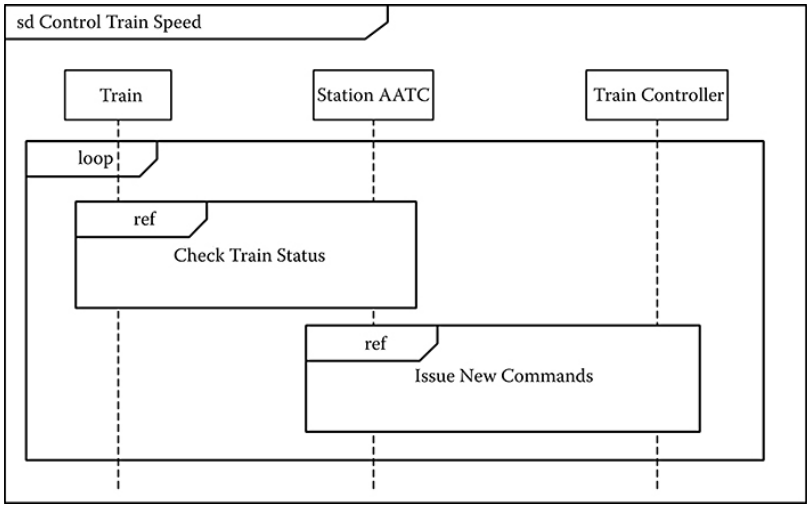


Figure 4.7. AATC system issuing new control commands

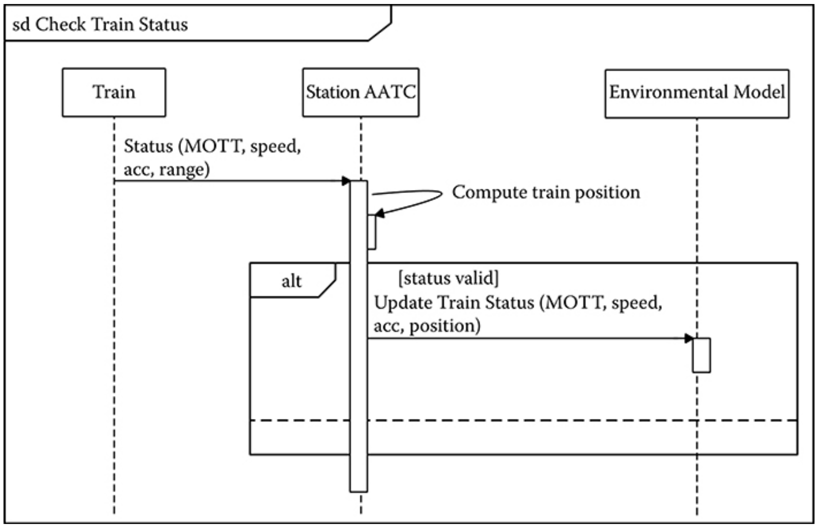


Figure 4.8. AATC system checking and updating train status

In Figure 4.9, the domain model of the BART system with the different roles is shown.

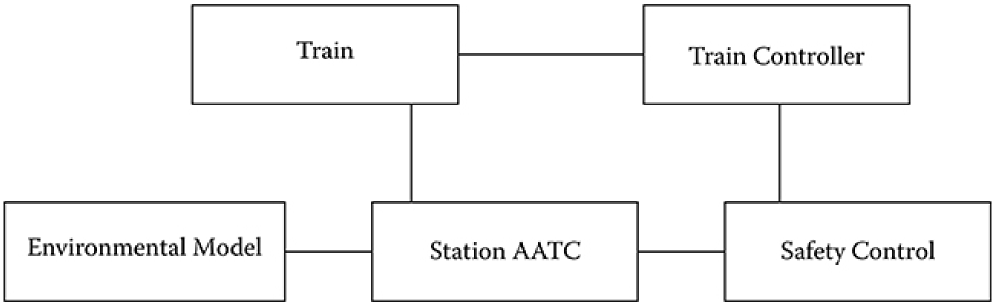


Figure 4.9. BART domain model with roles

Finally, in Figure 4.10, the behavior of the train controller as a state machine diagram with two states for normal operation and emergency mode is shown.

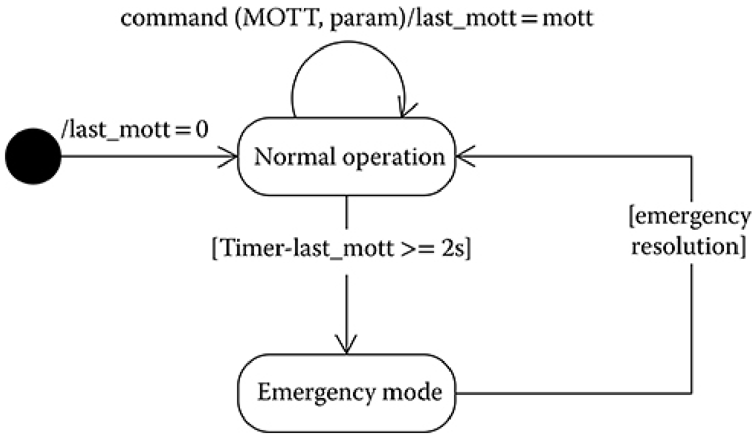


Figure 4.10. The train controller behavior as a state machine of two states

4.4. Modelling and Simulation of timing behavior with the timing definition language TDL

Time Definition Language (TDL) follows the time-triggered programming model. In a time-triggered system, all activities are triggered only by the ticks of a single global clock. To increase the range of applicability, TDL supports a limited form of event-triggered programming, which allows responding to hardware interrupts.

TDL has the following properties:

- Time determinism, which means that a program provides the outputs at the same times if it is provided with the inputs at the same times, where all times are relative to the program start.
- Value Determinism is when a program provides the same outputs if it is provided with the same inputs. TDL aims for both time determinism and value determinism. Thus, a TDL program provides the same outputs at the same times if it is provided with the same inputs at the same times.
- Portability, in the way that TDL programs represent a platform-independent description of the timing behavior of an application. Everything that is platform specific, for example, is defined outside the TDL program. TDL programs behave exactly the same independently of the underlying CPU, network bandwidth, or operating system.
- Transparent Distribution, which relates to portability, that a TDL application shows the same observable behavior in the case of a distributed system as on a single-node system. Thus, the fact that a distributed system is used as an execution platform is transparent.
- Time safety guarantees that a program behaves as expected for a particular target platform, given that the worst-case execution times for the tasks to be executed are known for that platform. In the case of a distributed platform, the compiler guarantees that the network communication preserves the expected observable behaviour application.
- Compositionality concerns the TDL modules that are executed in parallel and the data flow between them, in that way that adding another module to the application does not change the observable behavior of previously existing modules.

TDL constructs are:

- Modules: at the outermost level, a TDL application consists of a set of modules. Two modules can either be independent, i.e., share no data, or cooperate. Cooperating modules exchange data through ports. Statically, a module provides a namespace. Dynamically, modules are executed in parallel—possibly on different nodes in a distributed system. All modules share a common clock, which, in the case of a distributed execution platform, has to be distributed to the individual nodes of the platform.
- Ports: data flow within a single module, between multiple modules, and between a module and the physical environment is exclusively based on ports. A port is a typed variable that is accessed, i.e., read or written, at specific time instances only. Sensor and actuator ports are the only means for a TDL module to communicate with the environment. A sensor declaration defines a typed read-only variable to represent a particular value in the physical environment and provides input to the TDL application. An actuator declaration is an initialized and typed write-only variable that influences a particular value in the physical environment and provides output from the TDL application to the environment.
- Tasks: a task is a computational unit in TDL. It defines a namespace for input, output, and state ports. Each task is associated with a task function, which is a stateless piece of code without any synchronization points. A single invocation of a task at runtime creates a task activation. A task activation lasts for a strictly positive amount of time that starts at the release time and ends at the termination time. The time between these two instants is called the Logical Execution Time (LET) of the task activation. At the release time, the input ports of the corresponding task are updated with the values read from the output ports of others. Figure 4.11 shows the execution of a task and its associated times.

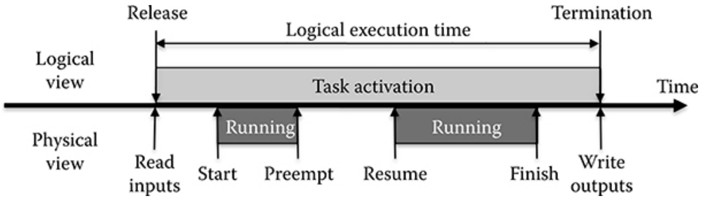


Figure 4.11. TDL task execution and associated times.

- Modes: modules that encapsulate a state machine have each a dedicated start mode and can switch between modes independently of others. A mode specifies a mode period – in microseconds – and a set of activities. As long as a module remains in a certain mode, the activities associated with it are repeated with its period. A mode activity is either a task invocation, an actuator update, or a mode switch.
- Asynchronous event-triggered activities: in addition to time-triggered synchronous activities, it is often necessary to execute event-triggered asynchronous activities. TDL supports asynchronous task invocations and actuator updates. An asynchronous activity is triggered either by an update of an output port, by the occurrence of a hardware interrupt, or by the tick of a timer that may potentially introduce its own time base. By integrating asynchronous activities into TDL, the TDL runtime system is able to provide the synchronization of the data flow between synchronous and asynchronous activities.

4.4.1. TDL Toolchain

TDL introduces appropriate abstractions to separate timing from functionality and platform-independent from platform-specific aspects. To obtain executable software, the textual TDL description has to be aggregated and combined with external functions that implement the required functionality. Figure 4.12 illustrates the separation of platform specifications from other functionalities.

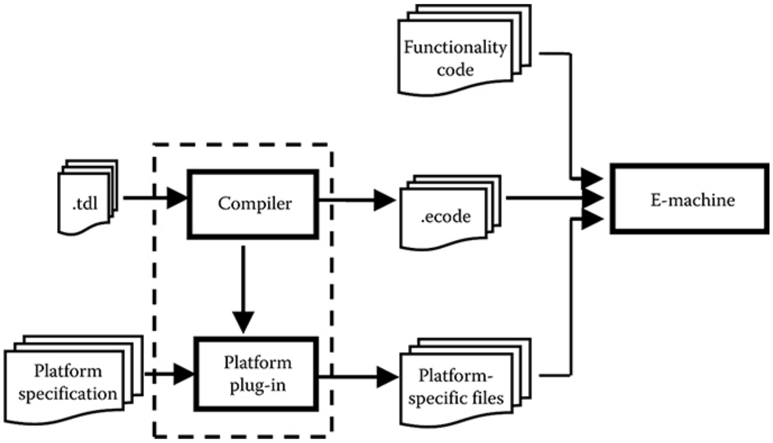


Figure 4.12. TDL abstraction separation of timing from functionality and platform aspects

4.4.2. TDL integration with MATLAB and Simulink

Simulink allows to model systems in a visual and interactive environment using block diagrams. Code generators can then automatically convert the block diagrams into program code, for example, in C code. It has been described that manually implementing LET semantics in Simulink is strongly discouraged. Even simple models of single-mode systems are cluttered with additional blocks to ensure that the timing behavior in the simulation conforms to LET semantics. It turned out that it is practically infeasible to model LET-based applications with multimodal behavior by hand, even when using the Simulink extension Stateflow. However, this could be overcome by means of the Toolchain for MATLAB since it provides the required separation between the model and the platform. In Figure 4.13, an example is shown where the Real-Time Workshop Embedded Coder (RTW-EC) tool is used.

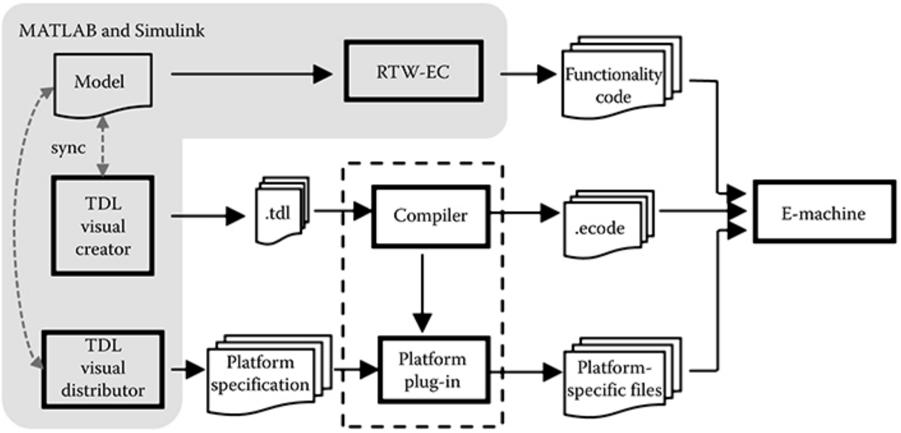


Figure 4.13. Simulink integration with TDL

5. REAL TIME SIMULATION FOR SYSTEM DESIGN

5.1. Progressive Simulation-Based Design for Networked Real-Time Embedded Systems

- Simulation is a valuable tool in the design and analysis of complex engineering systems. The use of fast simulations allows designers to experiment and analyze different design solutions without implementing the system in hardware. Real-time simulations are particularly useful in testing the real-time features of a system that interacts with the physical world and/or other hardware components. For example, testing with hardware components is beneficial in the design of real-time embedded systems such as mobile devices, manufacturing automation sensors/actuators, and Software-Defined Radio (SDR) systems.
- The design and implementation of these systems have been influenced by the demand for new products and recent advances in technology. However, the complexity and multidisciplinary nature of these systems make analytical modeling and analysis infeasible. Nevertheless, designs must be evaluated before proceeding with the implementation of an expensive solution. Traditional modeling and simulation can help in this goal, but their applicability is limited due to the gap between simulation models and implementation in hardware.
- One advanced technique frequently used in embedded systems development is Hardware-in-the-Loop (HIL) simulation, which involves replacing parts of a pure simulation with actual hardware. This allows unmodeled characteristics to be investigated and controls to be further refined. HIL is typically aimed at developing a single module in a larger system. However, HIL does not provide a general methodology that can scale to more complex and large-scale systems.
- The Progressive Simulation-based Design (PSBD) methodology is an advancement over HIL simulation. PSBD gradually incorporates actual system hardware into the simulated system and enhances the co-simulated model with each step before progressing further. The PSBD design process begins with simulating all models on computers and then replacing virtual components with real system components until all components are deployed and tested in the physical environment. The simulation model is continually updated whenever new design details are revealed, maintaining model continuity throughout the process.
- PSBD provides a systematic design process that focuses on transitioning from simulation models to system realization. Figure 5.1 illustrates the three stages of the PSBD design process, distinguished by the types of entities (virtual or physical) involved. The first stage involves conventional simulation (in fast simulation mode), where all models are simulated to develop the system model based on existing knowledge and assumptions regarding the physical system's hardware and operating environment. The second stage, virtual environment simulation (in real-time simulation mode), combines models and physical system components in a virtual testing environment to reveal any inconsistencies in behavior between simulation models and physical system components. This stage brings the simulation-based study closer to realization by incorporating physical system components into the simulation, with the objective of discovering overlooked design details and improving control models/algorithms under development. Finally, the third stage involves physical system experimentation, where the physical system is tested in its target operating environment.

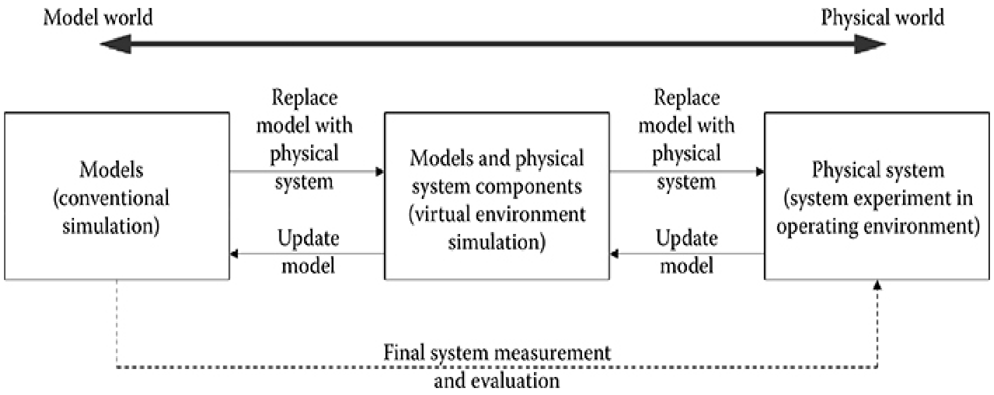

Model world
Physical world
Models (conventional simulation)
Replace model with physical system
Update model
Models and physical system components (virtual environment simulation)
Replace model with physical system
Update model
Physical system (system experiment in operating environment)
Final system measurement and evaluation

Figure 5.1. PSBD design stages

- The PSBD methodology involves two parallel activities, namely replacing models with physical system components and updating models in a progressive manner. At the end of the design process, a system model representing the system is developed along with the realized and tested system. This system model can be used for system measurement, evaluation, maintenance, and future development. Model continuity and virtual environment simulation are two significant features of the PSBD methodology. Model continuity refers to the ability to transition model specification through the stages of the development process as much as possible. The virtual environment simulation creates a virtual testing environment by combining physical and virtual system components, bridging the gap between conventional simulations using all models and physical system experiments using all physical system components. To support the virtual environment, simulation techniques are developed to synchronize and sense the physical and virtual system components.
- PSBD offers a number of advantages over HIL simulation when designing complex networked systems. Firstly, it brings simulation-based study closer to reality and provides valuable insights for designers. Secondly, it increases confidence in the final system's operation. Thirdly, it emphasizes a systematic design process that gradually replaces simulation models with physical system components. Lastly, the virtual environment simulation enables designers to experiment with their designs in a flexible virtual testing environment.

5.1.1. Bifurcated Design Process for Networked Real-Time Embedded Systems

- The design process for networked systems is bifurcated into the design of a single node and the design of the networked system, as depicted in Figure 5.2. Design on a single node involves the design of various functional modules, including sensing, modulation/demodulation, and channel coding, of the embedded device. On the other hand, the design of a networked system focuses on how multiple nodes work together as a whole, including designing and improving the communication protocols and the cooperative strategies among the nodes. Both design processes follow the PSBD process, which begins with models and gradually adds more physical system components.
- Incorporating hardware components into the design is crucial, and HIL simulations are performed to test the designed modules and algorithmic code's compatibility with the hardware. The process progresses gradually as more and more hardware components are added. When designing a

networked system, the first step is to develop individual node models and a model of the communicating network.

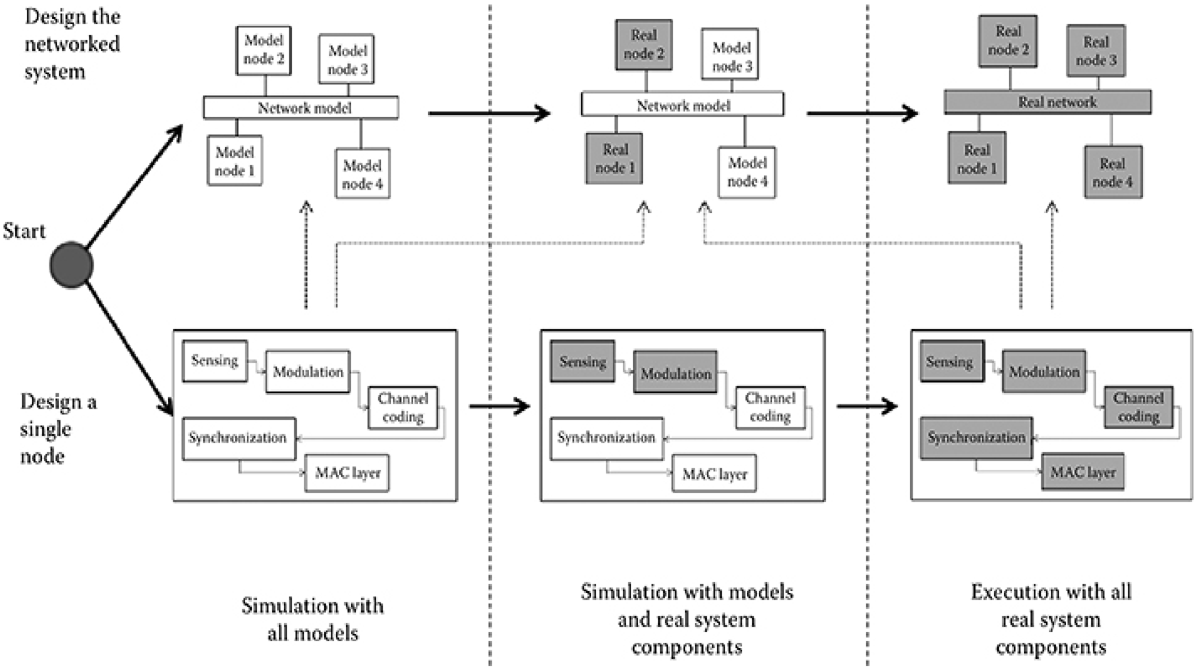


Figure 5.2. Bifurcated design process of a single node and networked system

5.1.2. Cognitive Radio CR Modem Example

- An example of the design blocks of a Cognitive Radio (CR) modem is presented in Figure 5.3. The CR modem carries out sensing, transmission, and reception functions, which are depicted by the different blocks.

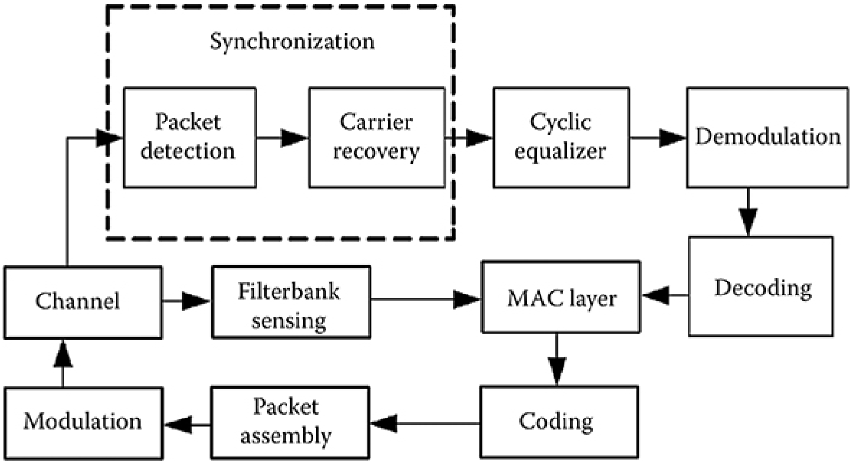


Figure 5.3. A single cognitive modem

- Figure 5.4 shows the progressive simulation of the CR modem. In figure 5.4, (PUs) represent the simulated Primary Units, the (SU) represents the Secondary Unit, the rectangles represent DEVS models simulated on a personal computer, and parallelograms represent implemented modules on the software-defined radio board. Regarding the actual implementation, it starts with the sensing module, and more simulated models are gradually implemented.

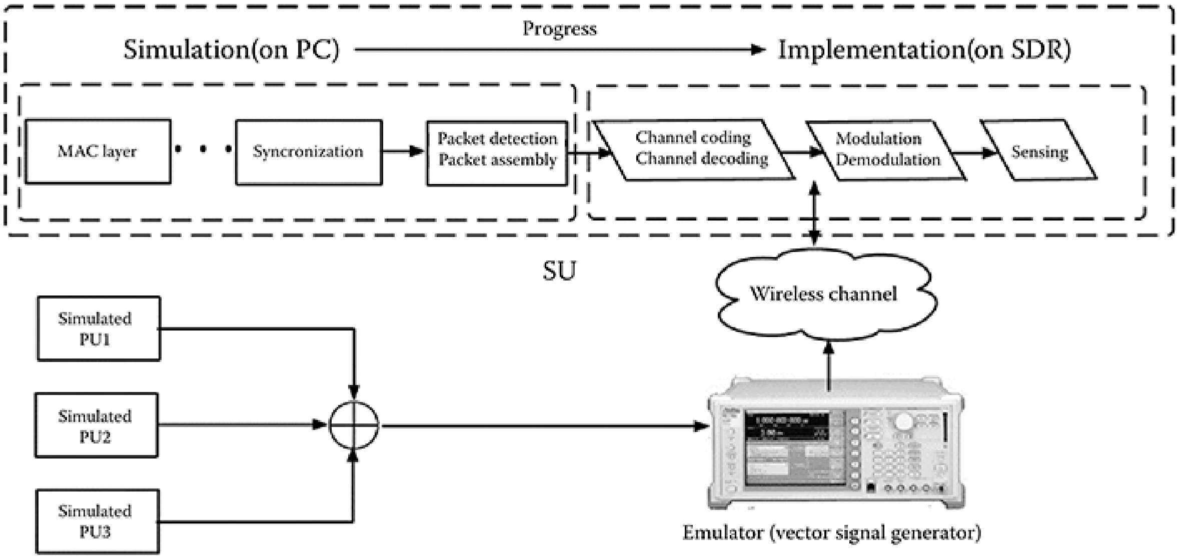


Figure 5.4. Progressive simulation of the CR modem

5.2. Validator Tool Suite Filling the Gap between Conventional Software-in-the-Loop and Hardware-in-the-Loop Simulation Environments

- Simulation is an effective method for testing embedded systems before deploying them in the real world. In simulation, the plant is represented by a software model that runs on a host computer, such as a personal computer. In HIL simulation, the entire embedded system is operated in closed loop with the plant model, which is executed in real time on a dedicated computer. This enables HIL simulations to verify the real-time properties of the embedded system. Any difference between the behavior of the embedded system in a HIL simulation and in the real world is due to the abstractions made in plant modelling. On the other hand, in SIL simulation, the embedded software is executed on a host computer different from the target platform, in closed loop with the plant model. Both the controller and the plant model simulations are usually run on the same host computer. The SIL model of an embedded system includes the embedded software and an abstraction of the target platform, which determines how close the software execution in the SIL simulation is to the HIL simulation, given that the same plant model is used. The level of abstraction ranges from a minimal representation of the target platform that allows testing only functional, transformational, or processing properties of the software to full-fledged hardware simulators like Instruction Set Simulators (ISS), which provide system behavior close to a HIL simulation while also offering better observability of software executions.
- In Figure 5.5, the difference between HIL and SIL is shown.

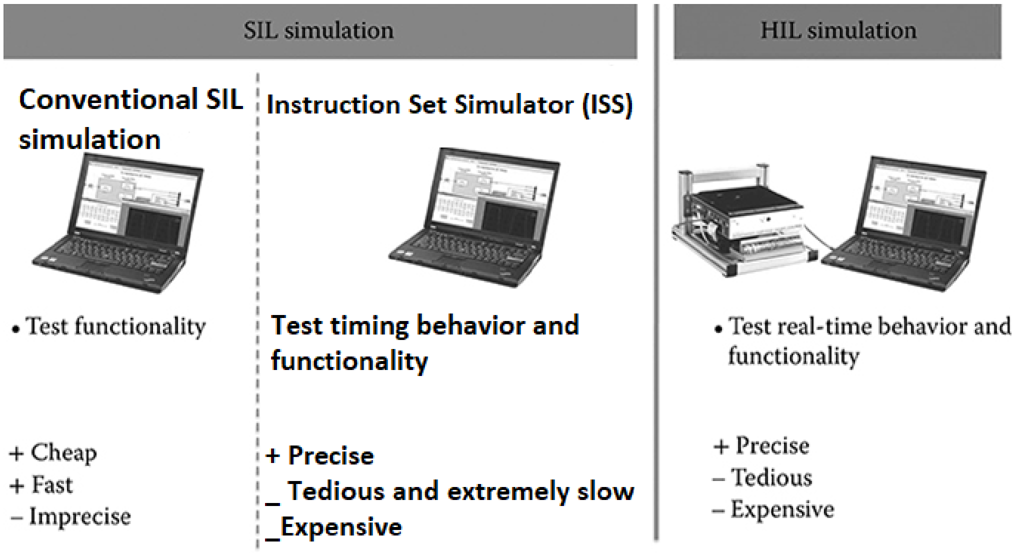

SIL simulation
HIL simulation
Conventional SIL simulation
Instruction Set Simulator (ISS)
• Test functionality
Test timing behavior and functionality
• Test real-time behavior and functionality
+ Cheap
+ Fast
– Imprecise
+ Precise
_ Tedious and extremely slow
_Expensive
+ Precise
– Tedious
– Expensive

Figure 5.5. Differences between HIL and SIL simulation techniques

- While pure functional SIL simulations are speedy, they lack the ability to test the timing properties of an embedded system. In contrast, ISS can analyze timing, but it tends to simulate the entire hardware, resulting in sluggish and costly simulations. To address this, the Validator simulation package replaces ISS. Figure 5.6 illustrates how the validator simulation works, showing that it outperforms ISS by separating the simulation of the plant and controller. The plant is simulated using traditional tools like MATLAB and Simulink, while the controller is simulated using the Validator suite, resulting in improved performance and reduced simulation costs.

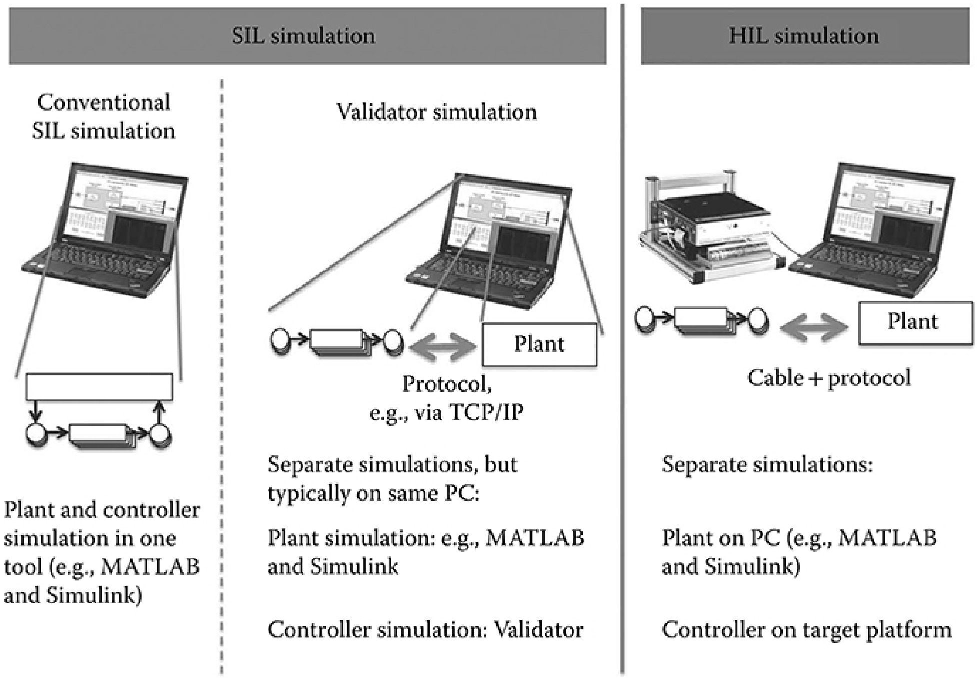

SIL simulation
HIL simulation
Conventional SIL simulation
Validator simulation
Plant
Protocol, e.g., via TCP/IP
Plant
Cable + protocol
Plant and controller simulation in one tool (e.g., MATLAB and Simulink)
Separate simulations, but typically on same PC:
Plant simulation: e.g., MATLAB and Simulink
Controller simulation: Validator
Separate simulations:
Plant on PC (e.g., MATLAB and Simulink)
Controller on target platform

Figure 5.6. The use of the Validator Tool Suite with SIL

5.2.1. Architecture of the simulation with the validator

- In Figure 5.7, it can be seen that the Validator simulation facilitates a closed-loop co-simulation of the plant under control and the controller tasks. The Validator is capable of supporting continuous-time plant models in MATLAB and Simulink by implementing a MATLAB and Simulink S-function for communication. TCP/IP is the underlying communication protocol between the plant and the simulation with the Validator. To optimize efficiency, both simulations can run simultaneously on the same computer or different cores, or even different computers.

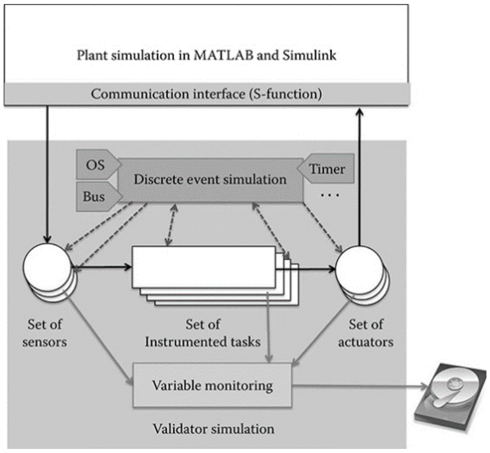

Plant simulation in MATLAB and Simulink
Communication interface (S-function)
OS
Bus
Discrete event simulation
Timer
. . .
Set of sensors
Set of Instrumented tasks
Set of actuators
Variable monitoring
Validator simulation

Figure 5.7. Validator simulation architecture

- To achieve accurate simulation of the controller, it's essential to provide the validator with configuration information about the target platform while setting up the simulation. This is done by adjusting the properties of the corresponding model components. The Validator library includes different types of actors for specifying the target platform, such as hardware actors that model the functionality and timing of common hardware components like interrupt controllers, timers, bus controllers, hardware sensors, and hardware actuators. Additionally, operating system actors are also available in the library, which implement the functionality of the operating system on the target platform, including scheduling, resource management, and communication between tasks.

5.2.2. Related work

- Co-Simulation: Co-simulation is a methodology aimed at validating the functionality of hardware and software components by simulating two or more system parts described on different levels of abstraction. However, a significant challenge lies in establishing an interface between different abstraction levels. Ideally, the simulation of a system should be possible throughout the entire design process, with a model of the same component refined iteratively. Co-simulation serves as a basis for co-design and system verification, but the accuracy-performance trade-off must be considered. In Hardware-Software (HW/SW) co-simulation, the processor model connects hardware and software models. While gate-level modelling is the most accurate, it is also the slowest with clock cycle accuracy, or instruction-set level models providing faster alternatives. Some co-simulation tools implement a synchronization handshake instead of modelling the processor, and some even offer a virtual operating system to emulate or simulate the hardware. ISS is a popular choice for obtaining correct timing information, despite their slow nature due to fine granularity simulation. To address performance issues, techniques such as caching and distributed simulation are employed.

- Modelling and Simulating Legacy Code: Several techniques are available for generating models from legacy code, but only a handful of them incorporate timing considerations. One method involves reverse engineering the software and identifying corresponding modeling constructs in a modeling language to recreate the same functionality, which often results in intricate models that are not easily understandable and therefore do not contribute to gaining new insights into the embedded software system. Another approach involves code instrumentation and task execution delay to achieve a specific behavior. This approach leverages code instrumentation to generate timed nets from the legacy code, capturing the blocking behavior of the software at points where shared resource locks are accessed. A controller is then synthesized from the code and utilized at runtime to ensure deadlock-free behavior of the software on multicore platforms by delaying task executions that could cause deadlocks. The objective of the validator is to reproduce the real-time behavior of a given application without altering its functional behavior. When implementing these processes or frameworks, it is crucial to consider the utilized platform, its functionality, and limitations. Modeling at an abstraction level and providing platform-based designs can be more effective.

5.3. Modern Methodology of System Design Using Rapid-Control Prototyping and Hardware-in-the-Loop

- A simulation refers to representing the features or operation of a system using another system. In discrete-time simulation, time moves forward in fixed time-steps or slices of equal duration, and each variable or system state is successively solved as a function of the preceding time slice's variables and states to solve mathematical functions and equations at a specific time step. Explicit or semi-explicit solvers, such as state space networks, are preferred in real-time simulation as their computational time at each iteration is predictable. However, the duration required to compute all equations and functions representing a system during a given time slice can be shorter or longer than the duration of the simulation time-step. The accuracy of the computations during simulation is determined not only by the precise dynamic representation of the system but also by the time taken to produce results. Figure 5.8 demonstrates the chronological principles of a real-time simulation, where the real-time simulator used must produce the simulation's internal variables and outputs within the same duration as its physical counterpart to be valid.

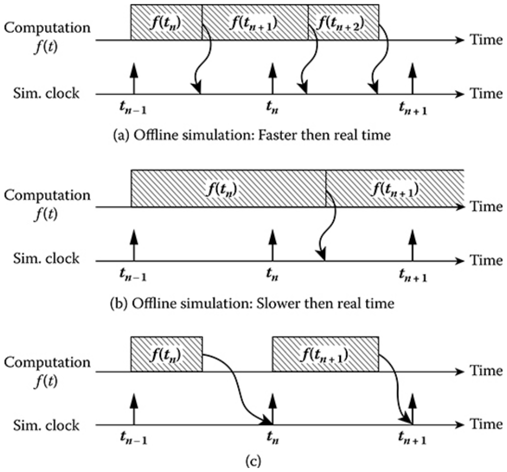


Figure 5.8. The chronological principles of real-time simulation

- Real-time digital simulation involves solving model equations at discrete time steps with proper step size duration to accurately represent the system's frequency response up to the fastest transient of interest. The simulator executes the same series of tasks for each time step, such as reading inputs and generating outputs, solving model equations, exchanging results with other simulation nodes, and waiting for the start of the next step. The states of externally connected devices are sampled once at the beginning of each time step of the simulation, resulting in the communication of the states of the simulated system to external devices only once per time step. If all timing conditions of real-time simulation are not met, overruns can occur, leading to discrepancies between the simulator results and the physical counterpart's response.
- The attainment of real-time performance is distinct from synchronous performance. Nonlinear systems do not ensure that switching events occur at a specific discrete-time moment, and multiple events can take place within one-time step, with the simulator being aware of only the last event. To address these concerns, discrete-time compensation techniques are proposed, which usually incorporate time-stamping, interpolation algorithms, and advanced I/O cards that operate at a considerably faster sampling rate than fixed-step simulation. These techniques are used to convey not only state information but also timing information about the moment at which a state change occurs. Figure 5.10 depicts the timing principles of a real-time simulator using a discrete-time compensation technique.
- Finally, to avoid complications that may arise when utilizing various simulation tools and differing step sizes, a technique known as co-simulation is employed. This involves transferring data between tools while upholding its integrity, effectively overcoming any issues that may arise due to synchronization discrepancies.

5.3.1. Rapid Control Prototyping

- The implementation of a plant controller in Rapid Control Prototyping (RCP) involves using a real-time simulator that is connected to a physical plant, as depicted in Figure 5.9. RCP offers several advantages over creating an actual controller prototype. Firstly, developing a controller prototype using a real-time simulator is more flexible, less time-consuming, and easier to debug. Secondly, the controller prototype can be tuned or modified with just a few mouse clicks. Finally, as every internal controller state is accessible, debugging an RCP is faster, and there is no need to remove its cover.

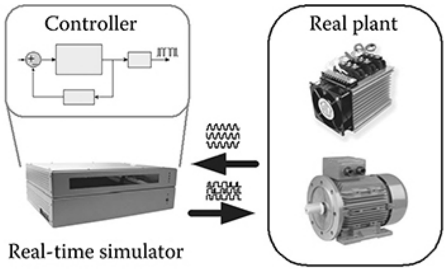


Figure 5.9. Rapid control prototyping

5.3.2. Application categories

- Hardware-in-the-Loop (HIL): Hardware-in-the-Loop (HIL) is a technique where a physical controller is linked to a virtual plant that is executed on a real-time simulator, as illustrated in Figure 5.10. By connecting the implementation of a controller using RCP to a virtual plant via HIL, testing of controllers can be carried out early, even when physical test benches are not accessible. Furthermore, virtual plants usually come at a lower cost and have parameters that have less standard deviation caused by manufacturing processes or environmental variations. In addition to the benefits of RCP, HIL offers an added advantage of early controller testing.

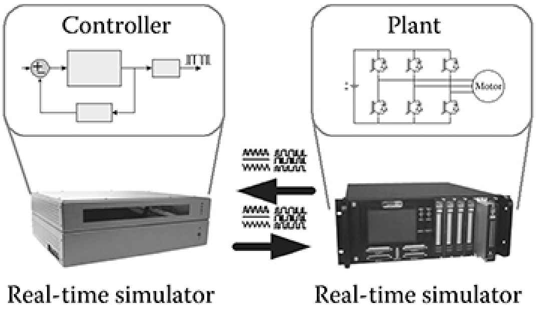


Figure 5.10. HIL using RCP

- Software-in-the-Loop (SIL): Software-in-the-Loop (SIL) can allow both the controller and plant to be simulated in real-time on a powerful simulator, as depicted in Figure 5.11. One of the benefits of SIL over other simulation techniques like RCP and HIL is that it does not use any I/O's, thereby maintaining signal integrity. Furthermore, as both the controller and plant models run on the same simulator, timing with the external environment becomes less critical, and execution time can be adjusted without affecting the validity of the results. SIL can also be used to perform accelerated simulation, where simulations run faster than real-time, allowing a large number of tests to be conducted in a short period. This technique is particularly suitable for statistical testing such as Monte Carlo simulations. Additionally, since SIL does not require any physical devices to be connected, it can be carried out at non-real-time speeds.

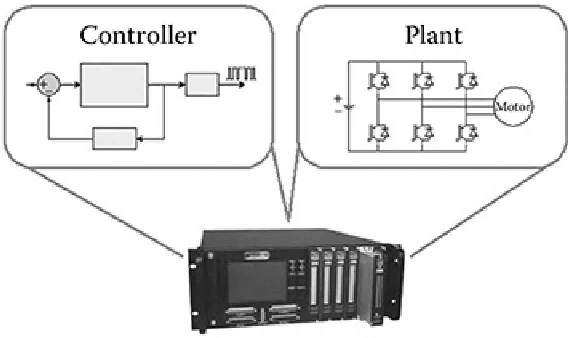


Figure 5.11. SIL using RCP

5.3.3. Model-Based Design Using Real-Time Simulation

- Model-based Design (MBD) is a powerful approach to tackle the challenges associated with the design of intricate systems. The technique leverages mathematical and graphical tools to create models that efficiently communicate knowledge about the system under development. The workflow, commonly known as the "V" diagram, as illustrated in Figure 5.12, forms the basis of MBD methodology, enabling various developers to work on a design and modelling project seamlessly. The MBD process involves four fundamental steps, namely: building the plant model, analyzing the plant model and synthesizing a controller, simulating the plant-controller combination, and deploying the controller. These steps ensure a systematic and organized approach towards MBD and enable developers to achieve desired outcomes with ease.

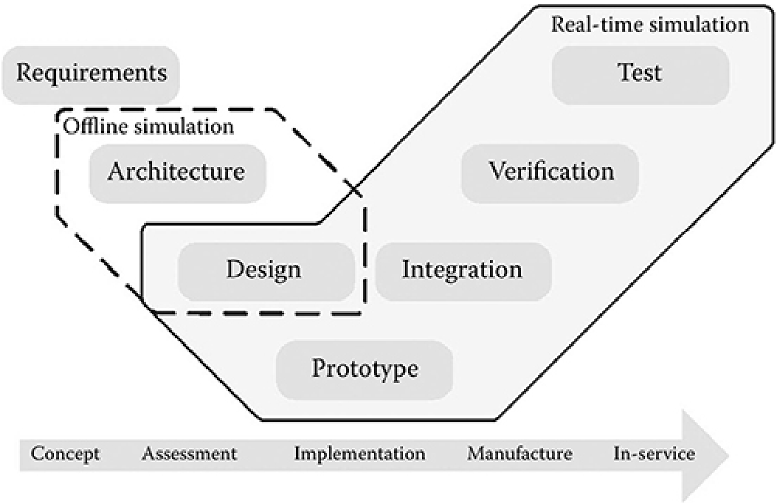


Figure 5.12. Model-based design V diagram

- One important aspect to consider when using a real-time simulator is that the model can be modified online, allowing for continuous reading and updating of any of its parameters, unlike a physical plant. This feature provides greater flexibility and adaptability to the model during execution. Additionally, all model variables are accessible in real-time during execution when using a simulator, which is not typically possible when working with a physical plant.

- The ability to configure models online and access complete data sets enables previously unthinkable applications. For instance, one such application involves testing a controller's ability to adapt to changes in the plant's dynamics. With this technology, developers can verify if a controller can compensate for changes caused by component aging. Such applications bring the concept of **Digital Twins**.
- RCP, due to its versatile capabilities, finds its application in various fields such as automation, aerospace, power generation, and hybrid vehicle testing. The HIL technique is crucial in many of these applications, especially in testing the fuel cell performance of hybrid vehicles. Moreover, SIL is equally important in running numerous random and diverse operational tests, such as in testing smart grid scenarios.

6. PARALLEL AND DISTRIBUTED REAL TIME SIMULATION

6.1. Test Bed for Evaluation of Power Grid Cyber-Infrastructure

- The power grid is an essential component of our infrastructure, providing the energy needed to carry out our daily activities. Upgrades to the power grid have been taking place for some time, with the introduction of the "smart grid." While the existing architecture is a combination of old and new machines and protocols, as well as vastly different networks across various domains, it is by no means "dumb." The power grid is a complex system that incorporates both physical and cyber devices. The power grid is comprised of three main parts: generation, transmission, and distribution. Generation occurs at power plants, which can derive energy from various sources such as nuclear, coal, or wind. Transmission consists of high-voltage lines that transport energy from generation facilities to areas of consumption. Distribution occurs at a local level and involves stepping down the high-voltage energy to lower-voltage energy suitable for consumers.
- Each of the three domains that make up the power grid - generation, transmission, and distribution - face their own set of challenges related to security and safety. For example, while security measures such as armed guards and hardened firewalls are in place at generation sites, the greater concern is safety, as an abnormal operation of the plant could pose a risk to people and equipment. In the distribution sector, protecting private data from being exposed is a key issue with the new smart grid. The transmission section of the grid has its own set of unique concerns, as many of the control systems that regulate transmission lines are now connected to the internet, making it necessary to address potential security risks.
- In the past, Modbus was used as the primary communication protocol for devices on the grid, while protocols like DNP3 and IEC 60870-5 were utilized for long-distance communication between substations and control stations, as well as between substations and RTUs in the field. However, there is now a trend towards implementing the IEC 61850 protocol for communication, reflecting a growing awareness of the need for security in SCADA systems. Due to the critical nature of the power grid, development must be approached with caution to ensure that modifications do not compromise the system's security or functionality. Physical testing is essential for understanding how equipment will behave in different scenarios, but creating a full-scale test system for the power grid is impractical. To address this, RT simulators and test beds have become increasingly important for conducting thorough testing of new systems and modifications before implementation.
- The Virtual Power System Testbed (VPST) serves as a platform for testing new technologies in a realistic environment, providing clear feedback on their effectiveness. Detailed simulation is particularly valuable for certain technologies such as cryptography. For example, cryptography can be implemented in SCADA to investigate the feasibility of retrofitting bump-in-the-wire devices, test various forms of key management, or examine the efficacy of using puzzles to confirm identity in large-scale networks or protocols like DNP3 with Secure Authentication or DNPSec. The VPST offers a valuable resource for such testing, enabling the identification of potential issues and improvements in a controlled setting.

6.1.1. Introduction to DNP3 protocol

- The Scalable Simulation Framework (SSF) is a versatile framework that can be expanded to accommodate intricate systems such as computer networks, raytracing, and fluid dynamics. One of the network simulators that is based on SSF is RINSE, which has a variety of features that make it ideal for large-scale simulations. SSF partitions graphs into sub-models, allowing for highly parallelizable models. These sub-models are arranged in a way that reduces communication between them to a minimum, enabling multicore systems to take full advantage of their potential. RINSE is capable of supporting multiple resolutions, allowing it to adjust the fidelity of a simulation to ensure that it runs in real time. RINSE employs a fluid model for traffic and allows both full-fidelity traffic and fluid models to exist within the same simulation. By utilizing these models for network topologies, RINSE achieves a significant speedup over using a full-resolution model. Moreover, RINSE can run simulations faster than real-time, making the simulator time-independent of wall-clock time, which is critical when interfacing with actual devices through emulation. Finally, RINSE is modular in such a way that new protocols and models can be developed with ease.
- In order to simulate the power grid using RINSE, it is necessary to establish the communication protocol that virtual SCADA devices will use to interact with one another. When simulating a new device or protocol, it is important to consider the desired level of accuracy. It may not be possible to estimate the new model by adjusting the parameters of a different model, as there may be fundamental characteristics that cannot be accurately represented by a different model. Therefore, to capture the security assumptions of a given protocol, it is necessary to model the protocol as accurately as possible, given the scale of the network being simulated. In the case of RINSE, this can be achieved by modeling the Distributed Network Protocol 3 (DNP3) protocol with a high degree of accuracy.
- The DNP3 protocol, depicted in Figure 6.1, is designed as a stack of three main layers, namely the data link layer, the pseudo-transport layer, and the application layer. One of the key benefits of DNP3 is that it provides interoperability and an open standard to device manufacturers, allowing it to function with any network. The physical medium used is typically either Ethernet or RS-485. The Data Link Layer can be run directly on the physical medium or it can be encapsulated by other protocols. This layer is responsible for providing framing information and reliability. The DNP3 Data Link Layer facilitates point-to-point communication, including addressing information, information about the direction of travel, the initiator of the communication, and the type of frame and its function. The Pseudo-Transport Layer is used to support fragmentation, with the DNP3 Pseudo-Transport Layer being responsible for segmenting the data from upper layers into lower-level frames. It is a simple layer that consists of only a byte. The Application Layer acts on behalf of the user and is responsible for requesting, confirming, sending, and receiving requests and data. The DNP3 Application Layer is composed of two sections: the Application Protocol Control Information (APCI) and the Application Service Data Unit (ASDU).

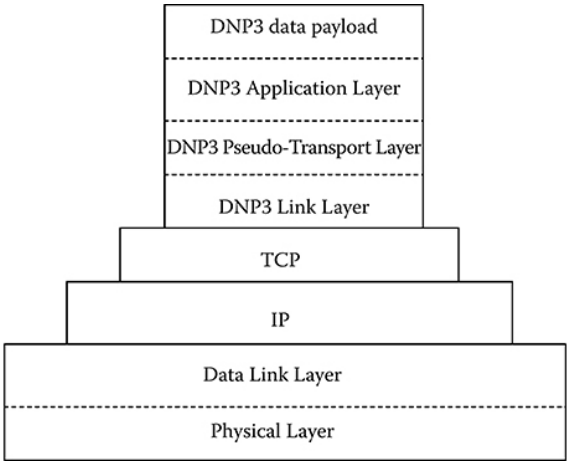


Figure 6.1. DNP3 protocol stack

6.1.2. Attacking DNP3

- It is crucial to model DNP3 accurately because of the numerous deficiencies in the protocol, which make it highly susceptible to compromise. Moreover, the vulnerabilities that DNP3 poses to SCADA systems are documented in attack trees, fault trees, and other risk analysis methods that are related to CIs. To mitigate these vulnerabilities, countermeasures for the protocol can be tested in a simulation environment before being integrated into the standard. One possible solution is to encapsulate DNP3 in another protocol such as SSL/TLS or IPSec, which would enhance the security of the communication channel.

6.1.3. Modelling DNP3

- The simulator's protocol must be able to handle all of its stack layers, where the three layers are combined as a payload and transported by TCP/IP, with IP headers used for packet routing instead of the Data Link Layer header. When working with emulated packets, the Data Link source and destination fields direct packets to and from the correct hosts. Emulated packets are crucial to the virtual DNP3 model as compliance with physical devices enables numerous use cases.
- Without external communication, the RINSE model would provide limited utility, as background traffic, metrics regarding correctness and scaling of technologies, and insight into large-scale SCADA networks would not be available. By being interoperable with physical equipment, the virtual DNP3 model can offer a wider range of use cases.
- To ensure scalability, it is suggested to use an abstract model view instead of full-fledged implementation of the DNP3 stack. Since there is a possibility of modeling hundreds of thousands of relays, modeling the full functionality of DNP3 may not be the most efficient option. Instead, the focus should be on supporting two classes of reads with only a few object types and one type of command, which simplifies the control flow, allowing for quick computation and low-latency replies.
- While the IP layer of most simulators provides routing, and the industry is moving towards encapsulating DNP3 to take advantage of its routing strengths, it is still recommended to use the DNP3 Data Link Layer. This is because there may be unknown interactions between layers that could

potentially impact the proper functioning of the Application Layer. For example, if an adversary tampers with a field in the Data Link Layer, it could lead to malfunctioning of the Application Layer.

- Consolidating the three layers into a single layer can speed up simulation time, but it also means that DNP3 alone cannot offer its entire range of functions. If direct modeling of DNP3 on the physical link is necessary, the DNP3 implementation would have to be revised. Additionally, RINSE's routing method would need to be rewritten because it currently uses IP for routing.
- The virtual DNP3 protocol is utilized by virtual hosts to communicate with one another. A host, which is a node in the simulator that possesses some level of computational power, models a physical device. Hosts such as relays and data aggregators are simpler to simulate than control stations, which have a more complex structure, are proprietary in nature, and require heavy customization. In cases where it is necessary to simulate control stations and similar components, it is recommended to rely on physical control stations as they provide the added benefit of allowing for the simulation network to be viewed as part of an operational network. Virtual hosts represent a purely cyber portion of the network.

6.2. System Approach to Simulations for Training: Instruction, Technology, and Process Engineering

6.2.1. Introduction to Distributed Real-time simulation-based training

- Distributed simulations are networks consisting of simulators situated in different locations that collectively execute a single model or share a common space. In contrast, real-time simulations advance at the same pace as actual time, while training simulations use various educational methods, such as scenario-based training, to provide a problem-solving environment that guides practice. Distributed real-time Simulation-Based Training (SBT) entails linking two or more geographically dispersed computer-based simulations through a network to track real-time progress. Multiple trainees can use SBTs simultaneously, and their inputs can impact the shared environment. SBTs offer educationally supported, authentic problem-solving contexts that allow trainees to gain or improve knowledge, skills, and attitudes. Distributed real-time SBT is a facilitative technology that supports joint training and mission rehearsal, tactic and technique development and testing, and personnel skill assessment. Some of the advantages of SBT include reduced risks, time and cost savings, and improved operational capabilities.

6.2.2. SBT CHALLENGES

- Distributed real-time simulation has emerged as a crucial training technology, offering numerous benefits. However, this approach also brings its own set of challenges. These challenges include technology-related issues, establishing and accomplishing distributed simulation instructional objectives, and determining the best way to implement SBT to reap its advantages. To be more specific, the main challenges faced in this domain are as follows:
    - Interoperability, which refers to the ability of various systems to link up and exchange data among their federates. While interoperability has come a long way in recent years, there are still significant technological gaps that remain. For instance, there are limitations in the software protocols, including scalability issues, time synchronization problems, and lack of plug-and-play capabilities. There is also a lack of interoperability

among different protocols and insufficient support for semantic interoperability. Another challenge is posed by differences in domain architectures, which may make it difficult to upgrade systems or add new components to government domain architectures. Furthermore, different versions of an architecture may not necessarily support federation. To overcome these challenges, standardization and the creation of homogenous systems are encouraged. Another challenge is ensuring "fair play," which involves guaranteeing that no trainee has an unfair advantage due to technical issues outside of the training, such as improved graphics or faster processors. This challenge is primarily associated with synchronization or model composability. To address this challenge, it is advisable to use similar or closely-matched simulation systems.

- Simulator fidelity, since it plays a crucial role in enhancing the learning environment. The SBT approach focuses on achieving a high level of realism in the simulator to replicate the real world as accurately as possible. This not only boosts confidence in the simulator's performance but also ensures that the training effectively supports the learning of the relevant skills. Achieving physical and functional correspondence is vital when designing a simulator to realistically and comprehensively duplicate a real-world system or equipment. Moreover, psychological fidelity is also important as it ensures that the visuals and experience provided by the simulator are similar to the real-world scenarios. Although high physical fidelity is not always necessary, the overall fidelity configuration should align with the educational goals of the system to ensure a successful simulation.
- Instructional strategies are an essential aspect of simulations and are influenced by cognitive, behaviorism, and constructivism learning theories. These theories have significantly contributed to the development and widespread use of simulation-specific instructional strategies, including event-based or scenario-based training approaches. To enhance the learning experience and prevent learning errors, it is recommended to avoid unstructured training. Instead, scenarios should be systematically organized around predictable objectives and employ guided-practice principles. By doing so, learners can acquire new skills and knowledge more effectively and efficiently.
- Instructor workload is a crucial factor to consider in distributed training as it can significantly increase when dealing with multiple trainees and entities. In such cases, instructors have to configure and initialize system setups, monitor distributed trainees and entities during the exercise, and manage the delivery of distributed post-exercise feedback, which can be time-consuming and challenging. However, to overcome this challenge, the use of automated tasks is encouraged. This approach can significantly reduce instructor workload by automating several routine tasks, allowing instructors to focus on more complex and strategic aspects of the training exercise.
- One significant issue with simulators is the lack of effective assessment, which refers to the inability to determine the effectiveness of training in simulators as no empirical results exist or have been conducted. In some cases, training effectiveness is evaluated after the acquisition and integration of simulators. To overcome this challenge, there is a need for general guidelines and guidance that cover various domains and areas. Furthermore, highlighting the benefits of SBT in terms of error prevention, impact assessment, and cost savings can encourage the development of more guidance and

guidelines. By doing so, it can help address the issue of effective assessment and enhance the overall effectiveness of simulators as a training tool.

- Finally, the utilization of distributed simulation has been restricted to certain sectors due to several factors. Firstly, the limited integration with commercial off-the-shelf simulation packages has hindered its widespread use. In addition, technical challenges relating to federating systems, inefficiencies of synchronization algorithms, bugs and lack of verification in distributed models, overly complex runtime management, and a perceived lack of practical return-on-investment have further contributed to the limited use of distributed simulation. Moreover, some existing distributed packages have been criticized for offering too much functionality that may not be relevant to industry needs. Addressing these challenges will be essential in expanding the use and applicability of distributed simulation across industries.

7. TOOLS AND APPLICATIONS

7.1. Toward Accurate Simulation of Large-Scale Systems via Time Dilation

- Validation of distributed systems, especially those that are heterogeneous, has proved to be challenging. Even with significant efforts to plan, model and integrate new and existing systems into a functional system-of-systems, end users often encounter unforeseen and undesired emergent behavior on the target infrastructure, even at small scales. Examples of unexpected emergent behaviors include unwanted synchronization of distributed processes, deadlock and starvation, and race conditions in large-scale integrations or deployments.
- Deadlock and starvation are not only limited to large-scale systems; they can occur when connecting a few computers or computer systems. For instance, the Ethernet Capture Effect can occur on networks as small as two computers. Problems like these during small integrations or technology upgrades pose many challenges when integrating large-scale systems consisting of thousands of computers, processing elements, and software services.
- In an ideal world, all technological upgrades and new protocols would be tested on the actual target infrastructure at full scale and speeds. However, developers and system integrators are often limited to testing on smaller-scale testbeds. As a result, technologies and methodologies that support representative at-scale experiments on target infrastructure or a faithful simulation are necessary. These should allow application developers to incorporate their application or infrastructure software into the simulator unmodified, so they run precisely as expected on the target system. This would involve processor time and disk simulation as well as network simulation.
- Time dilation, a set of phenomena that describes how two observers moving relative to each other will observe each other as having erroneous clocks, even if the clocks are of identical construction, is used in simulation to describe the process of altering time sources, clocks, and disk and network transactions to allow accurate simulation of multiple Virtual Machines (VM) on a single host. The term is used here since a simulator and the actual operating system running the simulator will see time flowing at different rates due to context switching and other modern operating system techniques, despite both using similar clock mechanisms. The time dilation mechanism in the simulation context attempts to correct this clock drift from the simulator and operating system perspective to allow for a closer approximation of target behavior by simulated tests. Using simulations based on time dilation, system integrators and planners can run unmodified executables, services, and processing elements to accurately emulate CPU, network, disk, and other resources for large-scale systems in much smaller testbeds.

7.1.1. Background

- Validating large-scale distributed systems can be a challenging task. Although simulators exist that can emulate networks consisting of millions of nodes, simulation technologies that are capable of scaling to this level often fail to accurately reproduce the performance and characteristics of the actual target large-scale systems. This makes it difficult to fully validate such systems, as the simulation results may not reflect the behavior of the real-world system.

7.1.2. Formal Composition Techniques

- Formal composition techniques are widely used in the development of mission and safety-critical distributed systems to ensure that software is validated before it is deployed. This involves modelling a target system in a manner that guarantees the distributed system is validated based on validated components and will execute properly on the target system. Techniques such as Step-Wise Refinement, Causal Semantics, Behavioral Modelling, and Object Modelling Technique are employed to achieve this goal.
- However, formal composition techniques can be time-consuming and domain-specific, requiring developers to model all system components to validate the target system. Furthermore, it is difficult to ensure a composition of heterogeneous components that interoperate correctly when constructing systems-of-systems with formal composition techniques. This remains a major challenge for large-scale distributed systems.
- To address this challenge, Domain-Specific Modelling Languages (DSMLs) have been developed. These languages allow developers to tailor a visual modelling language to a specific knowledge domain, enabling the creation of applications, device drivers, and other software/hardware artefacts. However, making a DSML that encompasses all hardware, device drivers, and operating systems for an Internet-connected application or even a small network of application processes remains difficult. The variety of personal computer architectures and configurations must be expressed to ensure proper validation, and threading remains a challenge, even in homogenous hardware, since composing multiple threads has no formally semantic definition that fits all compositions.

7.1.3. Simulation Techniques

- Simulation is a widely used to validate networked and distributed applications by reproducing the conditions of a target platform. Discrete event simulation is a popular simulation model that treats business application logic and operating system logic as separate, distinct events processed by a simulator engine. However, simulators often make approximations that bring testing closer to a target system but may not precisely match the actual system being simulated. This is particularly problematic in highly connected distributed or networked applications that frequently encounter Internet connections with high failure rates or resends.
- To address this challenge, many network emulators attempt to emulate Internet access times, intermittency, network congestion, and local area network testing. For example, Emulab and its derivatives Netlab can simulate dozens to hundreds of processing elements and their interconnections, and allow the swapping in of operating system images, applications, and test script setup to enable automated testing. These simulators also provide robust network emulation, including bandwidth restriction and packet loss. However, the accuracy of the simulation is left to the developer or user and how they configure operating system images, scripts, etc. Additionally, some simulators do not explicitly support multiple virtual machines per host, so operating system-specific VM managers may be used to scale a small testbed to a larger target system. It's important to note that the throughput difference between emulation and real-world performance can differ by as much as 20%, and closer performance is possible if more than just networking is emulated via time dilation.

7.1.4. Motivating Scenarios

- To ensure the successful deployment of large-scale computer systems, it is critical to perform accurate simulations of these systems in a smaller and less critical infrastructure before their deployment. This pre-deployment testing helps identify and analyze typical functional and performance problems and resolve them before they occur in the production environment. Here are three scenarios that can happen:
    - Ethernet capture effect: The Ethernet Capture Effect, as shown in Figure 7.1, is a type of misconduct that caused inequity in a networking system, which became prevalent after Ethernet hardware was capable of supporting optimal speeds indicated in the Ethernet protocol standard. Although it is particular to Ethernet, it can occur in any other system with a shared bus. To reproduce this behavior, two hosts that frequently send information are required. When a collision arises between Ethernet-connected hosts, a back-off time is randomly generated by all hosts involved in the collision to select a winner to send information through the Ethernet. The winner of this contest has its back-off timer reset, while the losers accumulate back-off timers until they successfully send information. The problem occurs when the winner of these contests has a significant amount of data to transmit, and the Ethernet hardware is quick enough to allow the winner to send its next data immediately. As a result of its timer being reset, this host or service would win each contest indefinitely, while the losing hosts would be essentially starved until the winning process or service is completed. If the winning process never stops, the starvation would be continuous until the last nodes involved are reset. To avoid such problems during production deployment, the issue should be identified and resolved during simulation on a testbed. If a simulator cannot identify such behaviors, significant problems can arise in production deployments.

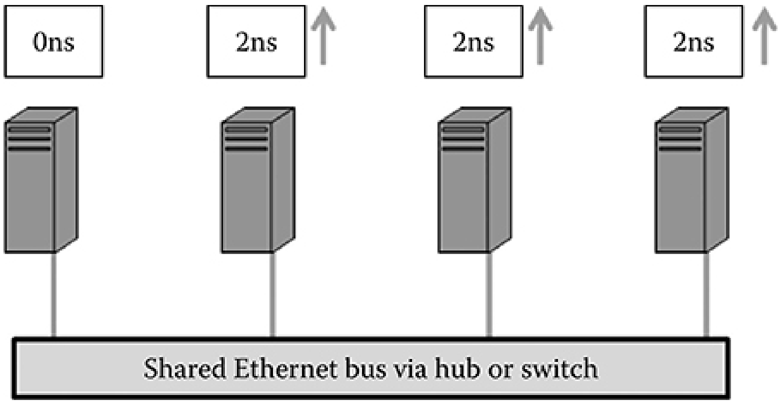


Figure 7.1. Ethernet capture effect in a four-host system

    - Application services: Numerous companies, such as Amazon and Google, operate thousands of servers to provide software services to millions of users. Application Services can vary in size, ranging from a few to several thousand hosts. While most service providers utilize proprietary networks and systems, some open-source auction, e-commerce, and specialty sites are available for general testing. However, many legacy and proprietary systems are closed-source, making it difficult to employ formal composition methods for most of the utilized systems. As a result, simulators can be utilized to assess and validate a target Application Service. However, simulation technologies may require significant modifications to be accurate enough for validation before deployment.

- Large-Scale Systems: Large-scale systems, consisting of multiple subsystems and evolving over time, can be highly complex. An example of this complexity is the integration of a dozen or more Application Services into a single large-scale system comprised of 2000 nodes with heterogeneous hardware and services. These subsystems are linked through a combination of Internet connections and local area networks. The time dilation method can be used to simulate medium to large networks with precision, maintaining accuracy and scale. However, simulations of phenomena such as the Ethernet Capture Effect and race conditions may require further modifications to the time dilation simulation process.

7.1.5. Applying Time Dilation

- The Time Dilation Factor (TDF) is a phenomenon rooted in the theory of relativity, whereby two observers may view each other as having inaccurate clocks, even if both are of equivalent scale and construction. In the context of VM emulation, time dilation refers to the process of dividing real time by the scale of the target system that will be emulated on a particular host. Each host machine potentially emulates multiple other hosts through VMs in an environment like Xen, requiring the same hardware, resources, and timing mechanisms as those found on the host machine. However, this approach does not accurately emulate the timing of the physical target system, as the testing system will be sharing each physical second between the emulated hosts or services, resulting in each VM believing a full second of computational time has passed when the VM was only able to run for a fraction of that time. This sharing can cause problems in emulation, such as affecting throughput to timer firings and sleep statements. Time dilation allows the system developer to adjust the passage of time to more accurately reflect the actual computational time available to each VM. Figures 7.2 and 7.3 illustrate the time dilation effect on the simulation of nine VMs on a single host and the time incrementation when using this concept, respectively.
- The Scale Factor (SF) is a parameter that specifically denotes the number of hosts being emulated, while the Time Dilation Factor (TDF) is used to scale time. Although these factors can be set to different values, in experimental contexts, both TDF and SF are typically set to the same values.

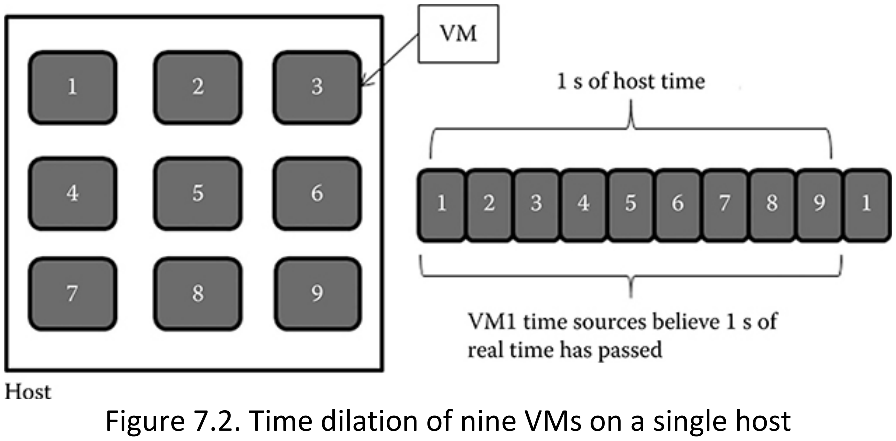


Figure 7.2. Time dilation of nine VMs on a single host

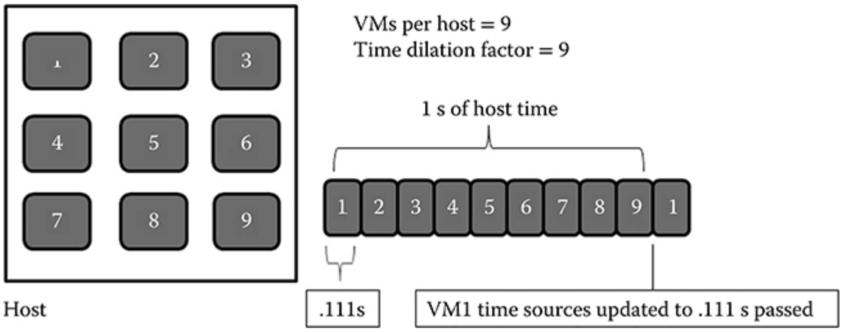


Figure 7.3. When running nine virtual machines on a single host, timers for each individual increment by the amount of processor time used by the VM

7.1.6. Para virtualized vs. Fully Virtualized VMs

- In order to emulate disk I/O, it is necessary to understand the complexities of two distinct types of virtual machines: Para-virtualized and fully virtualized.
    - A Para-virtualized virtual machine involves an OS image that has been partially emulated on the host, thus imposing some limitations.
    - On the other hand, a fully virtualized virtual machine requires hardware support, such as Intel Virtualization Technology or AMD Secure VM, but enables emulation of any operating system image directly on the hardware, regardless of the type of supported OS. Figure 7.4 illustrates the disk throughput both with and without time dilation.

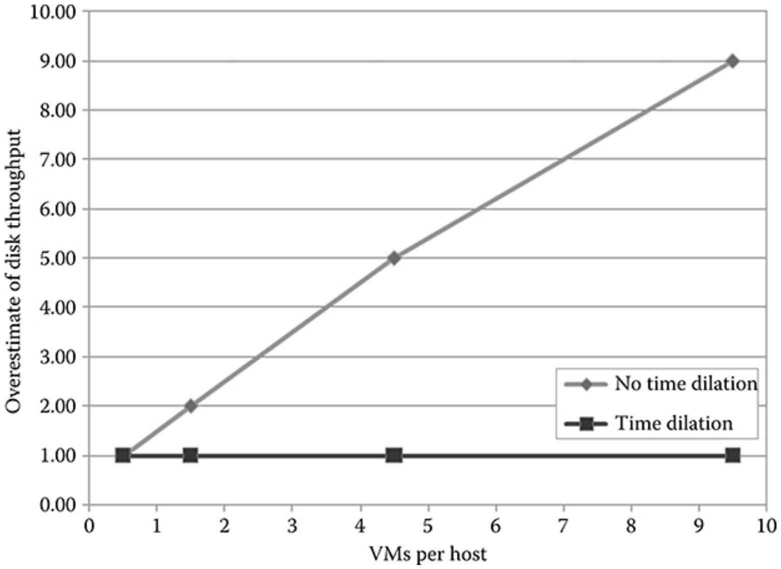


Figure 7.4. Overestimation of disk throughput without time dilation versus using CPU and disk time dilation scaling.

7.1.7. CPU Scheduling

- In order to implement CPU scheduling that is properly scaled, it is necessary to intercept and appropriately scale a variety of time sources. This involves intercepting and scaling timer interrupts, such as the Programmable Interrupt Timer, as well as specialized counters like the TSC on Intel platforms, and external time sources like the Network Time Protocol, before passing them on to virtual machines. However, timing is more complex than simply allocating 1/(time dilation factor) time to each virtual machine. For virtual machines that are input/output bound, they may not utilize their full CPU allocation, which can cause the CPU usage of non-IO-bound virtual machines to be skewed upwards and can have an impact on all aspects of the emulation that are dependent on timing.

7.1.8. Network Emulation

- Network emulation is implemented by capturing all network traffic and routing it through a network emulator. Time dilation plays a crucial role in emulating network throughput and latency. For instance, emulating a 1 Gbps network on a scaled host with a time dilation factor of 10 involves shipping 100 Mbps (1/10 of the total) to the host within a second. Latencies are also easily mimicked by slowing down each virtual machine (VM) to 1/10 speed. Consequently, a system requiring 100 µs latency on the target system can be emulated with data arriving every 1 ms.
- Time dilation is a powerful and robust mechanism for network emulation. It can be used to emulate network throughput and speeds that exceed the network capacity available over a link. Testers can simulate ultra-high-speed capacities and scale the number of VMs per host while simultaneously scaling the network capacity between hosts if required. The key to this ability is slowing down each VM according to a TDF. If all VMs operate at that time scale, the network can be emulated at a factor equal to the TDF. Thus, time dilation enables the effective emulation of networks, allowing testers to perform various tests and simulations on their systems.

7.1.9. Memory Emulation

- One challenge in virtualization is the emulation of physical memory required by the user-specified number of VMs per host. Due to the scarcity of physical memory, VM managers often resort to emulating physical memory using virtual memory on the host. However, virtual memory is usually allocated from a hard drive, which has significantly slower fetch times than physical memory. As a result, VMs may still function but with major timing differences between the testbed system and the target system, leading to race conditions, deadlock, and other types of emergent misbehavior during testing, especially when the physical memory is not enough to properly mimic target system performance.
- To resolve this issue, adding more physical memory to more hosts is the easiest solution. However, if this is not feasible, memory emulation is the only option. Memory emulation involves completely emulating the instruction set for all VMs on the host and running most of the VMs on virtual memory with a TDF that reflects usage of virtual memory instead of physical memory. This solution, however, results in a significant increase in the time required to run an experiment. Figure 7.5 illustrates the difference between accessing physical memory and hard drive data for memory needs, which is typically six orders of magnitude. Emulating all VMs in virtual memory and adjusting the TDF accordingly to the access time difference could result in a time dilation simulation taking over 1 million times longer with emulation on hard disks and over 10 thousand times longer with emulation on a flash memory type drive. In conclusion, virtualization presents challenges when emulating

physical memory required by the user-specified number of VMs per host. While memory emulation is the only option when adding more physical memory is not feasible, it comes with a significant increase in the time required to run an experiment. Therefore, it is crucial to carefully consider the number of VMs per host and the available physical memory before resorting to memory emulation.

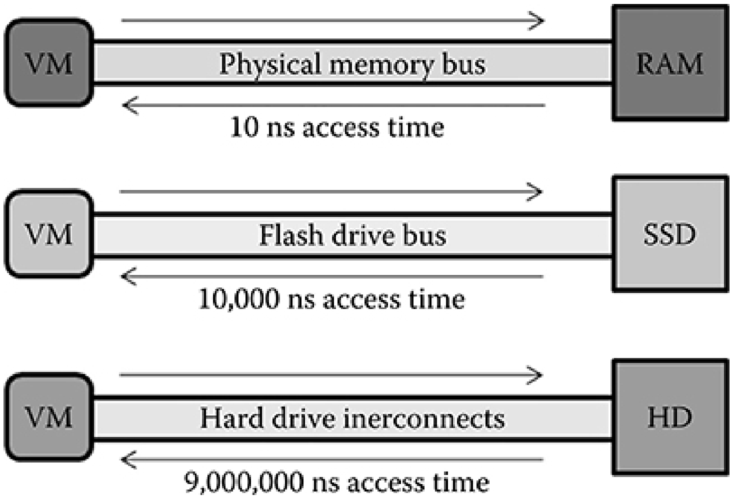


Figure 7.5. Memory access time comparison between physical memory and traditional hard drives

Exercises & Case Scenarios

Exercise 1: Hands on Training on CPS Real-time Co-simulation Tool (EXataCPS-HYPERSIM) – A guide for getting started with RT simulator EXataCPS and HYPERSIM OPAL RT

Goals	To learn about: Real time co-simulation SCALABLE EXataCPS and HYPERSIM OPAL RT
Pre-reading material	L. Zhang, S. Li, L. Wihl, M. Kazemtabrizi, S. Qaseem, J. Paquin, S. Labbé,” Cybersecurity Study of Power System Utilizing Advanced CPS Simulation Tools”,
Main material resources	https://wiki.opal-rt.com/display/DOCHS/Getting+Started, IEC 61850 IEC Just Published Just Published, IEC Webstore https://wiki.opal-rt.com/display/DOCHS/HYPERSIM+User+Documentation https://wiki.opal-rt.com/display/DOCHS/EXata+CPS+%7C+Installation+and+License+Request SCALABLE EXataCPS1.1 Documents
Hours assigned	
Assignment criteria	Training session followed by group discussion Extra work of creating complete models
Done by	All partners

Real-time co-simulation is a more advanced simulation technique used to enhance CPS security testing by simulating the behavior of multiple systems or components that interact with one another. In the context of Cyber-Physical Systems (CPS) security, co-simulation can be used to simulate the interactions between the physical (Operational technology (OT)) and cyber systems (information technology (IT)) of a CPS, as well as the interactions between the CPS and its environment. This can be useful for assessing the CPS's security, identifying vulnerabilities, and testing the efficacy of various security measures. Whereas, the CPS simulation's real-time interactions with the external environments will necessitate that the simulator execute/solve all computations and provide output precisely within the simulation time step (ts) i.e. This simulation time step should be synchronized to run in universal time in order to employ all the different hardware in the loop testing scenarios (e.g., C-HIL, P-HIL,…) as illustrated in Fig (1), [1-2].

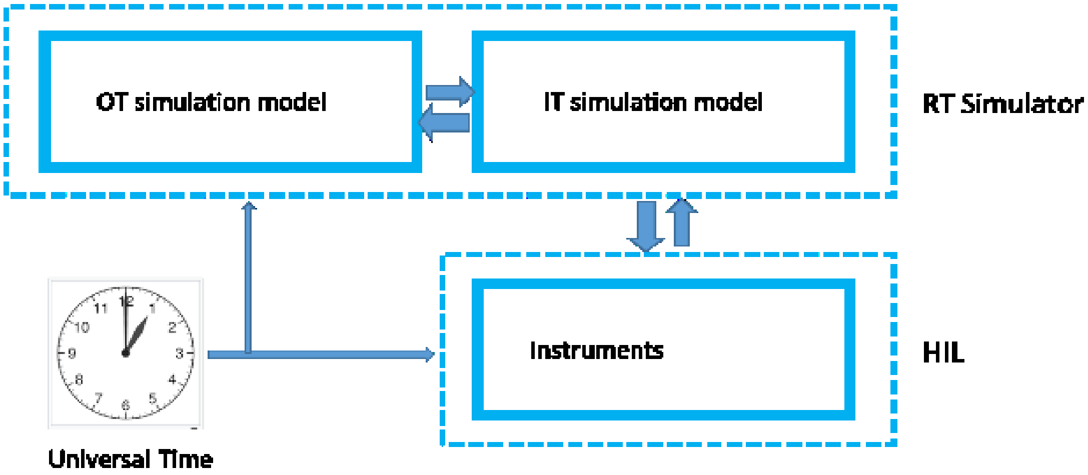


Figure 1: CPS RT co-simulation synchronized with the Universal time

EXataCPS and OPAL-RT Principles

SCALABLE EXataCPS:

Scalable Network Technologies is the leading provider of live, virtual, constructive communications networking modelling and simulation tools across all domains (undersea-to-space). Where EXataCPS is one of these Network Technologies offered by Scalable (now acquired by KEYSIGHT) with a high-fidelity network simulation/emulation tool used to test the performance and scalability of the communication networks. It allows users to simulate different network conditions and test the behaviour of various network protocols, as well as enables to identify and address potential issues before they affect real-world network performance based on various operational scenarios "what if", including cyber-attacks. The simulation/emulation run in real-time and models' connections, computers, protocols, firewalls and other defences. New advance Scalable EXataCPS tool is integrated with OPAL-RT within the HYPERSIM software GUI ruining in one physical target simulators. This advance new solution allows both software to share the available resources e.g., CPU cores, virtual and physical I/O etc., in order to reduce cost, simplicity, smoothly-internally exchanging real-time data via virtual adapters as well as more effectively identify and visualize the impact of cyber-attacks on OT systems. [3]

OPAL-RT

Opal RT Technologies is a pioneer in the development of real-time digital simulation software and engineering services for power system and electric drive research, testing, and validation, as well as high-performance, real-time digital simulators. Their solutions are used in a wide range of applications, such as power system stability and control, renewable energy integration, and electric vehicle charging and grid integration. The simulator hardware and the associated software platform HYPERSIM comprise the overall setup. The simulator's operating system is a proprietary version of Linux known as OPAL-RT Linux [4].

OPAL RT and SCALABLE are working together and have integrated EXataCPS and HYPERSIM within the same simulator as illustrated in Figure (2). Both software can execute on the simulator operating system and distribute processes on multiple CPUs to achieve parallel computation. The connection between the two software is virtual Ethernet links, but the data are still transmitted as standard communication protocols. This solution makes it possible for the communication emulator and the physical system simulator together executed on the same machine that hosted and controlled by individual host machine.

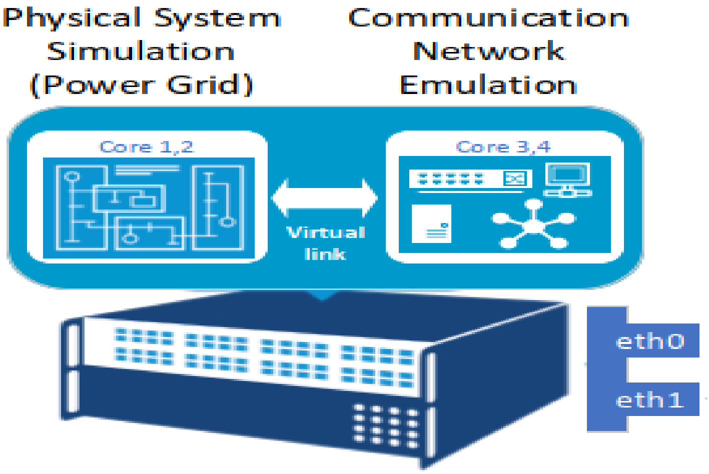

Physical System Simulation (Power Grid)
Communication Network Emulation
Core 1,2
Core 3,4
Virtual link
eth0
eth1

Figure 2: OPAL RT HYPERSIM and SCALABLE EXataCPS integrated at one RT simulator

Design Principles

HYPERSIM is a standalone software that been used for simulating three-phase electro-magnetic and electro-mechanical transients. User is able to design-create simulated models by using the rich component library of more than 300 validated power system components and controllers. Users also can import ready models directly from MATLAB Simulink [4]. In order to simulate/test physical cyber security application for power systems SCALABLE EXataCPS and HYPESIM need to be installed at the Host machine usually a Windows PC or virtual machine, as illustrated in Figure (3).

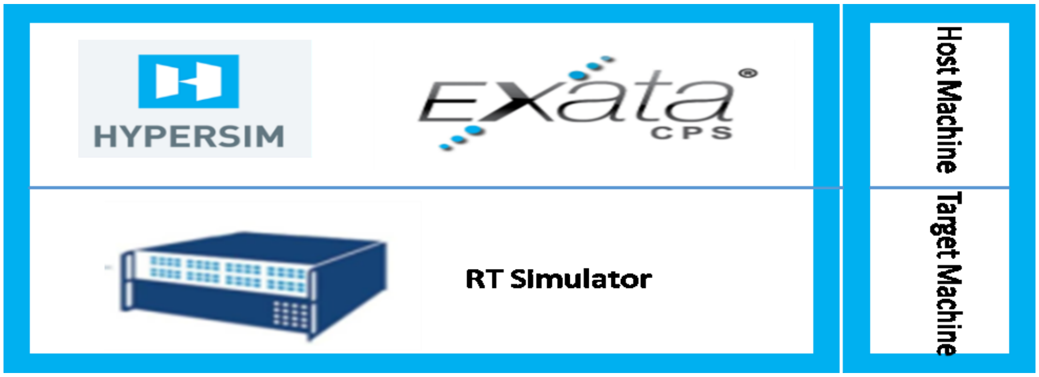


Figure 3: HYPESIM and EXataCPS GUIs

From Figure (3) the primary power system model (OT) is developed in HYPERSIM environment while the communication system (IT) model is developed in EXataCPS. Therefore, in order to design a complete model, the following steps can serve as a starting point:

1- Define the scope and objectives of the simulation, clearly define the problem that the simulation is intended to solve
2- Identify and list all the IT/OT systems components
3- Clearly define the specifications, characteristics, and operating conditions of both systems components defined in step 2.
4- Design and create the model using the components defined in step 2 along with their operation conditions specified in step 3 in order to accurately construct overall systems.
5- Set the simulation parameters, including the starting time, time step, simulation length, solver etc.,
6- Analyse, compile and execute the simulation, then using the visualization and analysis tools to help interpret the simulation results
7- Validate and verify the accuracy of the model by comparing its results to real-world measurements or other data

The process flow for predefined real-time simulation creating model steps using EXataCPS and HYPERSIM is illustrated in Figure (4).

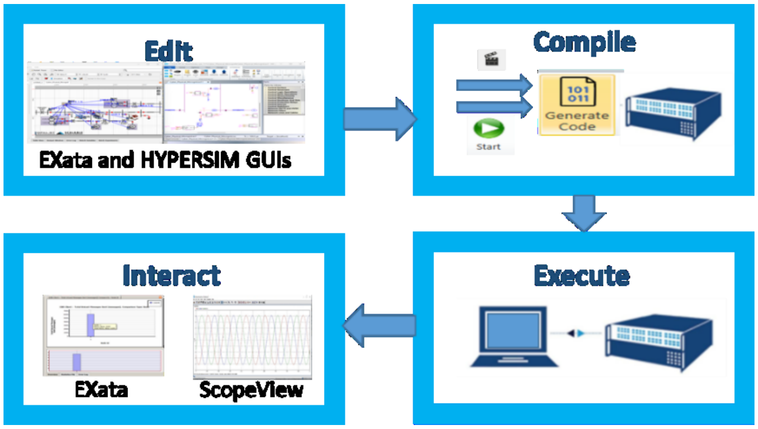


Figure 4: Process flow for creating model steps using EXataCPS and HYPERSIM

Conducting RT CPS model

Creating a model within the OPAL RT CPS environment requires to gather knowledge of both HYPERSIM to model power system (OT) and EXataCPS to model the communication system (IT) as well as some specific communication protocols that wildly used to connect the electrical units [3] e.g., IEC 61850 [5]. This combination of multidisciplinary different systems will make the task more challenging. For simplicity and for the first starting step is to start with pure HYPERSIM and pure EXataCPS models is recommended. This first step will help the user to familiarize himself with the software characteristics, and operating conditions. There is two option to start with, either using the available demo examples within the software provided library or by creating model from scratch. Both options need to be finalized by interacting with the running model using different analyzation and visualization tools. At HYPERSIM GUI for example is integrated a "Scopeview", which is analysing and visualizing tool. The "Scopeview" role is to gather all the measurements from different sensors that initiated within the designed model and analysed/visualized these measurements within a sub-window as illustrated in Figure (5).

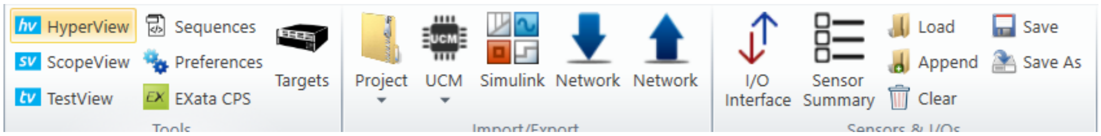


Figure 5: HYPERSIM GUI and ScopeView tool

While at EXataCPS GUI, within the components toolbar there is four components, one from them is "Analyzer" component, which is used for analysing and visualizing the simulation results.

Figure 6: EXataCPS GUI and "Analyzer" component

Hands-on with EXataCPS and HYPERSIM

This section explains step by step how to design, simulate and implement a model. Because our primary goal is to provide cyber security for power system training based real time simulator, authors present here a combined EXataCPS and HYPERSIM software model. Then subsequently the generated codes from both software are individually transferred and run in the real-time OPAL-RT simulator.

Installation

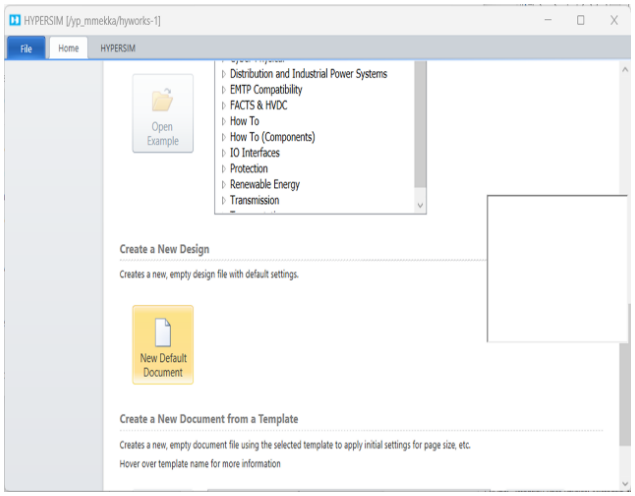
To start with the CPS model, SCALABLE EXataCPS newer version 1.1 or latter and OPAL RT HYPERSIM newer version 2022.2 or latter is required. These two software need to be installed in to the host-machine. OPAL-RT and SCALABLE provides detailed step by step guide for software installations, [6], [7]. Before proceeding further, it is worth mentioning that even EXataCPS software are integrated in to the HYPERSIM GUI, however the EXataCPS software can be run as standalone software and has individual GUI.

Developing a model from scratch

Starting with HYPERSIM at the beginning.

OPAL RT HYPERSIM

At the desktop of the host machine navigate to HYPERSIM icon, right click on it and chose the “Run as administrator” from the selection window, since HYPERSIM needs administrator privileges.



Create a New Design

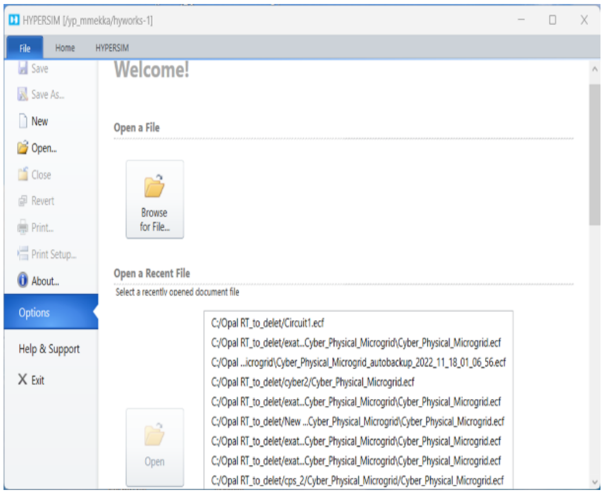
Creates a new, empty design file with default settings.

Create a New Document from a Template

Creates a new, empty document file using the selected template to apply initial settings for page size, etc.

Hover over template name for more information

Figure 7a: option window



Welcome!

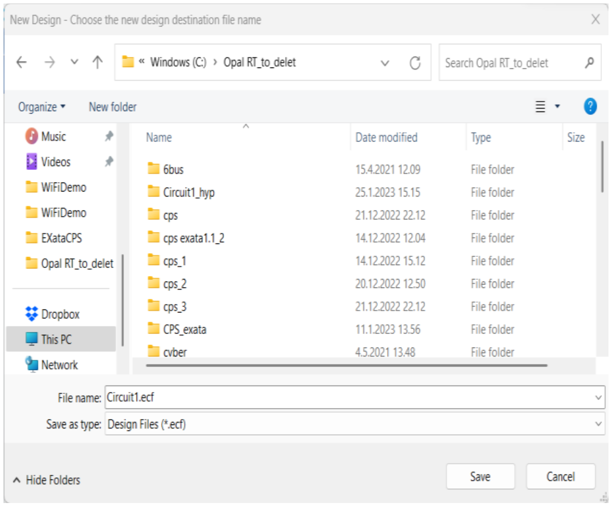
Open a File

Open a Recent File

Select a recently opened document file

C:/Opal RT_to_delet/Circuit1.ecf
C:/Opal RT_to_delet/exat...Cyber_Physical_Microgrid\Cyber_Physical_Microgrid.ecf
C:/Opal ...icrogrid\Cyber_Physical_Microgrid_autobackup_2022_11_18_01_06_56.ecf
C:/Opal RT_to_delet/cyber2/Cyber_Physical_Microgrid.ecf
C:/Opal RT_to_delet/exat...Cyber_Physical_Microgrid\Cyber_Physical_Microgrid.ecf
C:/Opal RT_to_delet/New ...Cyber_Physical_Microgrid\Cyber_Physical_Microgrid.ecf
C:/Opal RT_to_delet/exat...Cyber_Physical_Microgrid\Cyber_Physical_Microgrid.ecf
C:/Opal RT_to_delet/exat...Cyber_Physical_Microgrid\Cyber_Physical_Microgrid.ecf
C:/Opal RT_to_delet/cps_2/Cyber_Physical_Microgrid/Cyber_Physical_Microgrid.ecf

Figure 7b: New Default Document

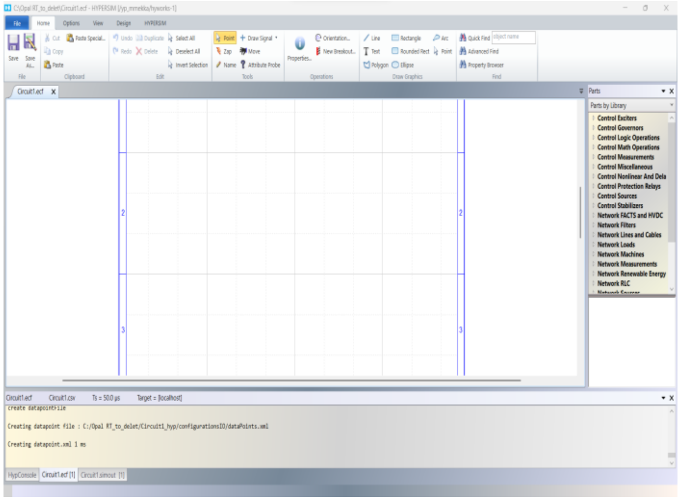


Name	Date modified	Type	Size
6bus	15.4.2021 12.09	File folder	
Circuit1_hyp	25.1.2023 15.15	File folder	
cps	21.12.2022 22.12	File folder	
cps exata1.1_2	14.12.2022 12.04	File folder	
cps_1	14.12.2022 15.12	File folder	
cps_2	20.12.2022 12.50	File folder	
cps_3	21.12.2022 22.12	File folder	
CPS_exata	11.1.2023 13.56	File folder	
cvber	4.5.2021 13.48	File folder	

File name: Circuit1.ecf

Save as type: Design Files (*.ecf)

Figure 7c: Saving option window



Circuit1.ecf Circuit1.csv Ts = 50.0 µs Target = [localhost]

```
create datapointFile

Creating datapoint file : C:/Opal RT_to_delet/Circuit1_hyp/configurationsIO/dataPoints.xml

Creating datapoint.xml 1 ms
```

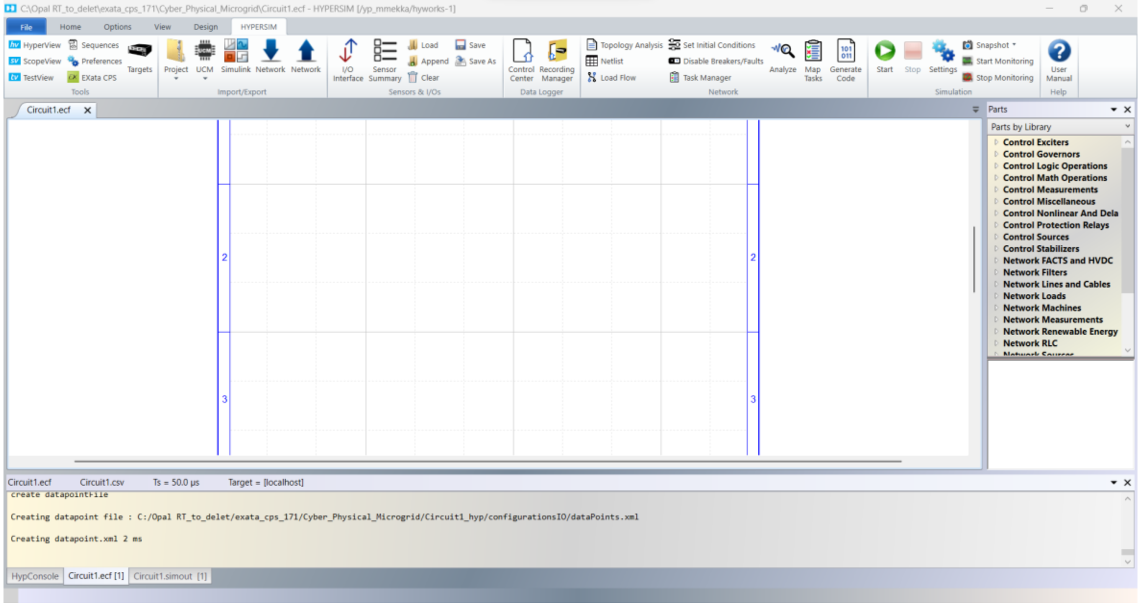
Figure 7d: Full HYPERSIM GUI

- Step 1: the option window will pop up (Figure 7a)
- Step 2: from the option window chose “New Default Document” under “Create New Design” section (Figure 7b)
- Step 3: Saving option window pops up, name new design file and save it to the work space directory within a predefined project folder (Figure 7c)

- Step 4: Full HYPERSIM GUI pops up (Figure 7d)
- Step 5: Create new model as per guidelines provided in OPAL-RT HYPERSIM manual [6]. The examples provided can also be used as a platform to build or develop advance new model.

SCALABLE EXataCPS

To start the EXataCPS on Windows host machine, do one of the following either starting as standalone software by right clicking at the EXataCPS icon located at the desktop of the host machine and chose the "Run as administrator" from the selection window, since EXataCPS needs administrator privileges, or from the HYPERSIM GUI by clicking on the EXataCPS button as illustrated in Figure (8). At this point CPS training second option has been explained here. Since within the first option user need to develop the EXataCPS environment manually by creating name space and assign eno2 (physical adapter used by EXataCPS) for it using command prompt, whereas for the second option HYPERSIM did the task internally.



```
create datapointFile

Creating datapoint file : C:/Opal RT_to_delet/exata_cps_171/Cyber_Physical_Microgrid/Circuit1_hyp/configurationsIO/dataPoints.xml

Creating datapoint.xml 2 ms
```

Figure 8: HYPERSIM GUI with the integrated EXataCPS starting button

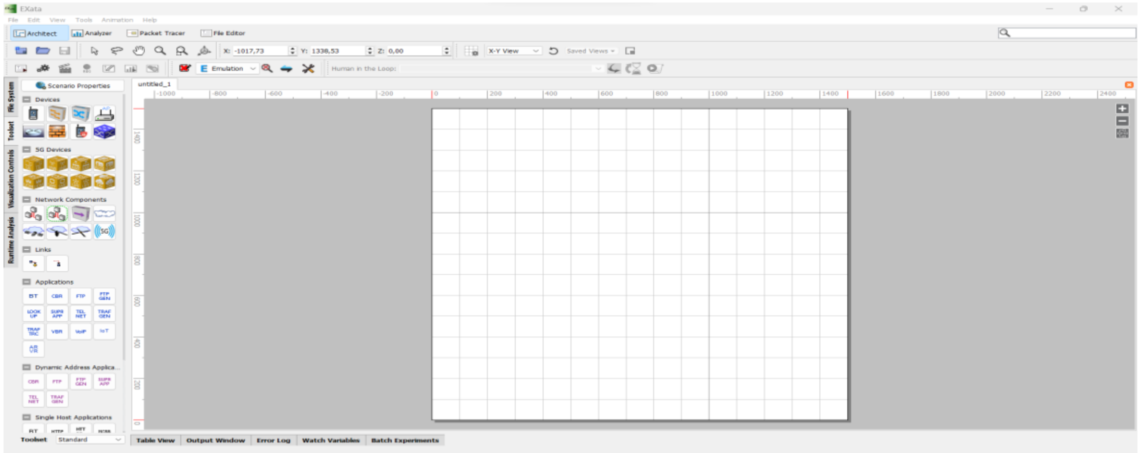
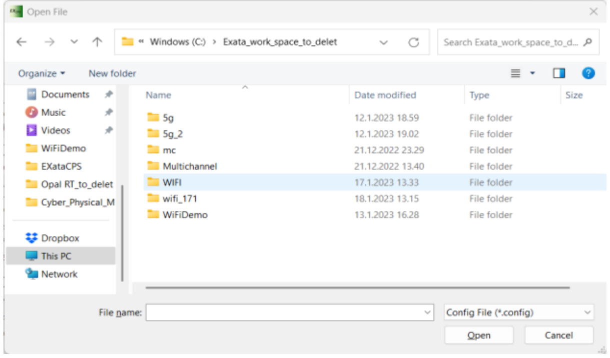


Figure 9: EXataCPS GUI starting by default with new design mode



Name	Date modified	Type	Size
5g	12.1.2023 18.59	File folder	
5g_2	12.1.2023 19.02	File folder	
mc	21.12.2022 23.29	File folder	
Multichannel	21.12.2022 13.40	File folder	
WIFI	17.1.2023 13.33	File folder	
wifi_171	18.1.2023 13.15	File folder	
WiFiDemo	13.1.2023 16.28	File folder	

File name: Config File (*.config)

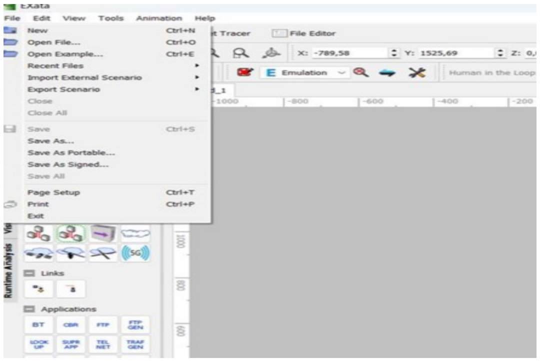


Figure 10a: new design file window

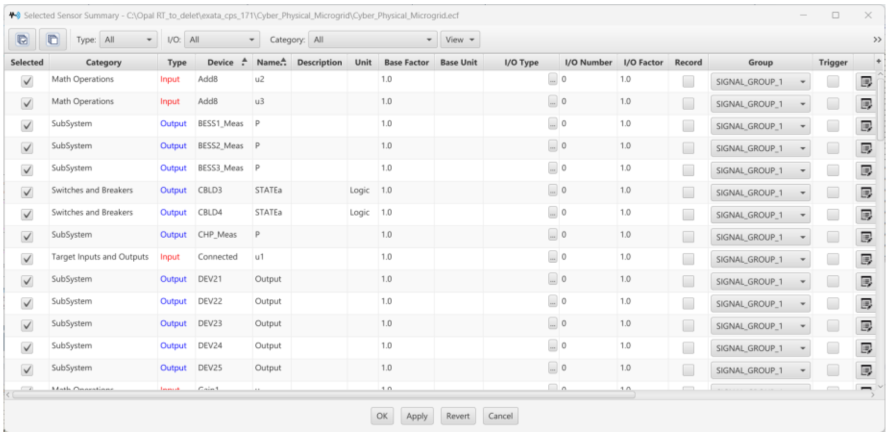
Figure 10b: file Explorer window

- Step 1: EXataCPS GUI window will pops up--> do one from the following (Figure 9)
- Step 2: File -–> New -–> new design file will open (Figure 10a)
- Step 3: File -–> Open file -–> new file Explorer window will pop up to navigate to existing design file (Figure 10b)
- Step 4: File -–> Open Example -–> a new file Explorer window will open, and the user should navigate to the SCALABLE examples folder. Other options are available under the File toolbar menu, which the user can explore.
- Step 5: after user open one from the above, at the open canvas user may create a model accordance with the guidelines provided in the SCALABLE EXataCPS "Product Tour" documents. The examples provided can also be used as a platform for developing a new model.

Executing a model in Real-time

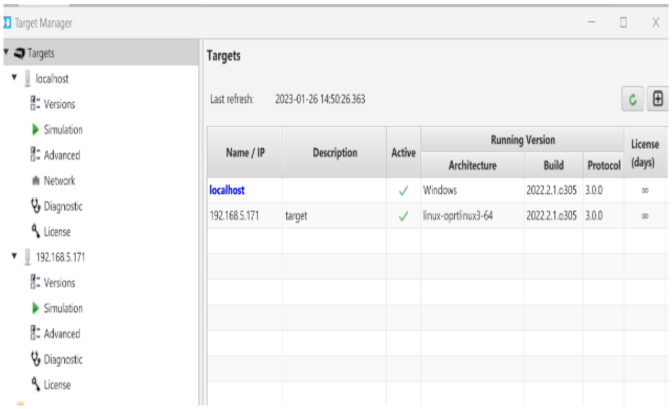
OPAL RT HYPERSIM

After building the model in HYPERSIM, show the selected/assigned sensors by clicking at "Sensor Summary" located at the HYPERSIM ribbon "Sensor and IOs" section. It is worth mentioning here that sensors play two roles: first, they allow monitoring of the various signals across the model by transferring the selected measurements to "ScopeView" in order to visualize them. Second, they can be used to configure analog and digital I/O as well as communication protocols as illustrated at Figure (11).


Selected Sensor Summary - C:\Opal RT_to_delet\exata_cps_171\Cyber_Physical_Microgrid\Cyber_Physical_Microgrid.ecf
Type: All
I/O: All
Category: All
View
OK
Apply
Revert
Cancel

Selected	Category	Type	Device	Name	Description	Unit	Base Factor	Base Unit	I/O Type	I/O Number	I/O Factor	Record	Group	Trigger
☑	Math Operations	Input	Add8	u2			1.0			0	1.0	☐	SIGNAL_GROUP_1	☐
☑	Math Operations	Input	Add8	u3			1.0			0	1.0	☐	SIGNAL_GROUP_1	☐
☑	SubSystem	Output	BESS1_Meas	P			1.0			0	1.0	☐	SIGNAL_GROUP_1	☐
☑	SubSystem	Output	BESS2_Meas	P			1.0			0	1.0	☐	SIGNAL_GROUP_1	☐
☑	SubSystem	Output	BESS3_Meas	P			1.0			0	1.0	☐	SIGNAL_GROUP_1	☐
☑	Switches and Breakers	Output	CBLD3	STATEa		Logic	1.0			0	1.0	☐	SIGNAL_GROUP_1	☐
☑	Switches and Breakers	Output	CBLD4	STATEa		Logic	1.0			0	1.0	☐	SIGNAL_GROUP_1	☐
☑	SubSystem	Output	CHP_Meas	P			1.0			0	1.0	☐	SIGNAL_GROUP_1	☐
☑	Target Inputs and Outputs	Input	Connected	u1			1.0			0	1.0	☐	SIGNAL_GROUP_1	☐
☑	SubSystem	Output	DEV21	Output			1.0			0	1.0	☐	SIGNAL_GROUP_1	☐
☑	SubSystem	Output	DEV22	Output			1.0			0	1.0	☐	SIGNAL_GROUP_1	☐
☑	SubSystem	Output	DEV23	Output			1.0			0	1.0	☐	SIGNAL_GROUP_1	☐
☑	SubSystem	Output	DEV24	Output			1.0			0	1.0	☐	SIGNAL_GROUP_1	☐
☑	SubSystem	Output	DEV25	Output			1.0			0	1.0	☐	SIGNAL_GROUP_1	☐
	Math Operations	Input	Gain1	u			1.0			0	1.0		[illegible]	

Figure 11: Sensor Summary window

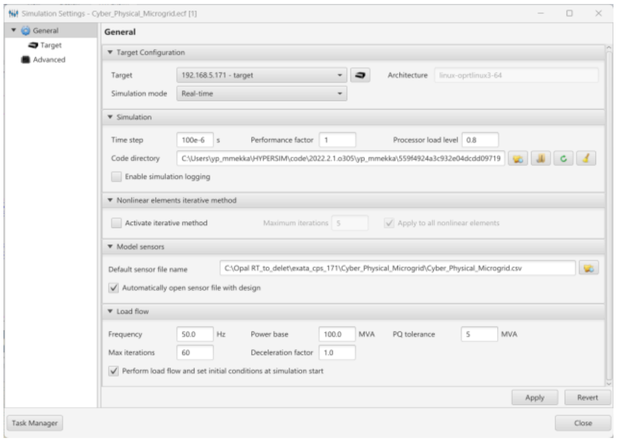


Targets

Last refresh: 2023-01-26 14:50:26.363

Name / IP	Description	Active	Running Version			License (days)
			Architecture	Build	Protocol	
localhost		✓	Windows	2022.2.1.o305	3.0.0	∞
192.168.5.171	target	✓	linux-oprtlinux3-64	2022.2.1.o305	3.0.0	∞

Figure 12a: target Manager window



General

Target Configuration

Target: 192.168.5.171 - target
Architecture: linux-oprtlinux3-64
Simulation mode: Real-time

Simulation

Time step: 100e-6 s
Performance factor: 1
Processor load level: 0.8
Code directory: C:\Users\yp_mmekka\HYPERSIM\code\2022.2.1.o305\yp_mmekka\559f4924a3c932e04dcdd09719
□ Enable simulation logging

Nonlinear elements iterative method

□ Activate iterative method
Maximum iterations: 5
☑ Apply to all nonlinear elements

Model sensors

Default sensor file name: C:\Opal RT_to_delet\exata_cps_171\Cyber_Physical_Microgrid\Cyber_Physical_Microgrid.csv
☑ Automatically open sensor file with design

Load flow

Frequency: 50.0 Hz
Power base: 100.0 MVA
PQ tolerance: 5 MVA
Max iterations: 60
Deceleration factor: 1.0
☑ Perform load flow and set initial conditions at simulation start

Figure 12b: file Explorer window

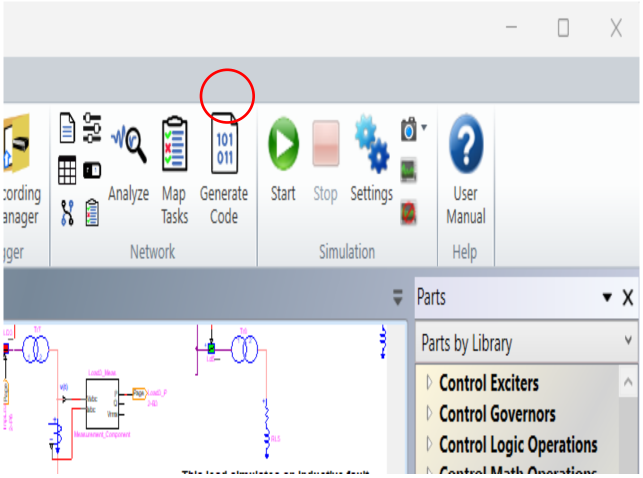


Figure 12c: Simulation Start button

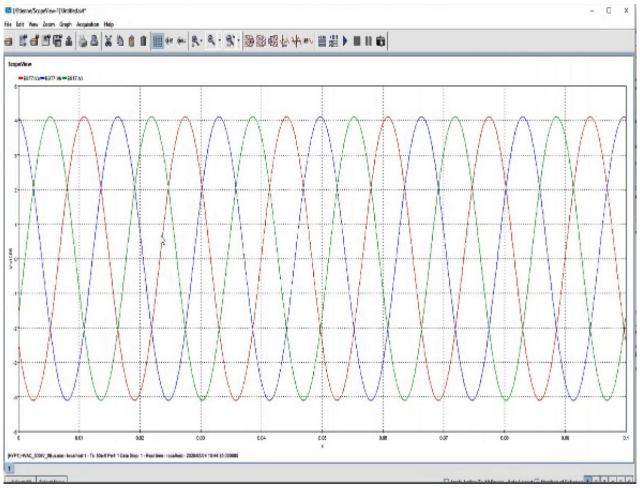


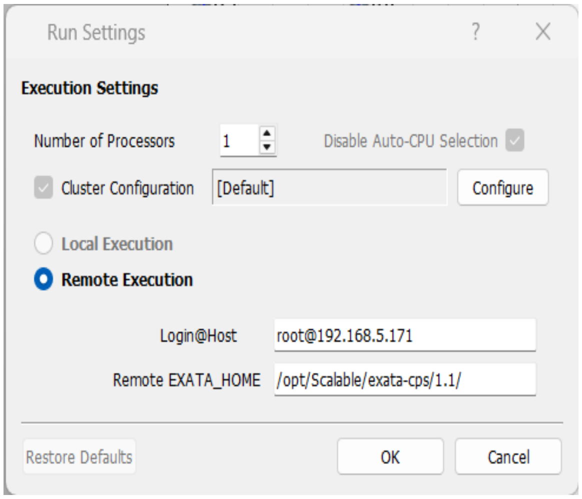
Figure 12d: ScopeView window

Running the model at real-time requires the following steps,

- Step 1: Targets –-> click (+) sign to add target
- Step 2: Add New Target –-> pops up window, define IP address for the target and name, --> ok target will appear at the target Manager window (Figure 12a)
- Step 3: Simulation Settings –-> Target Configuration –-> chose the target, and at the "Simulation mode" chose "Real-time" (Figure 12b)
- Step 3: Start –-> the tasks are mapped automatically to the various cores; the code is compiled and the simulation starts running. (Figure 12c),
- Step 4: ScopeView–-> visualize the simulation results (Figure 12d).

SCALABLE EXataCPS

To run the model in real-time after it has been built in EXataCPS, the model must be transported to the OPAL-RT real-time simulator, where initialization and execution can begin as follows;



Run Settings

Execution Settings

Number of Processors: 1

☑ Disable Auto-CPU Selection

☑ Cluster Configuration: [Default]

○ Local Execution

● Remote Execution

Login@Host: root@192.168.5.171

Remote EXATA_HOME: /opt/Scalable/exata-cps/1.1/

Figure 13a: Run Settings

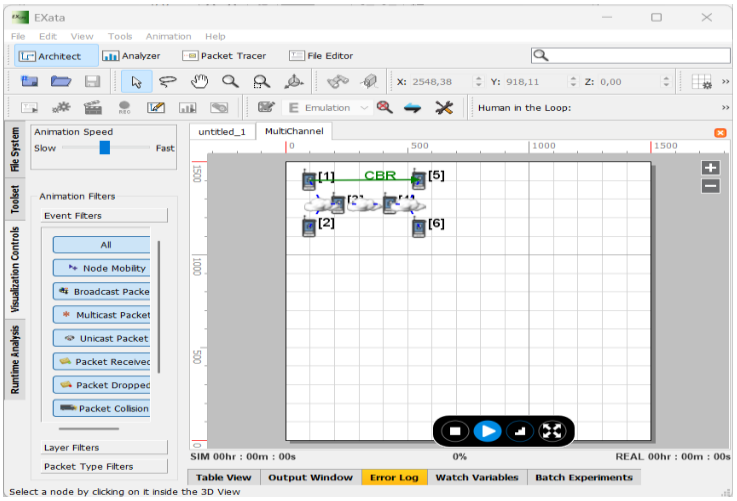


Figure 13b: visualise mode

- Step 1: click Run Settings ––> pops up window of Run Settings, make the admin SSH access to the target and define the directory for the execution folder at the target --> ok (Figure 13a)
- Step 2: select execution mode ––> Target Configuration ––>
- Step 3: click Initialize Simulation ––> pops up window, ask user to save example scenario
- Step 4: A copy of the scenario is saved ––> the design mode will change to visualize mode (Figure 13b)
- Step 5: Click play button ––> to run the scenario in real-time (Figure 14a)
- Step 6: Click he Analyse Results button ––> simulation results in EXataCPS Analyzer (Figure 14b)

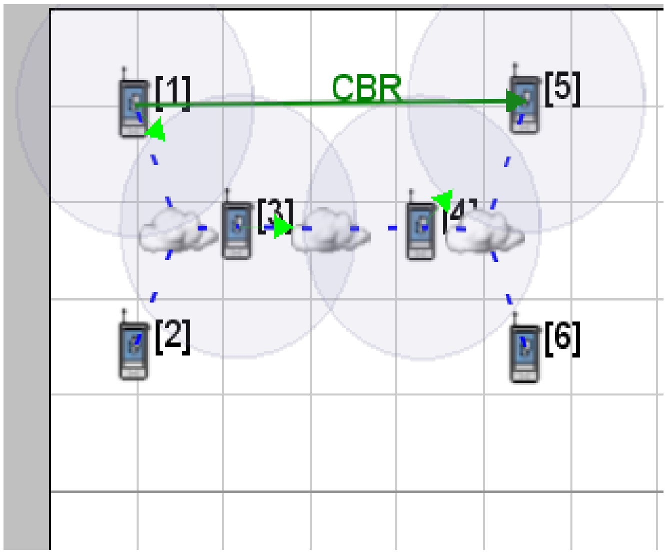


Figure 14a: simulation is running

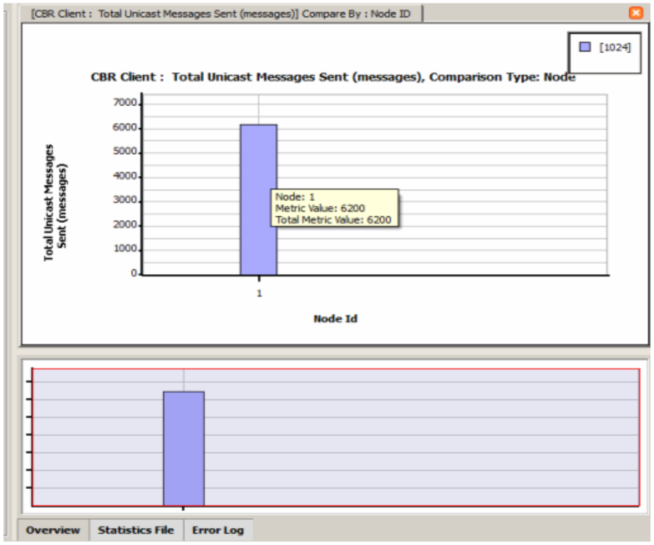


Figure 14b: simulation results

At Figure 14a when a node transmits a packet, green arrows representing successful packet reception whereas the circles representing radio transmissions boundaries.

Exercise 2: Hands on Training on CPS Real-time Co-simulation Tool (EXataCPS-HYPERSIM) – Real-Time Co-Simulations of a Microgrid Active power Managements Against Delay Attack

Goals	To develop CPS testing scenarios that assess offensive aspects of the cyber space Real time co-simulation of MG power managements against delay attack present by SCALABLE EXataCPS and HYPERSIM OPAL RT
Pre-reading material	L. Zhang, S. Li, L. Wihl, M. Kazemtabrizi, S. Qaseem, J. Paquin, S. Labbé,” Cybersecurity Study of Power System Utilizing Advanced CPS Simulation Tools”,
Main material resources	https://wiki.opal-rt.com/display/DOCHS/Getting+Started, IEC 61850 IEC Just Published Just Published, IEC Webstore https://wiki.opal-rt.com/display/DOCHS/HYPERSIM+User+Documentation https://wiki.opal-rt.com/display/DOCHS/EXata+CPS+%7C+Installation+and+Li cense+Request SCALABLE EXataCPS1.1 Documents
Hours assigned	
Assignment criteria	Training session followed by group discussion Extra work of creating complete models
Done by	All partners

Introduction

Real-time co-simulation is a digital model-based test system that can precisely mimic the reaction of multidisciplinary physical systems in real time. This digital replica of the actual systems (digital twins) has been regarded as a valuable tool in Cyber Physical System (CPS) security for power system (smart grid) analysis and assessments, for example, it can help reduce costs and test stresses on the actual systems. As well as assess both offensive and defensive aspects of the cyber space that affect smart grid operation. Furthermore, it enables interaction-based hardware-in-the-loop (HIL) in such a way that external instruments are unaware that the data received is from digital twin models rather than actual physical systems, [1-2].

Development of use-case Principles

To begin with, the OPAL RT and SCALABLE EXataCPS demo CPS security model's use-case study has been used for easy starting. This use-case study presents the offered advanced CPS security solution to simulate and run co-simulation in real-time for both systems (operation technology (OT) and information technology (IT)) at the OPAL-RT real time simulator.

MG Simulation Model at HYPERSIM OPAL-RT

In this demo use-case study, a 25kV MG distribution network is connected to a 120kV sub-transmission system via a 15 MVA delta-wye transformer. Where active power management is being evaluated for the management of MG in both grid-connected and is-landed scenarios [3]. Figure (1) illustrates a single line diagram of a 25kV MG created in HYPERSIM.

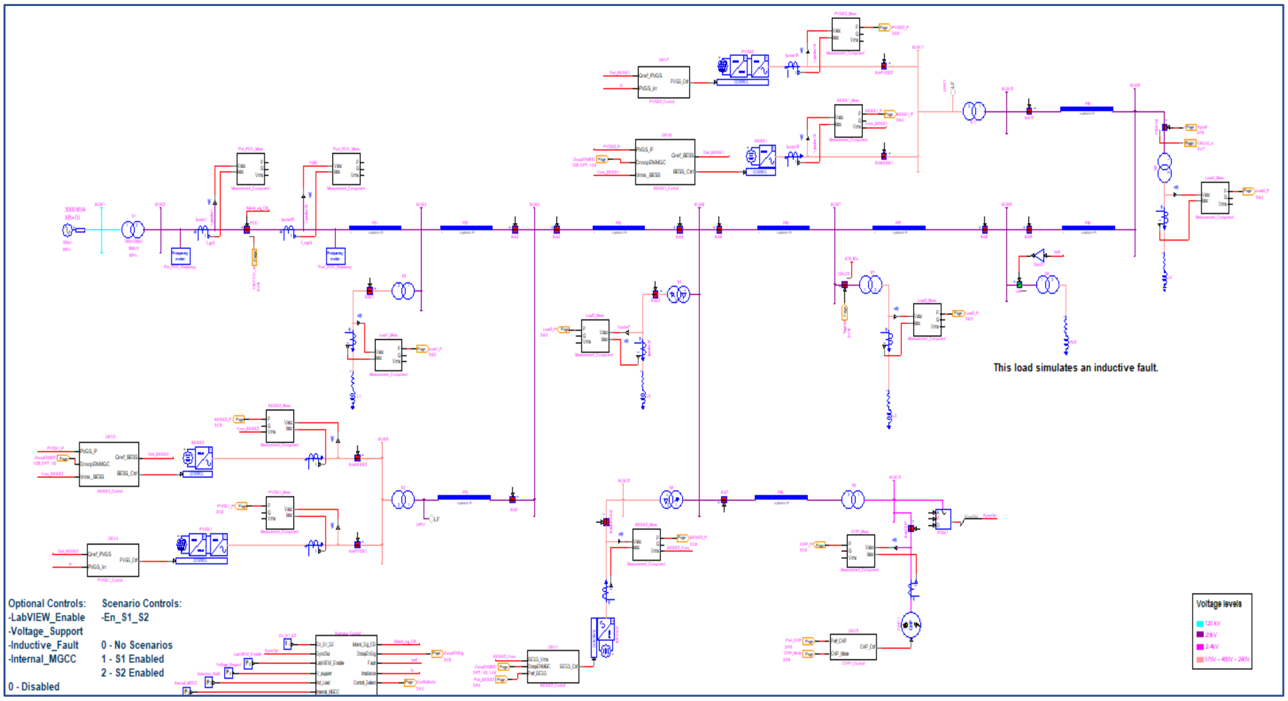


Figure 1: single line diagram of a 25kV MG

Table 1: The MG instruments

Asset	Type	Ratings	Operation Modes
Loads			
Load1	Critical	4MW	Always connected
Load2	Critical	4MW	Always connected
Load3	Hybrid	4MW	Can be disconnected on second priority
Load4	Non-Critical	3MW	To be disconnected in Islanded mode
Distributed Energy Resources (DERs)			
Combined Heat and Power (CHP) plant	Gas Turbine	10MW	P/Q (Grid-connected) V/f (Grid forming)
2 x PV Generation System	With Smoothing Battery Energy Storage System	1.5MVA + 0.5MVA (1.2MWh)	MPPT with smoothing battery
3x Battery Energy Storage System (BESS)	Lead Acid	1MW (5MWh)	Power smoothing

Table 1 lists the MG instruments and their specifications. It includes two 1.5 MVA PV systems, three energy storage systems (ESS) (two 500 kVA ESSs with 1.15 MWh capacity and one 1 MVA ESS with 5 MWh capacity), and one 10 MVA CHP unit. The MG also has four aggregated loads: the first two are critical loads rated at 4 MW, the third is a priority load rated at 4 MW, and the fourth is a non-critical load rated at 5 MW. For simplicity, there is no underlying protection scheme. Each MG unit is linked to a voltage and current measurement subsystem, which is programmed to produce P, Q, and Vrms measurements [4]. These measurements, along with the internal status of the MG unit, are sent to the MGC via IEC 61850 Generic Object-Oriented Substation Event (GOOSE) [5]. The MGC then gathered all MG operation parameters, executed the controlling algorithm, and returned the dispatching signal to MG units via various IEC 61850 interfaces GOOSE.

MG Emulation Model at SCALABLE EXataCPS

The MGC (Node 1) is a simulated and has been implemented on the same model based on software-in-the-loop (SIL). The primary role of the MGC is to receive measurements from the subsystems and use these measurements to send reference set points to some of the MG units. As well as to keep the balance between the generated and consumed power by the distributed energy resources (DERs) and loads respectively. The whole IEC 61850-based communication network is simulated by EXataCPS and presented in Figure (2).

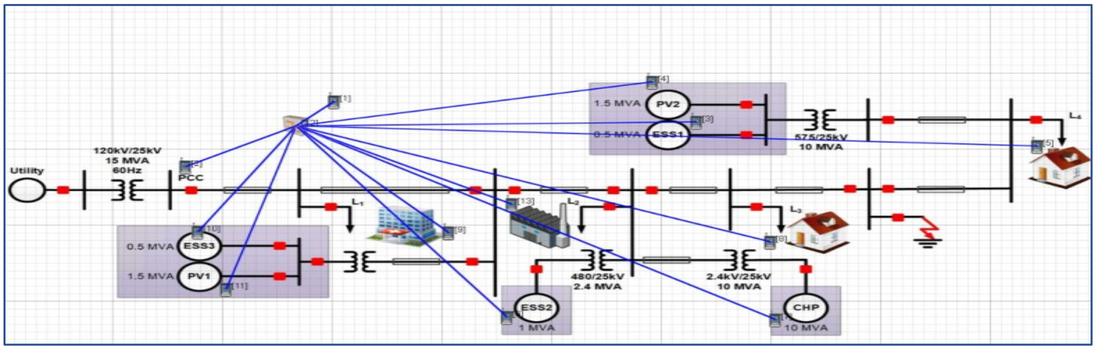


Figure 2: EXataCPS cyber-physical model, blue lines show the communication links between the MGC and different MG units based on GOOSE IEC 61850 interfaces.

Along with the IEC 61850 standard classification the MGC controller is located at the station level, the IEDs are at the bay level, and the physical components are at the process level. The communication network is simulated by EXataCPS using IEC 61850 GOOSE bidirectional interfaces (blue lines) as presented in Table 2.

Table 2: IEC 61850 GOOSE data points

MG Units	GOOSE MGC to Units	GOOSE Unitss to MGC
BESS1	Pref, Qref	Pmeas, Qmeas, Vrms, Status
PV	Pref, Qref, Mode	Pmeas, Qmeas, Vrms, Status
BESS2	Pref, Qref, Mode	Pmeas, Qmeas, Vrms, Status
BESS3	Pref, Qref	Pmeas, Qmeas, Vrms, Status
WIND	Pref, Qref, Mode	Pmeas, Qmeas, Vrms, Status
CHP	Pref, Qref, Mode	Pmeas, Qmeas, Vrms, Status
PCC Breaker	Trip	Pmeas, Qmeas, Vrms, Status
Load 1	n/a	Pmeas, Qmeas, Vrms, Status
Load 2	n/a	Pmeas, Qmeas, Vrms, Status
Load 3	Trip	Pmeas, Qmeas, Vrms, Status
Load 4	Trip	Pmeas, Qmeas, Vrms, Status

Real-time CPS Security Model Conducting at OPAL RT

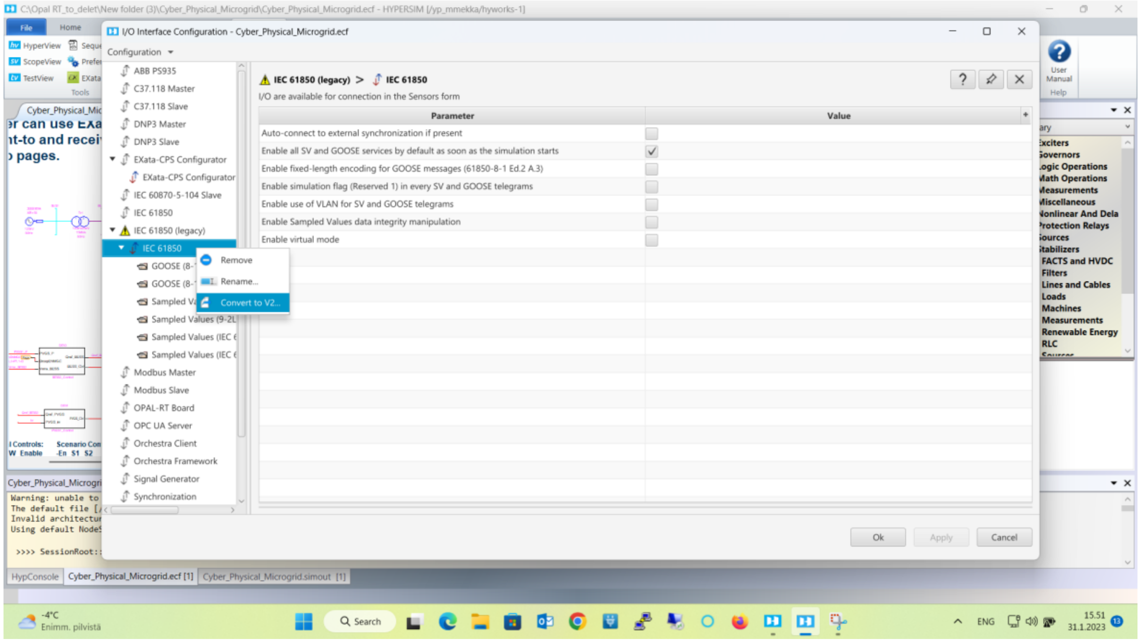
The CPS security simulation model is made up of two models: the HYPERSIM OT model and the EXataCPS IT model, which must be interconnected and exchange data using virtual adapters provided by HYPERSIM. Here is the procedure for implementing and running both models in real-time at the OPAL RT simulator.

HYPERSIM OT model Implementation

Open HYPERSIM software as presented at exercise one, then follow the steps below [6],

- Step 1: from the option window chose “Open Example” from “Open Example File” section

- Step 2: navigate to the “Cyber_Physical_Mirigrid” example within the examples tree --> double click --> choose a folder to copy the demo example
- Step 3: HYPERSIM GUI showing “Cyber_Physical_Mirigrid” demo example pops up (Figure 1). The HYPESIM model consists from two view pages one for the single line diagram and the second for the internal MGC. No changes are required at this point. The model is used exactly as is.
- Step 4: Click at “I/O interface” from the “Sensor and I/Os” section at HYPERSIM toolbar --> choose “IEC 61850” right click --> “Convert to V2” (Figure 3), this step converts IEC 61850 legacy to IEC 61850 version two (V2) standard.
- Step 5: Click at Targets “Target” to check that the target is connected and available
- Step 6: Click at “settings” and change the “Target” filed from “localhost” to online target, and “Simulation mode” filed to “Real-time” -->Apply-->close. Now the simulation is ready to run in real-time from the “Start” button.



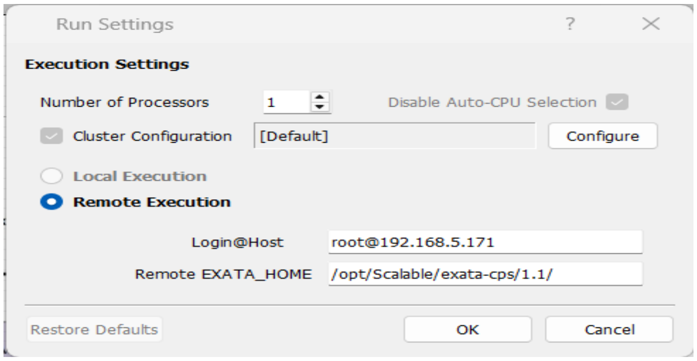
I/O are available for connection in the Sensors form

Parameter	Value
Auto-connect to external synchronization if present	□
Enable all SV and GOOSE services by default as soon as the simulation starts	☑
Enable fixed-length encoding for GOOSE messages (61850-8-1 Ed.2 A.3)	□
Enable simulation flag (Reserved 1) in every SV and GOOSE telegrams	□
Enable use of VLAN for SV and GOOSE telegrams	□
Enable Sampled Values data integrity manipulation	□
Enable virtual mode	□

Figure 3: HYPERSIM I/O interfaces window

EXataCPS IT model Implementation

- Step 1: click on EXata CPS “EXataCPS” located at HYPERSIM GUI toolbar--> Open EXataCPS GUI
- Step 2: click on located at EXataCPS GUI toolbar --> navigate to the folder that the user saves the CPS example (from step 2 above)
- Step 3: at EXataCPS saved folder --> find “Cyber_Physical_Microgrid.config” file --> Open (Figure 2)
- Step 4: Click at located at Run toolbar --> “Run Setting” window opened., check the configuration must be same as (Figure 4)



Run Settings

Execution Settings

Number of Processors: 1 — Disable Auto-CPU Selection ☑

☑ Cluster Configuration: [Default] — Configure

○ Local Execution

● Remote Execution

Login@Host: root@192.168.5.171

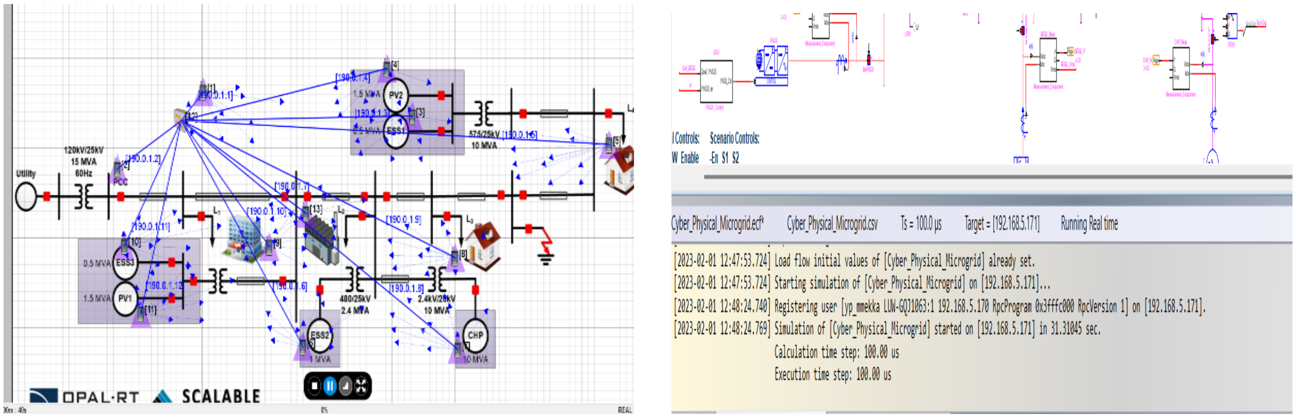
Remote EXATA_HOME: /opt/Scalable/exata-cps/1.1/

Figure 4: EXataCPS Run Setting configuration

Run both software configuration in Real-time at OPAL RT

After both software are ready to run the CPS demo example configuration files in real-time at OPAL RT simulator.

- Step 1: At the beginning user need to run HYPERSIM at first, since HYPERSIM need to create “name space” and assign “eno2” adapter for EXataCPS (Figure 5)
- Step 2: then user need to run EXataCPS as in exercise 1, section 6.2.3



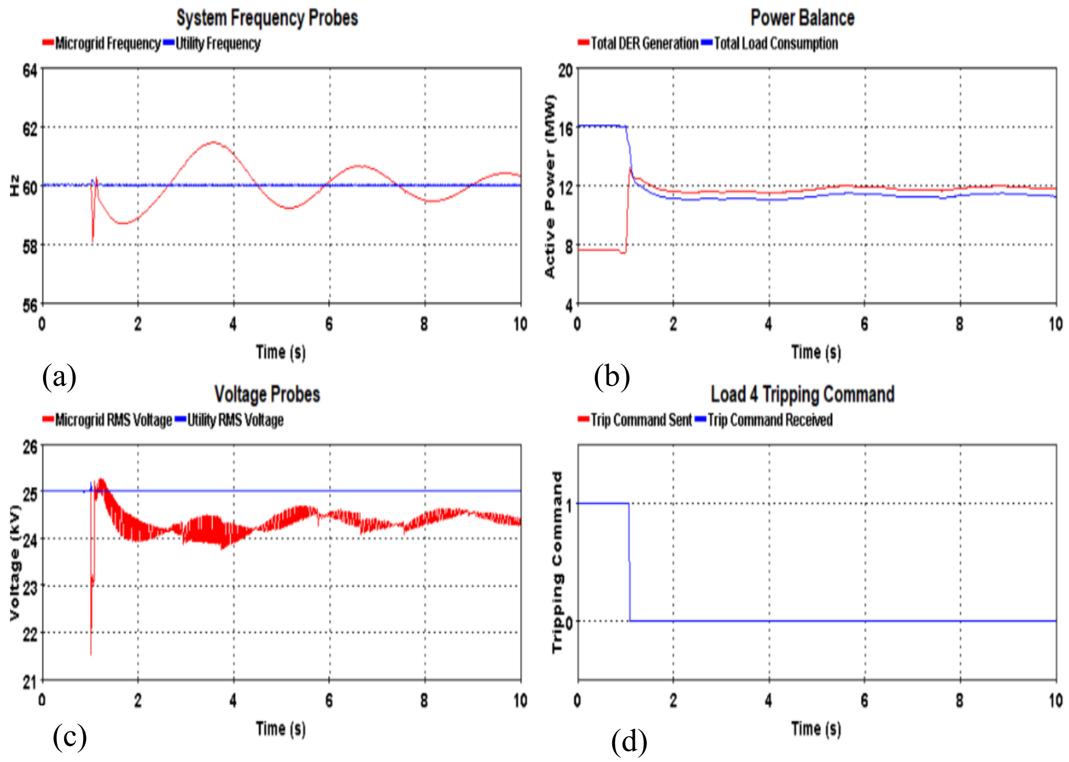
```
Cyber_Physical_Microgrid.ecf*    Cyber_Physical_Microgrid.csv    Ts = 100.0 µs    Target = [192.168.5.171]    Running Real time

[2023-02-01 12:47:53.724] Load flow initial values of [Cyber_Physical_Microgrid] already set.
[2023-02-01 12:47:53.724] Starting simulation of [Cyber_Physical_Microgrid] on [192.168.5.171]...
[2023-02-01 12:48:24.740] Registering user [yp_mmekka LUW-GQJ1063:1 192.168.5.170 RpcProgram 0x3fffc000 RpcVersion 1] on [192.168.5.171].
[2023-02-01 12:48:24.769] Simulation of [Cyber_Physical_Microgrid] started on [192.168.5.171] in 31.31045 sec.
                          Calculation time step: 100.00 us
                          Execution time step: 100.00 us
```

Figure 5: HYPERSIM, EXataCPS output logs show that it runs in real-time successfully at OPAL RT simulator [7]

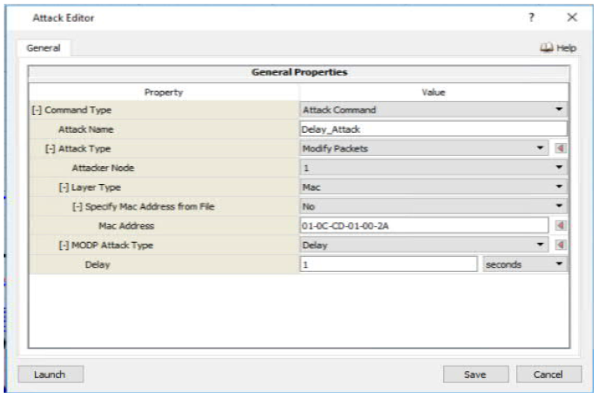
MG Active power Managements Against Delay Attack

The MG enters islanded mode after one second in this demo example. At this point, since the MGC is gathering and monitoring measurements from all MG units. As a result, the MGC will executes the controlling algorithm and check the power balance and implement the power balance operation emergency condition (the difference between power generation and power consumption is not exceeds 3MW). If the check emergency condition becomes true. The MGC attempts to immediately disconnect the sheddable Load 4 in order to maintain MG stability, as shown in Figures (6a-d).



Figures 6: Frequency deviation, frequency transient be maintained within acceptable boundaries Figure (6a). Power Balance show the matching between supply and demand in islanded mode Figure (6b). Microgrid RMS voltage is measured note that the voltage dip during islanding for a very short time, until the load shedding occurs Figure (6c). The sent and received trip commands are overlapping Figure (6d).

Then a cyber-attack (delays) is introduced to the MGC trip signal to disconnect Load 4, since IEC 61850 GOOSE protocol is more sensitive to the time (based on its specification need to reach their destination less than 4 ms). This cyber-attack Delay attack is triggered by using EXataCPS "Attack Editor" targeting node 1 MGC. Figure 7.



General Properties

Property	Value
[-] Command Type	Attack Command
Attack Name	Delay_Attack
[-] Attack Type	Modify Packets
Attacker Node	1
[-] Layer Type	Mac
[-] Specify Mac Address from File	No
Mac Address	01-0C-CD-01-00-2A
[-] MODP Attack Type	Delay
Delay	1 seconds

Figure 7: EXataCPS Attack Editor

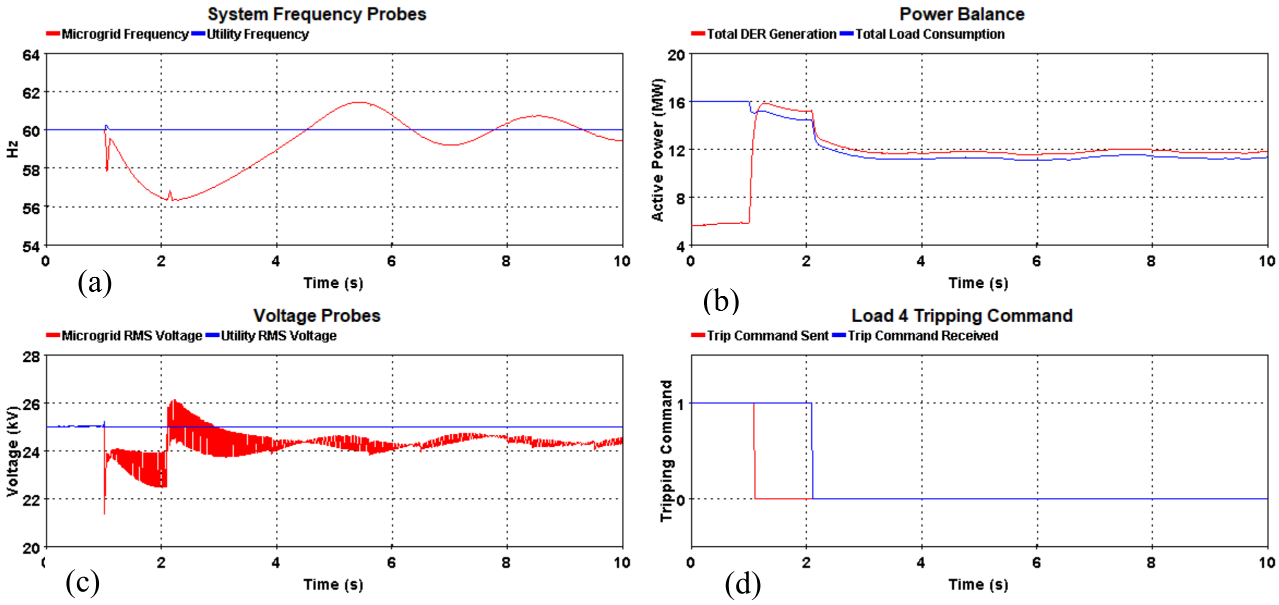


Figure 8: Large frequency deviation (8a), Unbalance power between generation and consumption before being regulated (8b), Hard and longer voltage dip down to 23 kV, with increased voltage oscillations (8c), Before being regulated back to its nominal value (8d)

As a result, the load 4 shedding function may fail to operate in the required timeframe. In this case this will cause severe unbalance between power generation and load consumption. Moreover, oscillations on MG nominal operation parameters such as e.g., frequency, voltages, etc., Figure (8) it may also result in severe consequences like a blackout

Conclusion

The goal of this testing is to introduce the CPS security testing platform based real time simulator and provide a user-friendly overview of how to set up IT/OT systems real-time security testing. A demo of a software-in-the-loop, MGC use-case study against a cyber delay attack is presented. This exercise begins by generating real-time traffic among MG units using the IEC 61850 GOOSE protocol. The impact on the MG nominal operation parameters is then demonstrated by introducing a cyber-attack (delay attack) to check and push beyond the limits, resulting in enhanced cybersecurity and resiliency of digital energy systems smart grid.

Exercise 3: Hands on Training on CPS Real-time Co-simulation Tool (EXataCPS-HYPERSIM) – Real-Time Co-Simulations of a Microgrid MG Active Power Managements Against Modify Data Packet MODP Attack

Goals	To develop CPS testing scenarios that provides a hands-on demonstration of the impact of MODP Attacks on the cyber space of energy systems It is used to train trainers and security professionals about the importance of securing network data and their impacts on OT systems Stimulus techniques that may be used to prevent such attacks.
	L. Zhang, S. Li, L. Wihl, M. Kazemtabrizi, S. Qaseem, J. Paquin, S. Labbé,” Cybersecurity Study of Power System Utilizing Advanced CPS Simulation Tools”,
Main material resources	https://wiki.opal-rt.com/display/DOCHS/Getting+Started, IEC 61850 IEC Just Published Just Published, IEC Webstore https://wiki.opal-rt.com/display/DOCHS/HYPERSIM+User+Documentation https://wiki.opal-rt.com/display/DOCHS/EXata+CPS+%7C+Installation+and+Li cense+Request SCALABLE EXataCPS1.1 Documents
Hours assigned	
Assignment criteria	Training session followed by group discussion Extra work of creating complete models
Done by	All partners

Introduction

The MG CPS security demo model explained within exercise two is used in third exercise. Where the MG description, list of units and their specifications based on HYPERSIM and EXataCPS are wildly presented and no need to overview again (refer to exercise two for more details). A hands-on demonstration of the impact of Modify Data Packet Attacks on a network, in exercise three is introduced to cyber system to alter the MG steady-state operation. Mainer changes within the HYPERSIM demo model need to be implemented and explained within the rest of this exercise [1-2].

MODP cyber attack

A Modify Data Packet Attack is a type of cyberattack that involves intercepting and modifying data packets sent across a communication network. The attacker captures network packets, modifies the contents of the initiated data, and then resends the data packets to their intended destination. The impact of this MODP attack on CPS security can be significant, as it can severely impact and alter the MG steady-state operation, resulting in a blackout and causing harm to the energy system network operator and customers.

To carry out this attack, the EXataCPS software is used to target the data packets sent by MG units. This MODP attack is executed by manipulating specific bytes within the transmitted packets (man-in-the-middle). These bytes are reserved for holding measurements within the data packets. The manipulated data packets are then resent to their final destination. The MGC received uncritical data (fabricated measurements) based on manipulated data packets and made a false controlling decision, which impacted the MG's steady-state operation and reduced its reliability.

Real-time CPS Security Model Conducting at OPAL RT

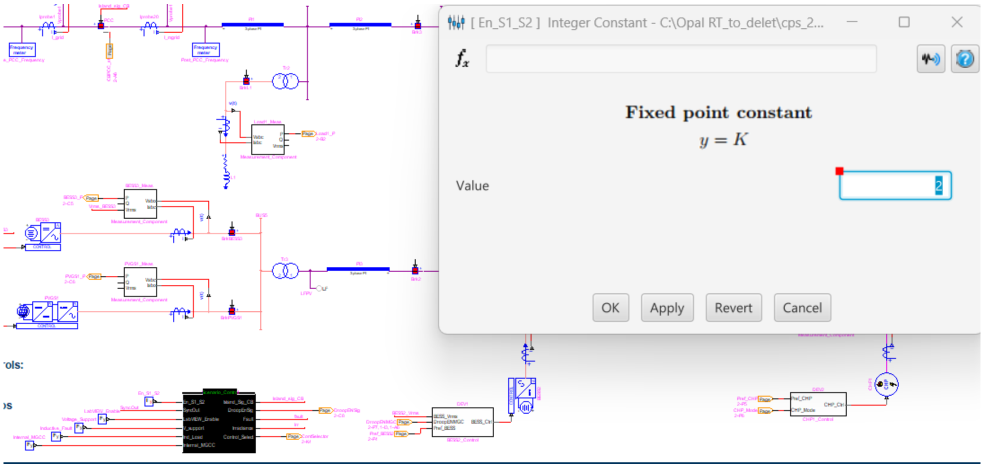
The CPS security simulation model is made up of two models: the HYPERSIM OT model and the EXataCPS IT model. These two models are linked internally by virtual adapters. Where the MODP attack is carried out by EXataCPS on the IT system, the impact is on the OT system, which is monitored by “ScopeView”. Here is the procedure for implementing and running both models in real-time on the OPAL RT simulator.

HYPERSIM OT model Implementation

Open HYPERSIM software as presented at exercise one, then follow the steps below [6],

- Step 1: from the option window chose “Open Example” from “Open Example File” section
- Step 2: navigate to the “Cyber_Physical_Mirigrid” example within the examples tree --> double click --> choose a folder to copy the demo example
- Step 3: HYPERSIM GUI showing “Cyber_Physical_Mirigrid” demo example pops up The HYPESIM model consists from two view pages one for the single line diagram and the second for the internal MGC. From page 2, change the “En_S1_S2” parameter to (2) (Figure 1) this will activate the MODP attack “secnario2”.
- Step 4: Click at Targets “Target” to check that the target is connected and available

- Step 5: Click at ⚙ “settings” and change the “Target” filed from “localhost” to online target, and “Simulation mode” filed to “Real-time” -->Apply-->close. Now the simulation is ready to run in real-time from the “Start” button.
- Step 6: Open “ScopView” change the view to “Scenario 2” at the bottom left.



[En_S1_S2] Integer Constant - C:\Opal RT_to_delet\cps_2...

Fixed point constant

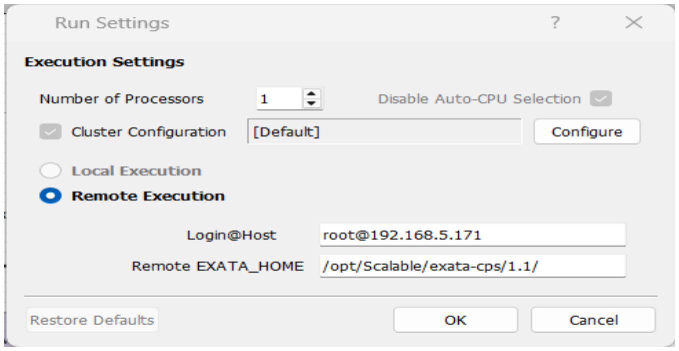
$y = K$

Value: 2

Figure 3: HYPERSIM En_S1_S2 block and changing parameter window

EXataCPS IT model Implementation

- Step 1: click on EXata CPS “EXataCPS” located at HYPERSIM GUI toolbar--> Open EXataCPS GUI
- Step 2: click on 📁 located at EXataCPS GUI toolbar --> navigate to the folder that the user saves the CPS example (from HYPERSIM configuration step 2 above)
- Step 3: at EXataCPS saved folder --> find “Cyber_Physical_Microgrid.config” file --> Open
- Step 4: Click at ⚙ located at Run toolbar --> “Run Setting” window opened. Check the configuration must be same as (Figure 2). the software are ready to run on real-time simulator



Run Settings

Execution Settings

Number of Processors: 1

Disable Auto-CPU Selection ☑

☑ Cluster Configuration: [Default]

○ Local Execution

● Remote Execution

Login@Host: root@192.168.5.171

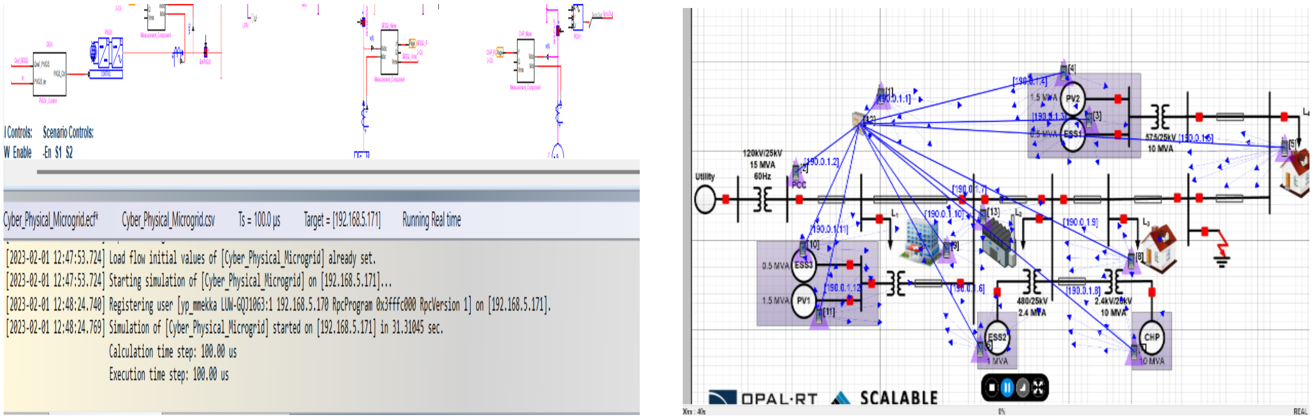
Remote EXATA_HOME: /opt/Scalable/exata-cps/1.1/

Figure 2: EXataCPS Run Setting configuration

Run both software configuration in Real-time at OPAL RT

After both software are ready to run the CPS demo example configuration files in real-time at OPAL RT simulator.

- Step 1: At the beginning user need to run HYPERSIM at first, since HYPERSIM need to create “name space” and assign “eno2” adapter for EXataCPS (Figure 3)
- Step 2: then user need to run EXataCPS as in exercise 1, section 6.2.3

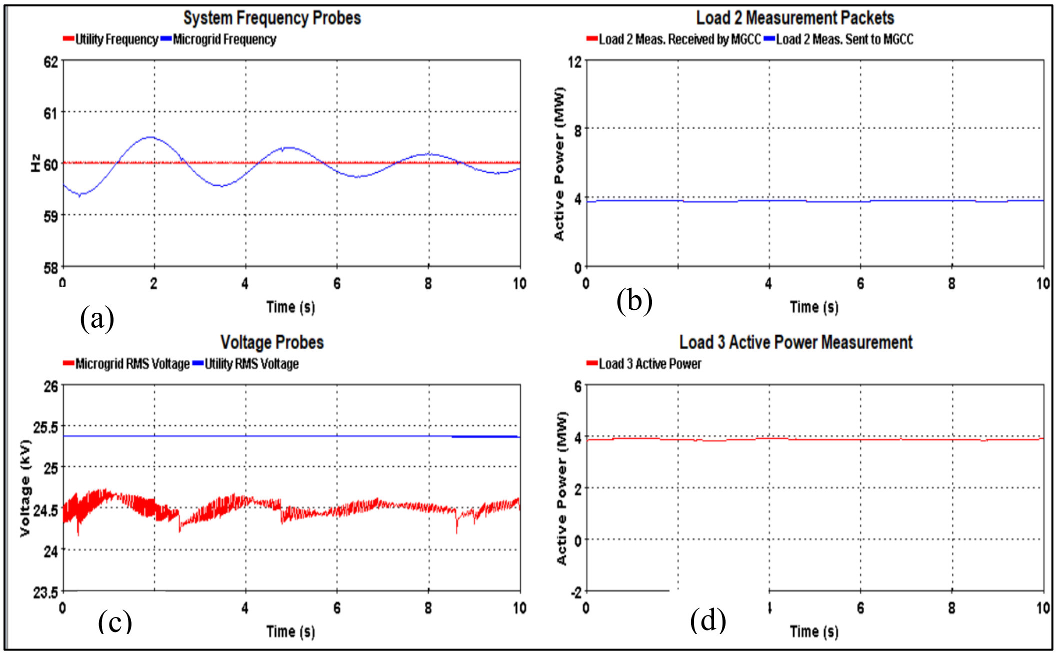


```
Cyber_Physical_Microgrid.ecf*    Cyber_Physical_Microgrid.csv    Ts = 100.0 µs    Target = [192.168.5.171]    Running Real time
[2023-02-01 12:47:53.724] Load flow initial values of [Cyber_Physical_Microgrid] already set.
[2023-02-01 12:47:53.724] Starting simulation of [Cyber_Physical_Microgrid] on [192.168.5.171]...
[2023-02-01 12:48:24.740] Registering user [yp_mmekka LUW-GQJ1063:1 192.168.5.170 RpcProgram 0x3fffc000 RpcVersion 1] on [192.168.5.171].
[2023-02-01 12:48:24.769] Simulation of [Cyber_Physical_Microgrid] started on [192.168.5.171] in 31.31045 sec.
                          Calculation time step: 100.00 us
                          Execution time step: 100.00 us
```

Figure3: HYPERSIM, EXataCPS output logs show that it runs in real-time successfully at OPAL RT simulator [7]

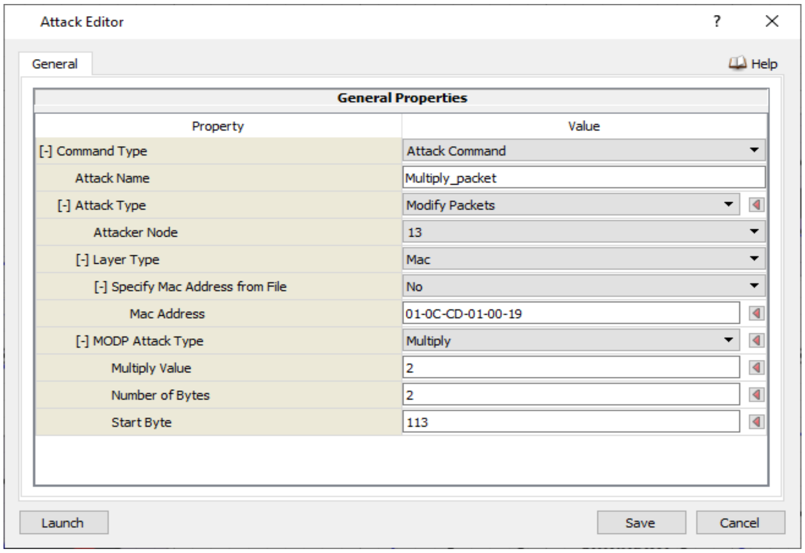
MG Active power Managements Against MODP

After the MG is islanded at second one in this demo example, and it has been disconnected Load 4 in previous exercise 2 based on the unbalance condition between the generated power and loads consumptions. As a result, the MG is operating in islanded steady-state operation mode as illustrated at Figure 4, from “ScopeView”.



Figures 4: (4c). PCC is stable at the utility nominal value 24.4 kV; frequency deviation be maintained within acceptable boundaries Figure (4a). Figure (4b) Load 2 measurements packets perceived and send to MGC are overlapped. Load 3 active power is steady near 4MW Figure (4d).

Then a cyber-attack MODP is introduced to Load 2 measurements that manipulated before they are received by the MGC by using EXataCPS “Attack Editor” targeting node 13 Load 2 as illustrated in Figure 5. Where the MODP attack manipulated 2 bytes starting from byte 113 which they are reserved for holding the load 2 measurements. As a result, the MGC trip and reconnect periodically Load 3 as illustrated in Figure 5d. Furthermore, the MG phase huge degradation in power quality Figure 6c and high frequency oscillations Figure 6a.



Attack Editor

General

General Properties

Property	Value
[-] Command Type	Attack Command
Attack Name	Multiply_packet
[-] Attack Type	Modify Packets
Attacker Node	13
[-] Layer Type	Mac
[-] Specify Mac Address from File	No
Mac Address	01-0C-CD-01-00-19
[-] MODP Attack Type	Multiply
Multiply Value	2
Number of Bytes	2
Start Byte	113

Figure 5: EXataCPS Attack Editor

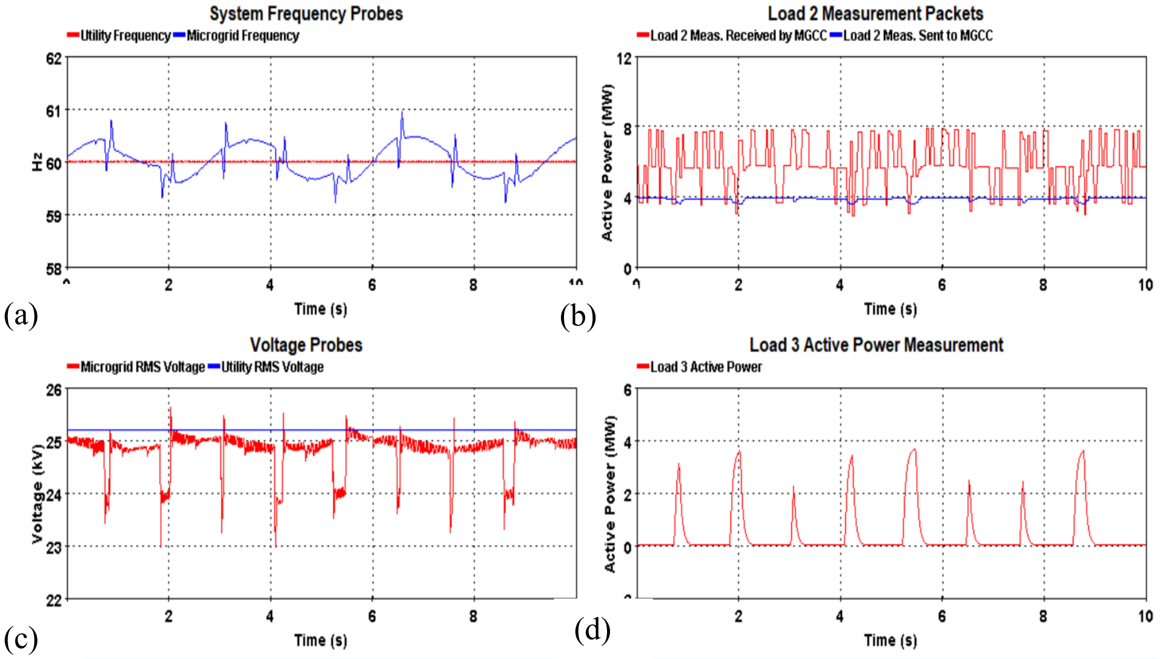


Figure 6: MG Large frequency deviation 6a, Hard and longer voltage dip down to 23 kV, with increased voltage oscillations 6c. Load 3 trip and reconnect periodically by MGC 6d. Load 2 measurements a data packets are duplicated and send to MGC by man-in-the-middle attack MODP 6b.

Conclusion

The aim of CPS security testing exercise 3 is to introduce the CPS security testing platform-based real-time simulator and to provide a user-friendly overview of how to set up the EXataCPS and HYPERSIM that simulates both the IT/OT systems in real-time. Hands-on demonstration of MODP's impact Attacks on the cyber space of energy systems are presented in order to educate trainers and security professionals on the importance of network data security and its implications for OT systems. In addition to stimuli techniques that could be used to prevent such MODP attacks, resulting in improved cybersecurity and resilience of digital energy systems smart grid.

Exercise 4: Hands on Training on CPS Real-time Co-simulation Tool (EXataCPS-HYPERSIM) – Real-Time Co-Simulations of a Microgrid based on Controller Hardware-In–the-Loop C-HIL

Goals	Demonstrate the CPS security testing platform based on C-HIL that provides a hands-on training on the developments of MGC on the external instruments field-programmable gate array FPGA development of real time co-simulation C-HIL communication network across MG and external controller using IEC 61850 GOOSE on the OPAL RT simulator
Pre-reading material	https://wiki.opal-rt.com/display/DOCHS/Getting+Started, IEC 61850 IEC Just Published Just Published, IEC Webstore libiec61850, [online], https://libiec61850.com/ https://wiki.opal-rt.com/display/DOCHS/HYPERSIM+User+Documentation https://wiki.opal-rt.com/display/DOCHS/EXata+CPS+%7C+Installation+and+License+Request
Main material resources	https://wiki.opal-rt.com/display/DOCHS/Getting+Started, IEC 61850 IEC Just Published Just Published, IEC Webstore https://wiki.opal-rt.com/display/DOCHS/HYPERSIM+User+Documentation https://wiki.opal-rt.com/display/DOCHS/EXata+CPS+%7C+Installation+and+License+Request SCALABLE EXataCPS1.1 Documents
Hours assigned	
Assignment criteria	Training session followed by group discussion Extra work of creating complete models
Done by	All partners

Introduction

Real-time co-simulation C-HIL is a valuable test method based on the control system hardware is integrated with a simulation environment a digital model (digital twins). The HIL test setup typically includes the controller hardware a real-time simulation environment, communication network with assigned communication protocol and measurement and analysis tools to verify the controller's performance Figure 1.

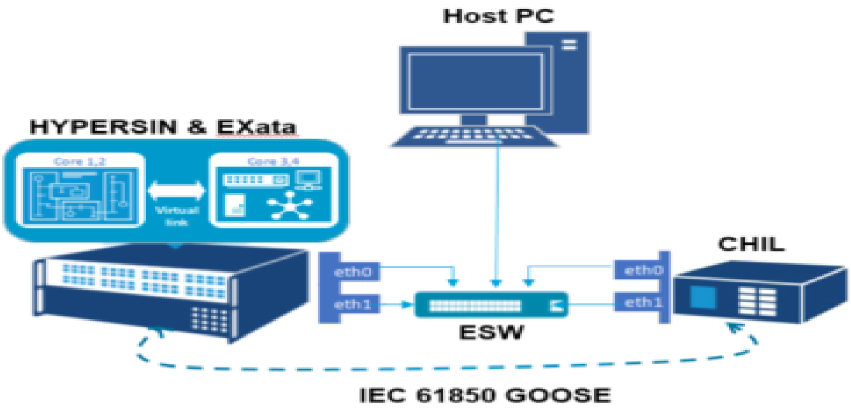


Figure 1: Real-time co-simulation C-HIL setup

Professionals can test the controller's behavior and performance in this environment without the need for physical MG equipment. This enhanced scalability and allows for faster more cost-effective testing, as well as increased safety, because the controller can be tested using the "what if" concept, which allows for the implementation of various scenarios without exposing the controller to physical risks. Furthermore, the external controller is unaware that the data received is from digital twin models or from actual physical systems. [1-2 and 4].

Development of the MGC

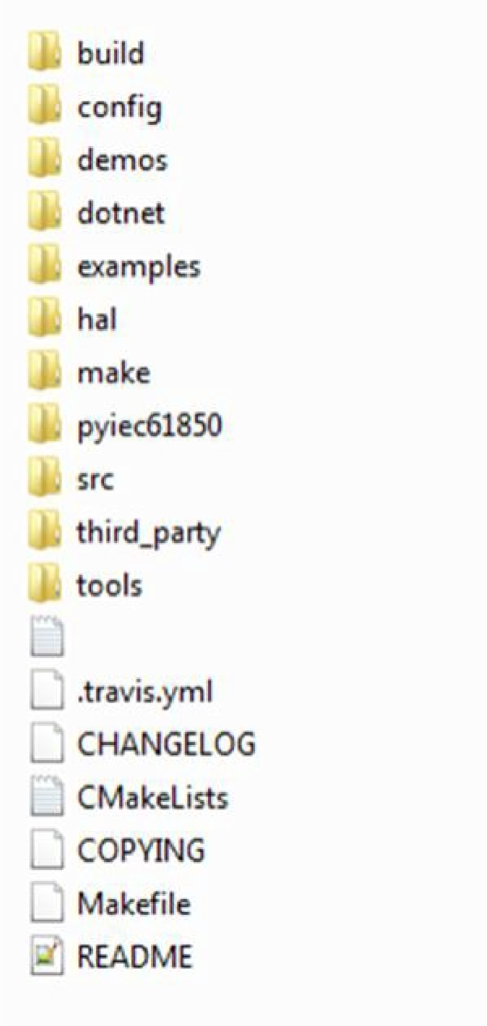
Before proceeding further, it is worth mentioning that this section is not intended to teach Linux neither C language or IEC 61850 protocol. It is a brief illustration of the open source libiec61850, library that been used to develop MGC based on the IEC 61850 communication protocols. This developed MGC can be implemented in to a low-cost microcontroller boards system on chip (SoC) e.g., Raspberry pi, BeagleBone etc., or ether in to field-programmable gate array (FPGA) and then been used for the C-HIL testing.

libiec61850 library introduction and installation

The libiec61850 library implements the IEC 61850 communication protocol on top of the Manufacturing Message Specification (MMS) in standard C language. It also sup-ports intra-substation communication via Generic Object-Oriented Substation Event (GOOSE) and sample value (SV). The goal of this project is to provide an easy starting point for the implementation of the IEC 61850 communication standard while minimizing the overhead information associated with the standard. That provide a very portable solution and can run on embedded systems and microcontrollers. Further-more, the library provides a set of simple examples that can be used as a starting point for the user to develop different MG applications. [3].

In order to install the library user needs to follow the steps:

- Step 1: In Linux system downloaded libiec61850 file and save it in a new directory and name it.
- Step 2: Unzip the libiec61850 file using the command tar –xvzf "name of the file".tar.gz
- Step 3: build the library libiec61850 using the "make" based system is required to cross-compile for embedded Linux systems. After this step the project folder should be as illustrated in Figure 2.



build
config
demos
dotnet
examples
hal
make
pyiec61850
src
third_party
tools

.travis.yml
CHANGELOG
CMakeLists
COPYING
Makefile
README

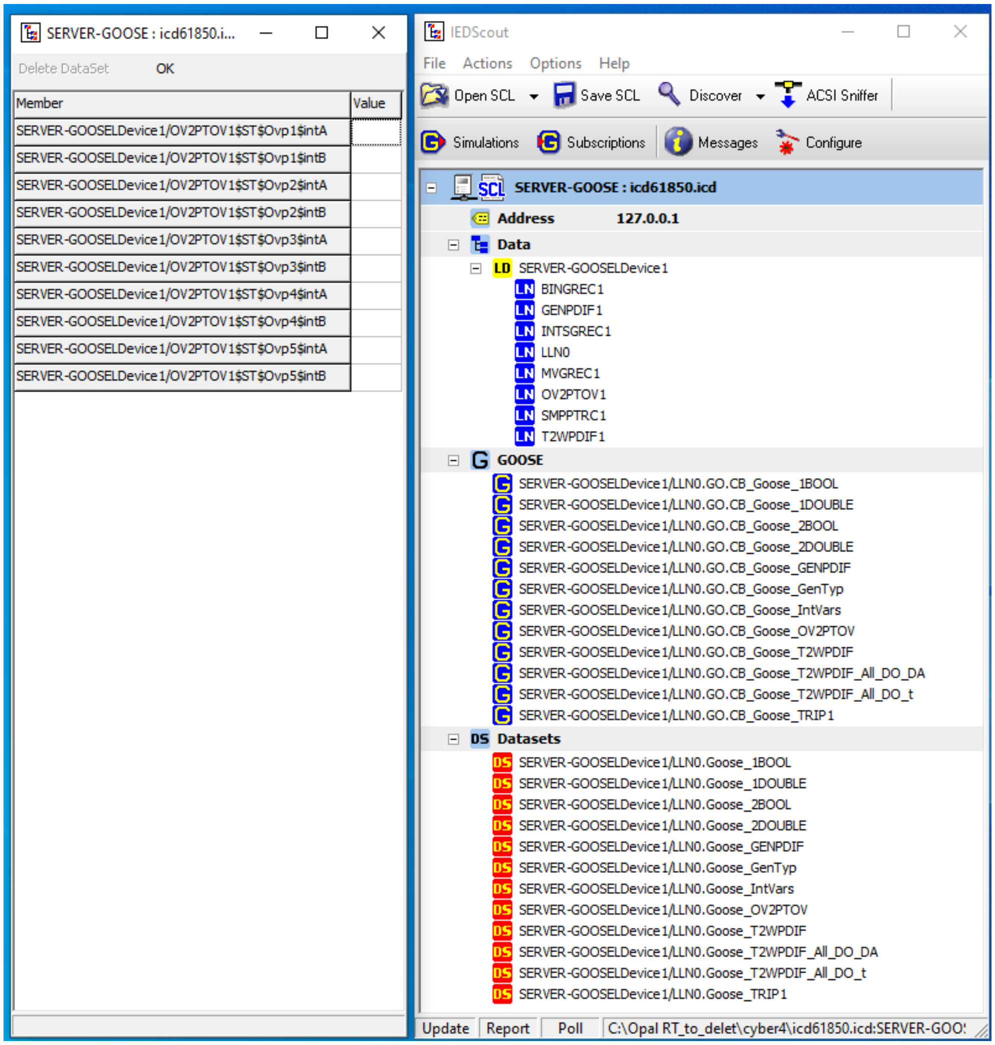
Figure 2: libiec61850 project folders

Develop the MGC along with IEC 61850 specifications

Development steps of the MGC that will be implemented as C-HIL is presented here. The development steps begin with [5].

- Step 1: creation of an IEC 61850 Substation Configuration Description (SCL) file.

- Step 2: Within the SCL file, user need to create the MGC object-oriented data model, that includes selected logical nodes (LN), data objects (DO), and data attributes (DA)s that are appropriate for handling and processing measurements data from the "MG units," to MGC
- Step 3: GOOSE datasets need to be created.
- Step 4: create and configure the GOOSE control blocks (GCB)s that use step 3 output.
- Step 5: finalize the GCB configuration by configuring the interfaces' GCB parameters such as GOOSE ID, GOOSE configuration revision, GOOSE publishing MAC address, GOOSE subscribing MAC address etc Figure 3.



Member	Value
SERVER-GOOSELDevice1/OV2PTOV1$ST$Ovp1$intA	
SERVER-GOOSELDevice1/OV2PTOV1$ST$Ovp1$intB	
SERVER-GOOSELDevice1/OV2PTOV1$ST$Ovp2$intA	
SERVER-GOOSELDevice1/OV2PTOV1$ST$Ovp2$intB	
SERVER-GOOSELDevice1/OV2PTOV1$ST$Ovp3$intA	
SERVER-GOOSELDevice1/OV2PTOV1$ST$Ovp3$intB	
SERVER-GOOSELDevice1/OV2PTOV1$ST$Ovp4$intA	
SERVER-GOOSELDevice1/OV2PTOV1$ST$Ovp4$intB	
SERVER-GOOSELDevice1/OV2PTOV1$ST$Ovp5$intA	
SERVER-GOOSELDevice1/OV2PTOV1$ST$Ovp5$intB	

SCL SERVER-GOOSE : icd61850.icd

Address 127.0.0.1

Data

- LD SERVER-GOOSELDevice1
  - LN BINGREC1
  - LN GENPDIF1
  - LN INTSGREC1
  - LN LLN0
  - LN MVGREC1
  - LN OV2PTOV1
  - LN SMPPTRC1
  - LN T2WPDIF1

GOOSE

- SERVER-GOOSELDevice1/LLN0.GO.CB_Goose_1BOOL
- SERVER-GOOSELDevice1/LLN0.GO.CB_Goose_1DOUBLE
- SERVER-GOOSELDevice1/LLN0.GO.CB_Goose_2BOOL
- SERVER-GOOSELDevice1/LLN0.GO.CB_Goose_2DOUBLE
- SERVER-GOOSELDevice1/LLN0.GO.CB_Goose_GENPDIF
- SERVER-GOOSELDevice1/LLN0.GO.CB_Goose_GenTyp
- SERVER-GOOSELDevice1/LLN0.GO.CB_Goose_IntVars
- SERVER-GOOSELDevice1/LLN0.GO.CB_Goose_OV2PTOV
- SERVER-GOOSELDevice1/LLN0.GO.CB_Goose_T2WPDIF
- SERVER-GOOSELDevice1/LLN0.GO.CB_Goose_T2WPDIF_All_DO_DA
- SERVER-GOOSELDevice1/LLN0.GO.CB_Goose_T2WPDIF_All_DO_t
- SERVER-GOOSELDevice1/LLN0.GO.CB_Goose_TRIP1

DS Datasets

- DS SERVER-GOOSELDevice1/LLN0.Goose_1BOOL
- DS SERVER-GOOSELDevice1/LLN0.Goose_1DOUBLE
- DS SERVER-GOOSELDevice1/LLN0.Goose_2BOOL
- DS SERVER-GOOSELDevice1/LLN0.Goose_2DOUBLE
- DS SERVER-GOOSELDevice1/LLN0.Goose_GENPDIF
- DS SERVER-GOOSELDevice1/LLN0.Goose_GenTyp
- DS SERVER-GOOSELDevice1/LLN0.Goose_IntVars
- DS SERVER-GOOSELDevice1/LLN0.Goose_OV2PTOV
- DS SERVER-GOOSELDevice1/LLN0.Goose_T2WPDIF
- DS SERVER-GOOSELDevice1/LLN0.Goose_T2WPDIF_All_DO_DA
- DS SERVER-GOOSELDevice1/LLN0.Goose_T2WPDIF_All_DO_t
- DS SERVER-GOOSELDevice1/LLN0.Goose_TRIP1

C:\Opal RT_to_delet\cyber4\icd61850.icd:SERVER-GOO

Figure 3: MGC SCL file hierarchy

Implementation of the designed MGC at FPGA

At this point the output of step 1.3.2 (configured SCL file) is used to build the MGC at specified SoC board e.g., FPGA used here and the process steps are presented as follows;

- Step 1; create two files (static_model.c and static_model.h) from the SCL file using the generating source codes offered by libiec61850 library.

- Step 2: Active power management function in C language for the MGC controlling user need to develop in a way that complies with software-in-the-loop (SIL) preliminary algorithms developed by Opal HYPERSIM in exercise 2.
- Step 3: dispatching controlling signal GOOSE message need to be created that will be send by MGC back to MG units. The overall procedure for designing the MGC HIL controller with all processing steps is presented in Figure 4

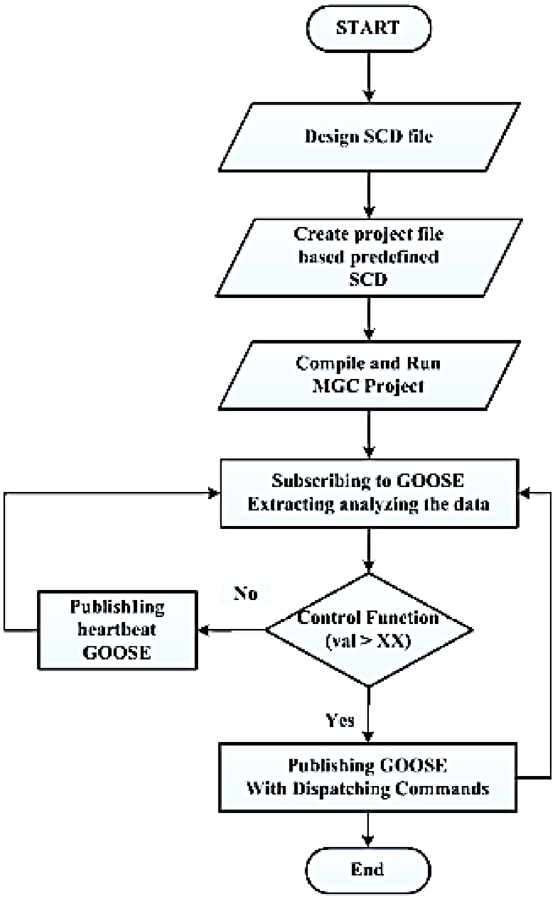

START
Design SCD file
Create project file based predefined SCD
Compile and Run MGC Project
Subscribing to GOOSE Extracting analyzing the data
No
Publishing heartbeat GOOSE
Control Function (val > XX)
Yes
Publishing GOOSE With Dispatching Commands
End

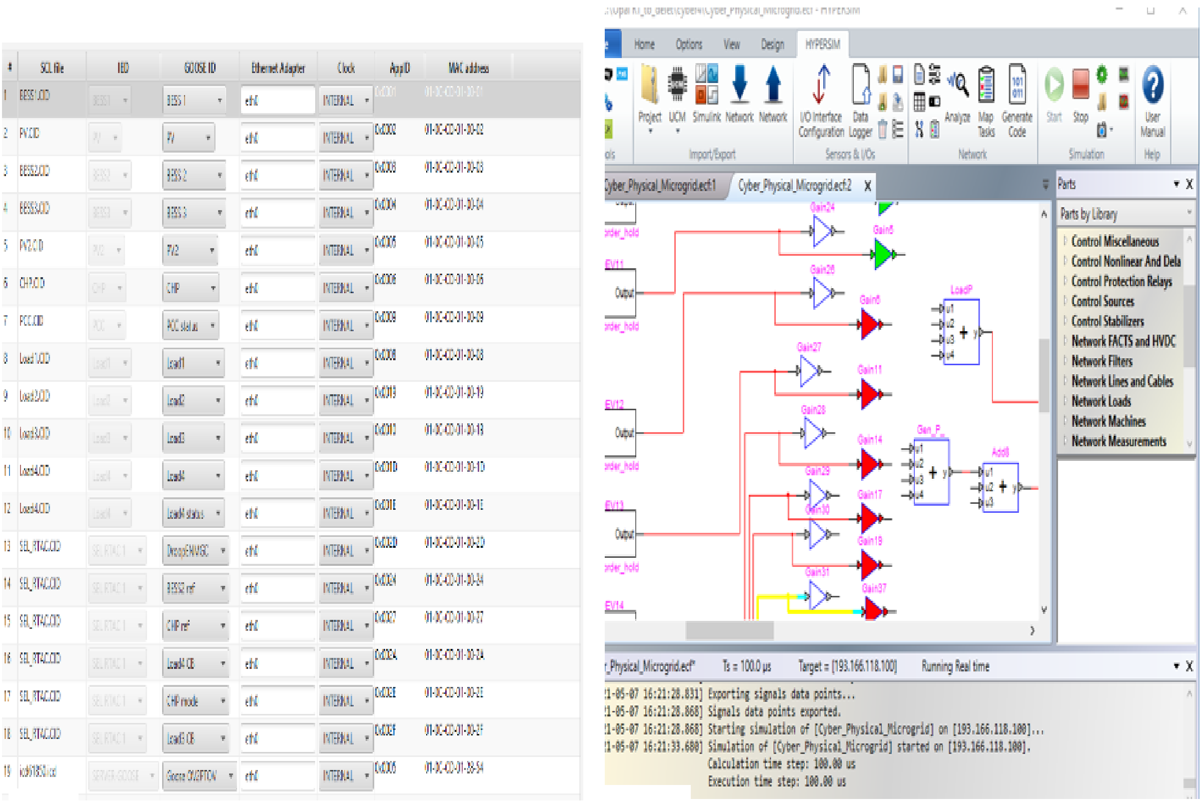
Figure 4: MGC development steps flow chart where the check condition If it is True, the MGC needs to go to step, which is publishing a new GOOSE message that needs to be subscribed by the model. According to this GOOSE message an open CB command is sent. Then, it will return back to the previous step. Whereas, if the output is False, the MGC will send a heartbeat GOOSE messages without any changes.

C-HIL MG Simulation Model at HYPERSIM OPAL-RT

In this forth exercise, the MG CPS security demo model described in exercise two is used. Where the MG description, list of units, and specifications based on HYPERSIM is wildly presented and there is no need to go over it again (refer to exercise two for more details). However minor changes within the HYPERSIM CPS security demo model need to be done before running the model in real time as follows [6-7],

- Step 1: Open HYPERSIM software as administrator that presented at exercise one

- Step 2: from the option window chose “Open Example” from “Open Example File” section
- Step 3: navigate to the “Cyber_Physical_Mirigrid” example within the examples tree --> double click --> choose a folder to copy the demo example
- Step 4: HYPERSIM GUI showing “Cyber_Physical_Mirigrid” demo example pops up. The HYPESIM model consists from two view pages one for the single line diagram and the second for the internal MGC. --> go to the internal MGC view page ad the blocks Figure 2a.
- Step 5: design a GOOSE data set that hold all the MG unites measurements that will be sent to external MGC
- Step 6: from HYPERSIM “I/O interface” the “Sensor and I/Os” section at HYPERSIM toolbar --> change the GOOSE transmission adaptor to physical adaptor “eth0” with which the GOOSE messages will be transferred via communication network Figure 2b. Where Ethernet RJ45 cables are used to link OPARL RT simulator and the external MGC Figure 1.
- to link the test instruments through the Ethernet switch
- Step 7: also, from “I/O interface” --> choose “IEC 61850” right click --> “Convert to V2”, this step converts IEC 61850 legacy to IEC 61850 version two (V2) standard.
- Step 8: Click at Targets “Target” to check that the target is connected and available
- Step 9: Click at zpnitts2 “settings” and change the “Target” filed from “localhost” to online target, and “Simulation mode” filed to “Real-time” -->Apply-->close. Now the simulation is ready to run in real-time from the “Start” button



#	SCL file	IED	GOOSE ID	Ethernet Adapter	Clock	AppID	MAC address
1	BESS1.CID	BESS1	BESS 1	eth0	INTERNAL	0x0001	01-0C-CD-01-00-01
2	PV.CID	PV	PV	eth0	INTERNAL	0x0002	01-0C-CD-01-00-02
3	BESS2.CID	BESS2	BESS 2	eth0	INTERNAL	0x0003	01-0C-CD-01-00-03
4	BESS3.CID	BESS3	BESS 3	eth0	INTERNAL	0x0004	01-0C-CD-01-00-04
5	PV2.CID	PV2	PV2	eth0	INTERNAL	0x0005	01-0C-CD-01-00-05
6	CHP.CID	CHP	CHP	eth0	INTERNAL	0x0006	01-0C-CD-01-00-06
7	PCC.CID	PCC	PCC status	eth0	INTERNAL	0x0009	01-0C-CD-01-00-09
8	Load1.CID	Load1	Load1	eth0	INTERNAL	0x0008	01-0C-CD-01-00-08
9	Load2.CID	Load2	Load2	eth0	INTERNAL	0x0019	01-0C-CD-01-00-19
10	Load3.CID	Load3	Load3	eth0	INTERNAL	0x0013	01-0C-CD-01-00-13
11	Load4.CID	Load4	Load4	eth0	INTERNAL	0x001D	01-0C-CD-01-00-1D
12	Load4.CID	Load4	Load4 status	eth0	INTERNAL	0x001E	01-0C-CD-01-00-1E
13	SEL_RTAC.CID	SEL RTAC 1	DroopCHMGC	eth0	INTERNAL	0x002D	01-0C-CD-01-00-2D
14	SEL_RTAC.CID	SEL RTAC 1	BESS2 ref	eth0	INTERNAL	0x0024	01-0C-CD-01-00-24
15	SEL_RTAC.CID	SEL RTAC 1	CHP ref	eth0	INTERNAL	0x0027	01-0C-CD-01-00-27
16	SEL_RTAC.CID	SEL RTAC 1	Load4 CB	eth0	INTERNAL	0x002A	01-0C-CD-01-00-2A
17	SEL_RTAC.CID	SEL RTAC 1	CHP mode	eth0	INTERNAL	0x002E	01-0C-CD-01-00-2E
18	SEL_RTAC.CID	SEL RTAC 1	Load3 CB	eth0	INTERNAL	0x002F	01-0C-CD-01-00-2F
19	iec61850.icd	SERVER-GOOSE	Goose OVERTON	eth0	INTERNAL	0x0005	01-0C-CD-01-28-54

(a

_Physical_Microgrid.ecf*   Ts = 100.0 µs   Target = [193.166.118.100]   Running Real time

```
21-05-07 16:21:28.831] Exporting signals data points...
21-05-07 16:21:28.868] Signals data points exported.
21-05-07 16:21:28.868] Starting simulation of [Cyber_Physical_Microgrid] on [193.166.118.100]...
21-05-07 16:21:33.680] Simulation of [Cyber_Physical_Microgrid] started on [193.166.118.100].
                       Calculation time step: 100.00 us
                       Execution time step: 100.00 us
```

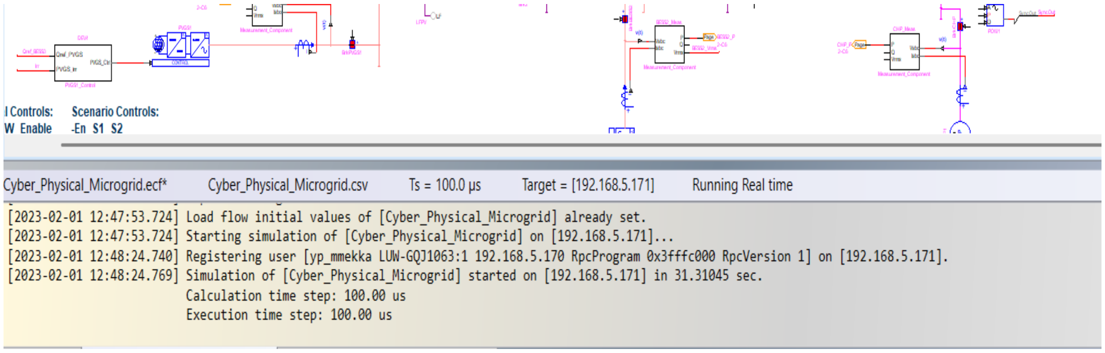
(b)

Figure 4: MGC model (a), IEC 61850 GOOSE transmission setting (b)

Run the C-HIL in Real-time at OPAL RT and the external MGC

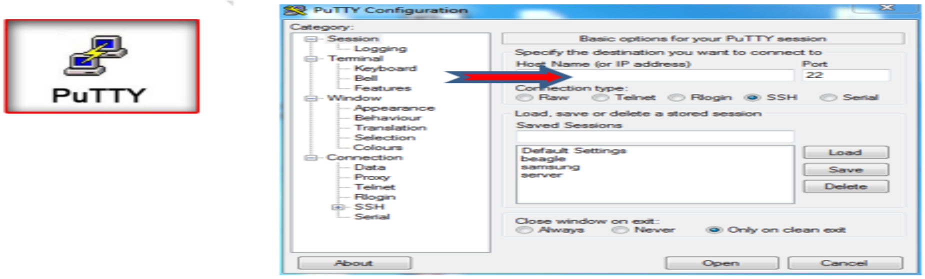
At OPAL RT simulator.

- Step 1: user need to run the model within the HYPERSIM software at first from the "Simulation Start button", (Figure 5)
- Step 2: Host machine need to Connecting to the MGC via SSH over Ethernet communication network IP address using e.g., "PuTTY" software, (Figure 6)
- Step 3: At the root directory navigate to "libiec61850-1" folder by (cd libiec61850-1) and press enter, do the same steps until you reach the "goose_subcriber" director
- Step 4: At the external MGC, user need to check the GOOSE subscription parameters in the "goose_subscriber_example.c" file by typing "nano goose_subscriber_example.c" and modify them based on the OPAL RT HYPERSIM model of the publisher GOOSE parameters such as (appId, goose control block reference), "Ctrl+x" to exit, "Y" to save changes.
- Step 5: In order to run the "goose_subscriber_example" execution machine file type ". /goose_subscriber_example" and press enter



```
Cyber_Physical_Microgrid.ecf*    Cyber_Physical_Microgrid.csv    Ts = 100.0 µs    Target = [192.168.5.171]    Running Real time
[2023-02-01 12:47:53.724] Load flow initial values of [Cyber_Physical_Microgrid] already set.
[2023-02-01 12:47:53.724] Starting simulation of [Cyber_Physical_Microgrid] on [192.168.5.171]...
[2023-02-01 12:48:24.740] Registering user [yp_mmekka LUW-GQJ1063:1 192.168.5.170 RpcProgram 0x3fffc000 RpcVersion 1] on [192.168.5.171].
[2023-02-01 12:48:24.769] Simulation of [Cyber_Physical_Microgrid] started on [192.168.5.171] in 31.31045 sec.
                          Calculation time step: 100.00 us
                          Execution time step: 100.00 us
```

Figure 5: HYPERSIM, output logs show that it runs in real-time successfully at OPAL RT simulator [5]

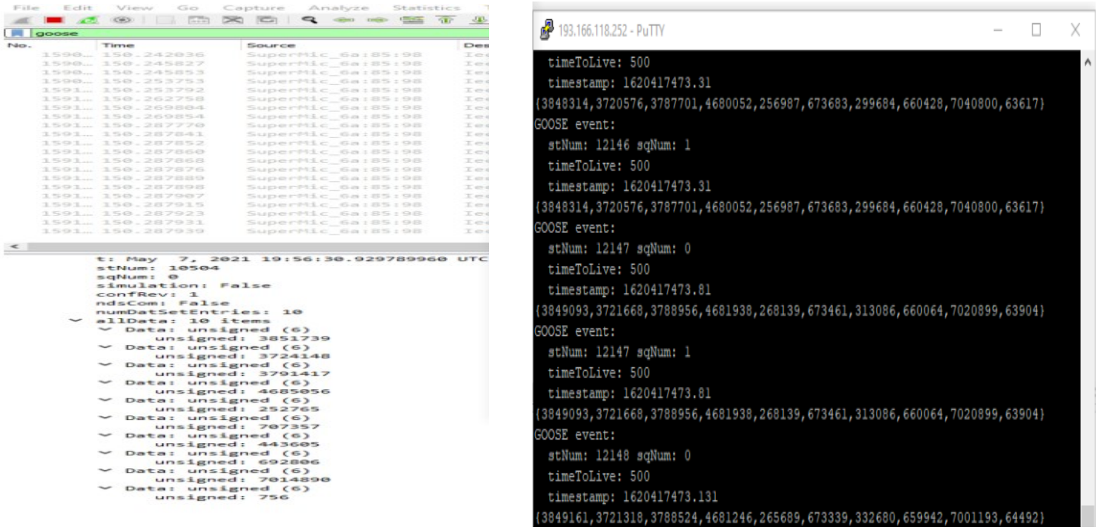


Host Name (or IP address) Port 22
Connection type: ○ Raw ○ Telnet ○ Rlogin ● SSH ○ Serial
Close window on exit: ○ Always ○ Never ● Only on clean exit

Figure 6: PuTTY software SSH connection

External MGC based C-HIL Testing Results

The external MGC has been successfully subscribed to the GOOSE messages published from the OPAL RT real-time simulator as illustrated in Figure 7. In addition, it shows the tenth measurements that were extracted from the received GOOSE messages. All these extracted parameters are printed out to be shown on the output of the MGC control terminal. As well as, Wireshark sniffing tool is used to capture the GOOSE traffic and analysed also the tenth measurements associated within the captured GOOSE messages. To this point building on this setup different attack scenarios need to be developed to test the performance of the external MGC e.g., to measure the 1-second delay attack effects on the MG behaviour. This delay attack is implemented within the MGC C code project file.



```
No.        Time            Source                    Des
1590...    150.242036      SuperMic_6a:85:98         Iec
1590...    150.245827      SuperMic_6a:85:98         Iec
1590...    150.245853      SuperMic_6a:85:98         Iec
1590...    150.253753      SuperMic_6a:85:98         Iec
1591...    150.253792      SuperMic_6a:85:98         Iec
1591...    150.262758      SuperMic_6a:85:98         Iec
1591...    150.269804      SuperMic_6a:85:98         Iec
1591...    150.269854      SuperMic_6a:85:98         Iec
1591...    150.287770      SuperMic_6a:85:98         Iec
1591...    150.287841      SuperMic_6a:85:98         Iec
1591...    150.287852      SuperMic_6a:85:98         Iec
1591...    150.287860      SuperMic_6a:85:98         Iec
1591...    150.287868      SuperMic_6a:85:98         Iec
1591...    150.287876      SuperMic_6a:85:98         Iec
1591...    150.287889      SuperMic_6a:85:98         Iec
1591...    150.287898      SuperMic_6a:85:98         Iec
1591...    150.287907      SuperMic_6a:85:98         Iec
1591...    150.287915      SuperMic_6a:85:98         Iec
1591...    150.287923      SuperMic_6a:85:98         Iec
1591...    150.287931      SuperMic_6a:85:98         Iec
1591...    150.287939      SuperMic_6a:85:98         Iec

    t: May  7, 2021 19:56:30.929789960 UTC
    stNum: 10504
    sqNum: 0
    simulation: False
    confRev: 1
    ndsCom: False
    numDatSetEntries: 10
  ˅ allData: 10 items
     ˅ Data: unsigned (6)
           unsigned: 3851739
     ˅ Data: unsigned (6)
           unsigned: 3724148
     ˅ Data: unsigned (6)
           unsigned: 3791417
     ˅ Data: unsigned (6)
           unsigned: 4685056
     ˅ Data: unsigned (6)
           unsigned: 252765
     ˅ Data: unsigned (6)
           unsigned: 707357
     ˅ Data: unsigned (6)
           unsigned: 443605
     ˅ Data: unsigned (6)
           unsigned: 692806
     ˅ Data: unsigned (6)
           unsigned: 7014890
     ˅ Data: unsigned (6)
           unsigned: 756
```

```
  timeToLive: 500
  timestamp: 1620417473.31
{3848314,3720576,3787701,4680052,256987,673683,299684,660428,7040800,63617}
GOOSE event:
  stNum: 12146 sqNum: 1
  timeToLive: 500
  timestamp: 1620417473.31
{3848314,3720576,3787701,4680052,256987,673683,299684,660428,7040800,63617}
GOOSE event:
  stNum: 12147 sqNum: 0
  timeToLive: 500
  timestamp: 1620417473.81
{3849093,3721668,3788956,4681938,268139,673461,313086,660064,7020899,63904}
GOOSE event:
  stNum: 12147 sqNum: 1
  timeToLive: 500
  timestamp: 1620417473.81
{3849093,3721668,3788956,4681938,268139,673461,313086,660064,7020899,63904}
GOOSE event:
  stNum: 12148 sqNum: 0
  timeToLive: 500
  timestamp: 1620417473.131
{3849161,3721318,3788524,4681246,265689,673339,332680,659942,7001193,64492}
```

Figure 7: The controller subscriber to OPAL RT GOOSE messages show in MGC secure Shell SSH window and in Wireshark

While other case study e.g., in order to simulate the MODP Man-in-the-middle attack, user need to duplicate load 2 measurements by multiplying the measurements by two before sending it to the MGC. The MGC will extracted the manipulated value from the attacked GOOSE message, and implement the check emergency condition in C code. In this case, MGC will send a dispatching command to disconnect load three to fulfil the emergency condition requirements. More analysis of the MGC GOOSE messages received data and discussion will be offered as extra work for the trainees. Table 1 present the external MGC test measurements and status.

Table 1: MGC test measurements and status

#	Load 1	Load 2	Load 3	Load 4	BESS 1	PV2	BESS3	PV1	CHP	BESS2	PCC	Load 4	Times
1	3875128	3781908	3848320	4745503	324750	12391	313977	657533	5051008	901410	0	1	0.0
2	3886356	3801598	3867824	4767855	259598	12483	298115	657889	6063510	918955	0	1	0.042
3	3895487	3816629	3883259	4787274	274980	12601	377263	658062	7273871	917240	0	1	0.242
4	3875302	3774808	3840516	4734144	247771	12376	426814	653924	6496688	876106	0	1	0.442
5	3628191	3564326	3625103	4470147	263691	16993	441651	637882	7891500	838352	0	1	0.592
6	3765598	3749752	3827546	1686397	248053	23351	461370	631774	10581547	834129	1	1	0.692

7	3909675	3902159	3986019	620417	245663	12185	469602	632663	10463299	804188	1	0	0.742
8	3974832	3972181	4058661	83989	277405	13576	477779	640318	10418736	701015	1	0	0.842
9	3890752	3891604	3976228	11382	255274	13001	300181	639228	10458335	576783	1	0	0.942
10	3826802	3825120	3908149	1550	251834	12469	322624	638311	10355982	451723	1	0	1.042
11	3732894	3727222	3808487	33	245095	11622	391792	637551	10235066	213576	1	0	1.242
12	3690010	3682557	3763187	4	256227	11291	425556	636786	10221290	50770	1	0	1.392
13	3641555	3632102	3711959	0	246946	11559	459342	636631	10287160	16591354	1	0	1.642
14	3633773	3623783	3703827	0	264419	11588	468186	636659	10321279	16508713	1	0	1.742
15	3590677	3583693	3663737	0	257502	11815	297380	636543	10566320	16331793	1	0	1.992
16	3598518	3590426	3670833	0	269892	11909	320684	637715	10613529	16271080	1	0	2.092
17	3609756	3600123	3681195	0	262805	12193	376833	639748	10676801	16190208	1	0	2.242
18	3619217	3608758	3690369	0	255081	12704	416467	640535	10742324	16121152	1	0	2.392
19	3625488	3614647	3696720	0	262562	12717	435432	640812	10775080	16080759	1	0	2.492
20	3626222	3615002	3696941	0	246758	12848	449834	641852	10813148	16045081	1	0	2.592
21	3632638	3621064	3703834	0	268585	13121	469777	643488	10851610	15984287	1	0	2.792
22	3632638	3621064	3703834	0	268585	13121	469777	643488	10851610	15984287	1	0	2.942
23	3626697	3614814	3697480	0	269335	13077	487092	643417	10894398	15908551	1	0	3.142
24	3620534	3608708	3690556	0	253137	12728	492642	642716	10922676	15872675	1	0	3.392
25	3625170	3613655	3695241	0	262391	12730	495402	642562	10947493	15851677	1	0	3.592
26	3635648	3624687	3705456	0	246261	12075	497503	641979	11015321	15833760	1	0	3.842
27	3652699	3642308	3723380	0	261976	11988	498563	641891	11061342	15826077	1	0	3.992

28	3663804	3653738	3734697	0	262258	12213	499013	643486	11097553	15822607	1	0	4.092
29	3682987	3676505	3757403	0	261760	11780	340178	660614	11322578	15817035	1	0	4.392
30	3730492	3723068	3804667	0	246047	12057	416973	676022	11389880	15816532	1	0	4.642
31	3756276	3748616	3830909	0	247682	12198	450782	678971	11427376	15819407	1	0	4.842
32	3744688	3740558	3822954	0	261123	12726	295332	681969	11538362	15825974	1	0	5.042
33	3767777	3761076	3844537	0	270543	12807	383797	687555	11489827	15832647	1	0	5.292
34	3766261	3758344	3841976	0	247153	13027	430778	691158	11444541	15840773	1	0	5.492
35	3420712	3026410	3097530	0	254640	12482	431382	574554	13162528	15900753	1	0	5.592
36	3481128	3080810	3142453	2461343	261153	55416	446532	558530	14976107	15913154	0	1	5.642
37	3695901	3437038	3499438	4124898	248874	12305	296997	640848	8820065	16028727	0	1	5.742
38	3284208	2808361	2857941	3503537	233906	6080	305507	676664	18641205	16206118	0	1	5.842
39	2185473	951665	971935	1210253	170248	16739997	316376	654396	8566525	16347311	0	1	5.992
40	2524608	1476101	1511032	1878933	194460	37169	287098	656788	9263116	16118254	0	1	6.142
41	3003512	2307134	2348593	2904624	216277	890	383725	678090	16462531	16358187	0	1	6.392
42	2183750	947049	966862	1204584	166107	16739091	373027	654642	7105342	16475085	0	1	6.492
43	3365212	2888729	2940613	3631400	229022	5964	478575	685109	15257266	16526335	0	1	6.842
44	2679160	1799928	1833785	2272158	219546	56372	453296	670359	16689547	16598107	0	1	6.942
45	2033099	674768	691849	867267	153846	16725867	422931	645703	4613850	16511752	0	1	7.142
46	2241362	1029269	1054758	1316209	175922	16732549	446194	651958	4286363726	16350067	0	1	7.442
47	3490675	3052096	3109465	3840381	245479	4538	527203	687073	792295	16504408	0	1	7.642
48	3362295	2907956	2960414	3655672	256122	5806	510359	692531	17456987	27897	0	1	7.842

49	3211773	2675600	2725831	3368524	282416	2211	503972	690994	19365029	16742147	0	1	7.992
50	3583477	3274431	3333838	4113662	244491	7432	517104	699436	15932444	16606684	0	1	8.092
51	3605799	3319334	3377019	4166183	244917	9007	508935	700682	15919332	16479252	0	1	8.242
52	3077862	2492068	2538611	3137973	229706	247	489138	695139	4284319804	16716571	0	1	8.442
53	3599355	3349466	3410801	4208360	249106	9223	512640	705241	12373359	410397	0	1	8.642
54	3930460	3927243	3995515	4924185	248776	13423	304915	711150	13442109	718635	0	1	8.842
55	4636077	4653300	4747321	2098696	268523	23734	275530	717507	15092822	729369	1	0	9.092
56	5005018	5031641	5136746	104749	267168	17947	257677	719715	14089063	576036	1	0	9.242
57	4857688	4882122	4983069	2030	277816	16448	255330	713545	13722050	358826	1	0	9.442
58	4616189	4635409	4730578	83	266288	15154	260752	701329	13175867	157475	1	0	9.642
59	4251829	4262130	4349867	12	253596	13391	280713	695326	12272546	16681857	1	0	9.942
60	4134640	4140856	4226667	7	254309	12547	317121	693895	11928201	16611125	1	0	10.042
61	3930348	3929973	4012176	3	269399	11678	389496	692142	11314555	16488264	1	0	10.242
62	3787650	3782931	3862756	1	250291	10521	431577	690397	10939397	16386566	1	0	10.442
63	3703386	3695994	3774444	0	246741	10379	456955	689842	10732796	16302856	1	0	10.642

From Table 1, columns 2-5 (e.g., Load1 to Load 4) shows the loads active power consumption, while columns 6-11 (e.g., BESS1, PV2, NESS3, PV1, CHP, BESS2) illustrates the DERs active power generation all in Watts. Moreover, columns 12-13 (e.g., PCC, load4) presents the status of the PCC and load 4 circuit breakers, while column 14 (e.g., Time s) shows the time stamp in second. Here, each data object collects three data attributes for each parameter (Val, q, t) i.e., and they are encapsulated in the GOOSE message data set. Row 6 illustrates the first islanding that is implemented in 0.692 s by disconnecting the MG by changing PCC status to true (islanded), whereas load 4 CB will be shed in 0.742 s since at this point no delay attack is implemented. In a similar vein, row 36 shows the MG that is reconnected to the grid and load 4, it is also reconnected and starts consuming active power and column 4 starts showing measurements

Conclusion

The aim of this exercise is to introduce the CPS security testing platform-based on OPAL RT real-time simulator and provide a user-friendly overview of how to set up real-time security testing for IT/OT systems. As well as a demonstration of a hardware-in-the-loop HIL, external MGC use-case study against a cyber-attack. This exercise begins by generating real-time traffic among MG units using the IEC 61850 GOOSE protocol and published across communication network. External MGC design, deployment and execution on a micro-controller board is presented. The MGC control solution and its relevant communication system have been designed in compliance with the IEC 61850. The obtained results demonstrate that the external MGC approach and data modelling of various IEC 61850 predefined data object LNs are correct for the design of the power balance control/protection function against cyber-attack. Further extra work will be offered for trainees on the analysis of the data received by MGC, implementation of different cyber-attacks on different MG application/functionalities to extend the knowledge and to validate the feasibility of the developed approach.