Improving the robustness of urban electricity networks
IRENE

D4.1 Toolsets of supply demand prediction and threat identification and security classification

Document version: 1
Document status: Final
Project Document Date: 26/01/2017
Workpackage Contributing to the Project Document: WP4
Dissemination level: confidential
Author(s):
Eng Tseng Lau (QMUL)
Anhtuan Le (QMUL)
Michael Chai (QMUL)
Yue Chen (QMUL)
Andrea Ceccarelli (UNIFI)
Tommaso Zoppi (UNIFI)
Andrea Bondavalli (UNIFI)
Alexandr Vasenev (UT)
Oliver Jung (AIT)
Sandford Bessler (AIT)
Edward Lambert (Ethos)

TABLE OF CONTENTS

List of Tables

List of Figures

1 EXECUTIVE SUMMARY

The purpose of this WP4 deliverable (Toolsets of supply demand prediction and threat identifications and security classification) of the project IRENE is to develop and describe an open modelling toolset that brings together the methodologies and policies to evaluate and measure the mitigation outcomes. Models of different configurations and mitigation methods are integrated into this tool to enable the efficiency of fault and attack mitigation measures, the energy resilience analysis, and the impact on different critical infrastructures. The tool is used to investigate threats in the smart grid and to put into practice the identified solutions. Based on identified use case scenarios, risk analysis and criticality rating the developed analysis framework provides a decision support for planning approaches.

The open modelling tool will focus on the coding, graphic-user-interface (GUI) design and analysis. It is based on an iterative approach to software development that is intended to deliver working software quickly and evolve this quickly to meet the changing requirements.

The tasks for this deliverables are:

Task 4.1: Design and develop a modelling tool
An IRENE open toolset is envisaged that is fully integrated with IRENE's methodologies and policies, and that facilitates a range of services.

Task 4.2: Implementation of supply demand prediction and security assessment toolset
In this task the security assessment and prediction model are implemented which developed in WP2 and WP3 into the collaborative tool to perform the ranking and analysis of each use case scenarios.

In Chapter 3 of this deliverable, the evolutionary threat analysis (ETA) tool is developed that supports city evolution and planning for response. Such tool builds on WP2, taking the current grid scenario and an envisioned grid change. The tool identifies new threats whenever an envisioned plan is applied to the current scenario.

In Chapter 4 of this deliverable, the BayesianFAIR tool is developed that BayesianFAIR that allows numerical threat assessment based on the states of the four FAIR factors including Contact, Action, Threat capability, and Control Strength. The numerical outputs given by BayesianFAIR can help to further rank threats in the same category (e.g. High or Very High) that helps to highlight the impact of a single factor on the overall assessment, and from that, help to point out the most influential factor that a system operator should focus on to build effective mitigation plans.

In Chapter 5 of this deliverable, the IRENE Overall Grid Modeling (OGM) tool is developed based on the methodologies developed from WP1 – WP3, in combination with the threat assessment tool – BayesianFAIR (Bayesian Factor Analysis of Information Risk). Such modelling tool serves as a Graphical User Interface (GUI) based engineering tool for fellow user (Stakeholder) to integrate, evaluate and update the existing grid infrastructure, critical infrastructures, islanding opportunities, demand changes and revised policies from Stakeholders. The tool is fully integrated with IRENE's methodologies and policies, and that facilitates a range of services.

In Chapter 6, the Microgrid Evaluation (MGE) Tool is presented that is an event based simulation of the interacting Customer Energy Management System (CEMS) controllers and the microgrid

(MG) controller. The load models, load predictions, flexibility are updated and the optimization models produce new local control actions, the MG controller creates new setpoints for the respective building controllers, etc.

The MGE tool is further used together with the Single Line Failure Simulation (SILFAST) tool that tests whether using the reduced microgrid loads calculated by MGE in each node of the high level topology still create overloaded lines.

2 INTRODUCTION

2.1 DESIGN AND DEVELOP A MODELLING TOOL

This subsection describes how the previously derived Stakeholder's Collaborative Framework and requirements from WP1 results are integrated.

Based on WP1 deliverables [1], the collaborative framework, in the perspective of IRENE project is established that aims to support the Stakeholders in maintaining the system resilience across the pre-defined scenarios and to promote necessary redundancies in outage events in order to minimise stress on demand/network load, and to maximise the social welfare. Overall, the Collaboration Framework allows Stakeholders to exhibit their fundamental or important role in maximising the grid performance. The lists of Stakeholders listed in [1] in the collaborative framework include Municipal authority planner, Distribution Network Operator (DNO), Developers, Critical Infrastructure Operator, Business and Citizen Representative.

The energy resilience planning system requirements in the Stakeholder's Collaboration Framework encompass:

1) Smart grid architecture and topology;
2) Ranking of critical services;
3) Clarification of the threats, impacts and mitigation to smart grid system;
4) Resilience enhancement;
5) Evaluation of mitigation options by fellow Stakeholders;
6) Development of policies to implement actions to improve the overall smart grid performance.

The main finding from WP1 [1] also envisaged a decision support tool as a critical function that will enable planners to model, manipulate, observe and evaluate the smart grid infrastructure by implementing the tool with elements such as existing grid infrastructure, critical infrastructures, islanding opportunities, demand changes, revised policies from Stakeholders, microgrid, distributed generations, ICT control, contingency and outage analysis, social and cost impact analysis.

Overall, an open modelling tool will be developed based on planning system requirements from fellow Stakeholders to foresee the outcome of the grid performance with respect to implemented/used case scenario, based on the policies and methodologies developed from IRENE deliverables within WP1 – 3. Such modelling tool serves as a GUI-based engineering tool for fellow Stakeholders to integrate and update the methodologies and policies. The tool is fully integrated with IRENE's methodologies and policies, and that facilitates a range of services.

2.2 IMPLEMENTATION OF SUPPLY DEMAND PREDICTION AND SECURITY ASSESSMENT TOOLSET

When the development of the IRENE toolset is completed, the security assessment and prediction model which developed in WP2 and WP3 are implemented into the toolset that are necessary to perform the ranking and analysis of each use case scenarios. The analysis for each used case scenario (from initial grid configurations towards decarbonisation scenario) through the integration of different IRENE toolset is validated in IRENE Deliverable D4.2 [2].

From the previous work within IRENE WP2 of D2.1 [1], a threat analysis methodology that provides incremental results is identified. Such methodology is intended to support city evolution and planning for response. An evolutionary threat analysis tool is developed that builds on methodologies developed in [1] to deal with smart grids that have strong dynamical topology, along with threat identification and lists of mitigations to reduce the effect of threats in smart grid. Such methodology is intended to support city evolution and planning for response.

The BayesianFAIR (Factor Analysis of Implemented Risk) threat assessment from WP2 of D2.2 [2] is implemented into the tool where the Stakeholders are informed of the infrastructure analysis of relevant effect vectors - *Contact, Action, TCap (Threat capability),* and *Control Strength*. All the Bayesian parameters are obtained from the FAIR tables as guided in the FAIR model [4] and encoded to the tool. Although the parameters are fixed for this particular implementation, they can be updated manually if users want to assess based on different FAIR tables. The assessments of the tools are adjusted to always be in-line with the FAIR assessments [3]**Fehler! Verweisquelle konnte nicht gefunden werden.**.

The IRENE WP3 [3] models the characterization of consumption data, predicts the electricity consumption using the day-night, week-weekend and seasonal periodicity and an aggregation technique using the Ensemble Kalman Filter, as well as the architectural basis to develop tools for the evaluation the effect of outages in different levels of the grid. In a mid-voltage, urban distribution grid, outages are handled that require topology changes such as individual link and node (generation) failures using mid-grid mathematical optimisation module. In contrast, at the microgrid (low) level, individual node or link failures are not handled but applied flexible loads and demand management mechanisms of microgrid and building controller mathematical optimisation technique for a “soft” degradation of service.

Overall, this deliverable is the design and implementation of IRENE toolset for WP4, which applies the policies and methodologies developed in WP1 – WP3. Such integration and validation of IRENE toolsets are presented in D4.2 [2] of WP4.

3 EVOLUTIONARY THREAT ANALYSIS (ETA) TOOL

Most of the approaches supporting the achievement of safety and security requirements are based on threat analysis processes that only focus on static scenarios. For example, the NIST 800-30 standard for conducting risk assessments [1], along with the NISTIR 7628 [2] standard that focuses on Smart Grids, provide a consolidated background for conducting such type of analysis but they lack in guidelines and methodologies amenable for supporting *planned evolution of infrastructures and especially of Smart Cities*, that is instead central in IRENE.

Tackling evolution of Smart Cities as main challenge, we identified in WP2 a threat analysis methodology that provides incremental results. This methodology is intended to support city evolution and planning for response. We present here a tool that builds on WP2 to support the application of such methodology. The rest of this section describes:

- the concept of evolutionary-oriented threat analysis;
- the design of our Evolutionary Threat Analysis tool;
- details on the tool: inputs, outputs and parameters;
- an example of evolutionary threat analysis taking as input data from [13].

3.1 THREAT ANALYSIS FOR PLANNED EVOLUTION

In the literature, different approaches considering *evolving scenarios* to provide different type of assurances are found, but they do not consider safety and security. Among others, the approach presented in [7] considers evolving scenarios in detecting recurring software failure patterns. The authors show the utility of considering evolution concerns in the detection process. Differently from [7] our objective is to consider how the Smart City evolves but with a different goal, i.e., detecting threats and corresponding mitigation strategies instead of software failure patterns.

Other approaches provide means to analyse security threats in evolving scenarios but they do not support the complete threat analysis process from threat identification to the enactment of corresponding security requirements. Approaches to the evolutionary threat analysis as [8] [7] are not thought for Smart Grids/Smart Cities since they do not consider explicitly emergent phenomena (i.e., behaviours originating by interactions among connected components) originated by evolutions thus they cannot primarily focus on risks associated on flow of information and control among autonomous grid components.

3.1.1 Emerging Phenomenas

A smart grid can be viewed as a complex system in which different constituent systems (smart meter, DER, Power Plants …) act their role depending on the implemented requirements and the mechanisms. The interaction between these separate components could lead to new macro level behaviours (considering the constituent components belonging to the micro level) which therefore are emerging ones because they are not built-in micro level properties but are generated due to these interactions. In [9] the authors formalize the following definitions:

Emergence: A phenomenon of a whole at the macro level is emergent if and only if it is new with respect to the non-relational phenomena of any of its proper parts at the micro level.

Resultant phenomenon: A phenomenon at the macro-level is resultant if it can be reduced to a sum of phenomena at the micro level.

These emerging phenomenas can be beneficial or adverse: for example, if we consider a plurality of water molecules under appropriate environmental conditions fluidity and wetness are beneficial concepts while a traffic jam due to the interaction between cars (that are the micro level compo-

nents) is an example of adverse emergent behaviour. Especially when you are looking for protecting your system from dangerous actions, it is mandatory to predict as best as you can the emerging behaviours with the aim to avoid situations in which some unexpected adverse behaviours compromise the correct execution of the system functionalities [9]. Being this work focused on threats, we focus on detrimental emergence.

3.1.2 Our contribution: Evolutionary Threat Analysis (ETA)

In order to provide safety and security requirements for Smart Grids it is essential to analyse the interdependency regulating the flow of information among entailed constituent systems. Their interactions and interdependencies may generate cascading effects, i.e., emerging phenomena, which represent possible security threats and damages. To avoid such situations, the transient threat analysis should be supported by new approaches able to deal with cascading contingency chains revealing the effect of evolving the grid.

The evolutionary threat analysis described in D2.1 [6] [10] adopts the guidelines defined from NIST in the SP 800-30 [4] regarding both the approach to follow and the main steps to perform to validate the risk assessment process. In particular, we followed an asset-oriented approach as defined in the NIST standard, by identifying threat events depending on critical assets of the grids, i.e., the internal behaviour of a component (e.g., a hospital) and their possible interactions. Differently from the NIST standard, we supported an incremental threat identification process that is carried out after the grid evolution.

Starting from the identification of impacts or consequences of the addition/removal of assets, our approach identifies the threats and/or the vulnerabilities that can arise due to this scenario's evolution. Consequently, the mitigation strategies to apply/remove are identified according to their threats traceability. The main steps of our methodology are (please refer to D2.1 [6] for more details):

1. *Evolve the scenario*: every time we update the scenario we repeat the analysis focusing the attention on the new assets introduced from this evolution, e.g., the addition of a Wind Farm;
2. *Detect Structural Threats*: considering the updated assets, we look at the considered threat list to understand if these components carry one or more intrinsic threats, meaning that these are introduced in the scenario with the addition of the new components;
3. *Detect Emerging Threats*: we investigate the interactions involving the novel added components thus highlighting corresponding complex emerging behaviours, e.g., the stadium competing against other buildings for energy during a match;
4. *Merge and Mitigate*: The results coming both from the structural and emerging threat analysis are merged and added to the partial results of the process. Since the threat events' set is now completed, we link the threat set with a corresponding mitigation set.

Once an evolved scenario is analysed, all the results coming from each iteration of the process are merged and added to the threat list, which contains information about threat events concerning its nature (structural or emerging), affecting components and corresponding mitigations strategies.
It is worth noting that since it is very difficult to link a threat event with a reasonable quantitative evaluation of its impact and likelihood in such a generic context, we will not weight the degree of harm and likelihood of threats' occurrences. In this section we present a tool that realizes the above methodology [10].

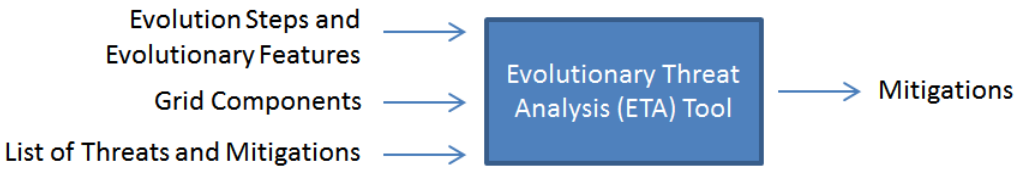
3.1.3 Evolutionary Scenarios

This evolutionary threat analysis is built on a dynamic and evolving Smart City /Smart Grid scenario. Following the guidelines defined in [1], we consider as evolution step a set of evolutionary features that lead the grid to change its functionalities. For example, considering features such as "*Changing grid maintaining strategy*", "*Adoption of an Automated Metering Infrastructure*" and "*Creating specific micro-grids with specific requirements and functionalities*" we call for an evolution that leads to build a new industrial district (see evolution step "*Building an industrial district*" in [6]).

Starting from a user-defined initial scenario, all the considered evolution steps are sorted and individually considered to obtain the threat analysis outcomes for all these intermediate steps. In such a way, emerging behaviours can be identified by analysing each changing in the scenario, taking into account all the connections among component targeting the possible arising behaviours.

3.2 DESIGN AND IMPLEMENTATION OF THE TOOL

The evolutionary analysis presented above has been implemented into an integrated tool along with the proposed algorithm determining the variation of mitigation strategies and the scenario-based distribution analysis. It is written in *Java* and it makes use of the *Colibri-Java* FCA API[1] to analyse the distribution of threats. Using evolutionary scenarios defined in terms of evolutionary features – and, consequently, grid components - and the mapping between threats and their corresponding required mitigation strategies, the algorithm performed satisfactorily with the scenario's inputs and given that it is polynomial w.r.t. to the inputs, we expect to have an acceptable scalability with larger scale Smart Grids.



According to the purpose of the whole IRENE Toolset, the tool is intended to support city planners when they have to plan – or to assess – an evolution of the existing grid that contributes to provide smart services in the near future. Anyway, evolving the grid leads to modifications that can introduce new threats or vulnerabilities (e.g., a substation that is fundamental for critical grid components) as well as architectural changes that need to be supported by the whole grid (e.g., energy rebalancing due to a failure in a connection or a generic component).

Moreover, considering threats happening in a scenario as a formal relation between two components (i.e., threats and scenarios) leads us to view the results of the threat analysis as a FCA structure, which can be seen from a *lattice* perspective (see Figure 3-3). The lattice can support city planners in evaluating the sets of mitigations associated to each threat, finally helping estimating the costs needed to implement mitigations and improve safety and security in the targeted scenario. This view established hierarchical relations between scenarios depending on threats, allowing also evaluating different "branches", i.e., different ways to mitigate threats among the ones identified by the evolutionary threat analysis tool.

3.2.1 Implementation Choices

We describe here our implementation choices, both regarding the tool respect to the project and regarding the efficiency of the tool itself.

[1] https://code.google.com/archive/p/colibri-java

- *Language*. We choose *Java* as reference platform since it is not OS dependent and since other tools in the toolset were developed with the same language. This will help the future integration of the single tools to build a unique toolchain.

- *Interface and I/O*. The tool has not a graphical interface since it is intended to be used in co-operation with other tools that offer a graphical user interface. However, the tool can be considered as a standalone resource that has its inputs and outputs into text files. This allows a simple integration with other tools that can read and write the input and output files to tune the preferences of the threat analysis tool according to their actual needs. Note that, as shown in Figure 3-3, the FCA results can be reviewed with the aid of *ConExp*, a tool that offer a graphical view on these results.

- *Performance*. The complexity of the threat analysis implemented in this tool is not so high to require deep performance analysis. However, during the CPU-intensive phase – while threats for each evolution step are listed – the tool executes the most expensive tasks in dedicated threads, to not lock the main thread responsible to collect the outcomes of the created threads. This will increase the performances of the tool in workstations where several (physical or virtual) CPUs are available. The scheduling of such threads is left to the default *Java* process that runs a preemptive priority-based scheduling algorithm.

3.2.2 Class Diagram

In Figure 3-1 we report the UML class diagram that summarizes the most relevant structures and relations of the evolutionary threat analysis tool.

Grid components are represented by the abstract class "*Component*" which has a type ("*ComponentType*") and consequently a category "*ComponentCategory*", e.g., smart home – *SH* - components are buildings (category *BLD*). Components can be "Connection" or "Building", representing respectively connections and nodes of the grid.

Keeping the evolutionary dimension of analysis in mind, a "*Scenario*" is defined by i) the old scenario (if any), and ii) an evolution step "*EvolutionStep*", which is defined by two sets of added and removed grid components. Considering both the old scenario and the evolution step we can call the "*threatAnalysis*" method, which activates the "*ThreatManager*" module to obtain a listing of all the structural and emerging threats in the targeted scenario.

The threat manager is responsible for loading the threats, the mitigations and the emerging rules defined by the user. As in [6], each "*Threat*" belongs to a "*ThreatCategory*" and can be associated to a list of "*Mitigation*" which are the abovementioned NISTIR security requirements. Through "*getCategoryThreats*", "*getComponentThreats*" and "*getEmergingThreats*", the threat manager is able to provide a list of threats that includes both structural and emerging threats for the scenario under analysis.

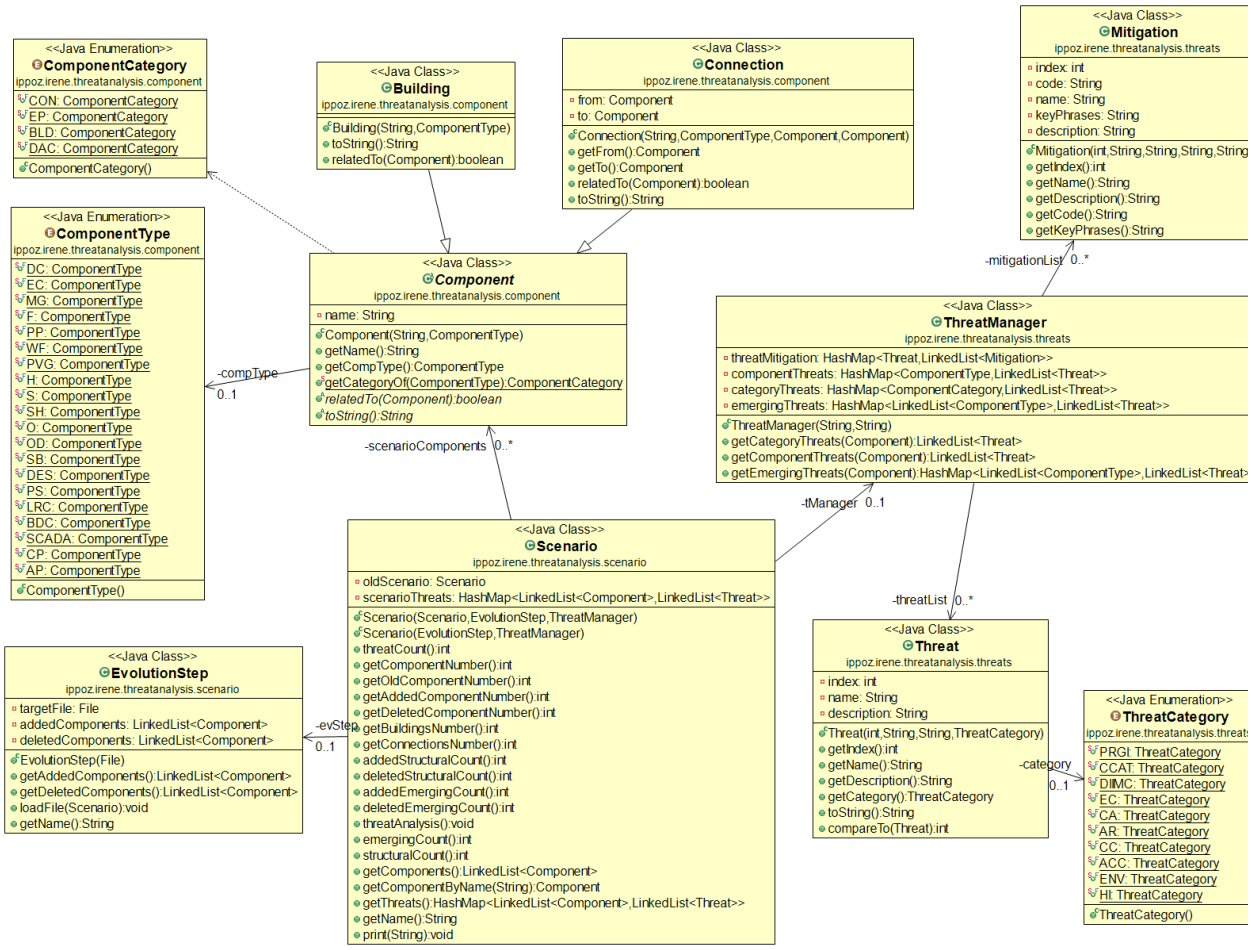


Figure 3-1: UML Class Diagram of the Evolutionary Threat Analysis Tool

Wrapper classes (that are not shown in Figure 3-1) are responsible to load the preferences, the evolution steps, the threat library and the user-defined rules for the threat analysis. Output is managed by such classes, which are also responsible to build an xml summary file that can be read by the *ConExp Colibri-Java* FCA tool.

3.2.3 Code Quality

The implemented code was checked to obtain quality metrics in order to give an overview on its complexity and on how it is written. In Table 3-1 we reported some of the main quality metrics [11] that are widely used as code quality indicators in software engineering. Moreover, we refactor this code aiming at eliminating the code flaws identified by *FindBugs* [12]. *FindBugs* was tuned to identify the following bug categories: security flaws, bad practices, dodgy code, and multi-threading correctness. All the bugs identified by *FindBugs* were eliminated, finally improving the robustness of the code according to these rules.

Table 3-1: Main quality metrics for the ETA Tool

Metric	Name	Detail	Average	St. Dev.	Max	Total
PAR	Number of Parameters	method	1.093	1.142	5	
CE	Efferent Coupling	package	1.167	0.687	2	
SIX	Specialization Index	type	0.101	0.193	0.667	
NOC	Number of Classes per Package	package	2.333	0.745	3	14
NOF	Number of Attributes	type	2.5	1.991	6	35
RMA	Abstractness	package	0.033	0.075	0.2	
RMD	Normalized Distance	package	0.565	0.308	0.857	
NSM	Number of Static Methods	type	1.857	3.889	13	26
TLOC	Total Lines of Code					1430
WMC	Weighted methods per Class	type	18.571	14.783	52	260
NOM	Number of Methods	type	6.571	7.178	29	92
NOP	Number of Packages					6
VG	*McCabe* Cyclomatic Complexity	method	2.203	2.573	21	10
LCOM	Lack of Cohesion of Methods	type	0.411	0.316	0.909	
MLOC	Method Lines of Code	method	8.229	9.164	39	971
NORM	Number of Overridden Methods	type	0.286	0.452	1	4

3.3 TOOL INPUTS, OUTPUTS, CONFIGURATION, EXECUTION

3.3.1 Inputs

Inputs are defined by:

- *Threat List*:
  - file "Threats.csv", defining the list of threats. The file is composed by 5 fields (Event Category, NIST Indexes, IRENE Index, Threat Event, Description) separated by commas, with one line of header;
  - file "Component_Threat.csv", defining the list of threats that can involve each component. The file is composed by 3 fields (Component Type, Threat Index, Motivation) separated by commas, with 1 line of header;
  - file "Category_Threat.csv", defining the list of threats that can involve each category of component. The file is composed by 3 fields (Component Category, Threat Index, Motivation) separated by commas, with 1 line of header;
  - file "Emerging_Threats.csv", defining the rules that define an emerging behaviour resulting from the interaction of two or more grid components. The file is composed by 3 fields (IRENE Index, Involved Components, Description) separated by commas, with 1 line of header. Components are separated by semicolons.
- *Mitigation List:*
  - File "Mitigations.csv", defining the list of mitigations (*NISTIR 7628 Security Requirements*). The file is composed by 5 fields (Mitigation Index, Code, Name, Key phrases, Description) separated by commas, with 1 line of header;
  - file "Threat_Mitigations.csv", defining the list of mitigations associated to each threat. The file is composed by 4 fields (Threat Index, Threat Description, Mitigation Indexes, Motivation) separated by commas, with 1 line of header.

- *List of Grid Components*: components are taken from the list in [6]. Allowed components are connections (EC, DC, MG, PS, LRC), buildings (F, H, S, SH, O, OD, SB, DES), energy providers (PP, WF, PVG) and other (BDC, SCADA, AP, CP).
- *Evolution Steps and Evolutionary Features*: these are user-defined. Each evolution step defines the addition or the removal of one or more components, to expand or reduce the wideness of the scenario. The evolution step is defined in a file with extension ".scenario" where the user defines the addition or the removal of one or more components. Note that removing a components leads to the cascading delete of the components that are now disconnected from the grid. In the example, deleting SH 2 would lead to delete all the connections with SH 2 if these are not linked with other components that are still considered in the grid.

3.3.2 Outputs

Lastly, the outputs are collected and stored in the same folder. The tool reports:

- ".summary" files, one for each evolution steps in which the constituent components (splitted in building and connections) and the involved threats (splitted in structural and emerging) are listed;
- "analysisStats.csv", which collects all the main information on the threat analyses executed on each considered evolution steps;
- the *Formal Concept Analysis (FCA)* output "*fcaGraphic.cex*", which defines a graph of the abovementioned analysis and that can be explored using the *conexp* tool. Opening this file with *conexp* tool leads to visualize all the threats affecting each scenario in a hierarchical view. In the figure we can see how the 9 considered scenarios are linked. While the 0, 1, 2 steps simply add components and thus increment the amount of threats that can affect the scenario, the step 3 both deletes and adds components, leading the identified threats to be different – and not a simple increment – respect to the ones identified in the step 2. Consequently, scenario 3 is not a "child" of scenario 2, but instead needs to be considered as different.

3.3.3 Setup of the parameters

All the preferences must be set using a "*threat_analysis.preferences*" file in the same directory of the executable *jar* file. This is used to define the main parameters of the tool, such as:

- FILE_FOLDER: the folder containing the files related to threats and mitigations;
- SCENARIO_FOLDER: the folder containing the scenarios
- OUTPUT_FOLDER: the folder in which outputs will be put
- THREAT_FILTER: specifies a filter on the threat list. Write "ALL" if you want to consider all the threats in the list; otherwise accepted filters are threat categories from **Fehler! Verweisquelle konnte nicht gefunden werden.** (e.g., CCAT), index intervals (e.g., 11-20) or single indexes (e.g., 1; 4; 23) separated by commas.

3.3.4 Running the Tool

The tool is an executable *.jar* that can be run via command line on *Windows*, *OSX* and *UNIX* systems invoking the *Java Virtual Machine* with "java –jar <pathname>/WP2_ThreatAnalysis.jar". The tool is compiled with the current standard version of *Java* (*Java* 8); therefore, it cannot be run on systems where *Java* is not installed or if *Java* 7 or previous versions are installed.

3.4 THREAT ANALYSIS EXAMPLE

Here we report the results of an execution of the tool using the threats, the mitigations and the evolutionary steps described in [6] (threats can be seen in Appendix B, mitigations in Appendix C, while the evolutionary steps are in Section 4.1.4 and reported in Figure 3-2 for clarity). First, files regarding threats and mitigations are read, summarizing the valid threats and mitigations. In the example, the tool shows that

[Info][executable.Main] Reading Threat Library
[Info][threats.ThreatManager] Available Threats: 38 threats
[Info][threats.ThreatManager] Available Mitigations: 19 mitigations
[Info][threats.ThreatManager] Emerging rules: 120 rules

38 threats, 19 mitigations and 120 possible emerging behaviours were loaded, corresponding to the outcomes reported, as example, in Tables 18 and 20 of [6]. These constitute the basis of the threat analysis process. Then, the evolution steps are loaded

[Info][executable.Main] Analyzing evolutions
[Info][engine.Analyzer] Analyzing 9 evolutionary scenarios

and each of them is analysed considering the information read in the previous step.

[Info][scenario.EvolutionStep] Evolution Step '0_InitialScenario' read: 15 added and 0 deleted components
[Info][engine.Analyzer] Scenario '0_InitialScenario': 172 structural and 17 emerging threats
[Info][scenario.EvolutionStep] Evolution Step '1_DiscoveringResources' read: 4 added and 0 deleted components
[Info][engine.Analyzer] Scenario '1_DiscoveringResources': 225 structural and 30 emerging threats
[Info][scenario.EvolutionStep] Evolution Step '2_GrowingNumberOfPeople' read: 6 added and 0 deleted components
[Info][engine.Analyzer] Scenario '2_GrowingNumberOfPeople': 304 structural and 51 emerging threats
[Info][scenario.EvolutionStep] Evolution Step '3_AddingKeyBuildings' read: 4 added and 2 deleted components
[Info][engine.Analyzer] Scenario '3_AddingKeyBuildings': 327 structural and 58 emerging threats
[Info][scenario.EvolutionStep] Evolution Step '4_InseringStorages' read: 6 added and 0 deleted components
[Info][engine.Analyzer] Scenario '4_InseringStorages': 413 structural and 94 emerging threats
[Info][scenario.EvolutionStep] Evolution Step '5_BuildingIndustrialDistrict' read: 2 added and 4 deleted components
[Info][engine.Analyzer] Scenario '5_BuildingIndustrialDistrict': 384 structural and 81 emerging threats
[Info][scenario.EvolutionStep] Evolution Step '6_InsertionSCADA' read: 2 added and 2 deleted components
[Info][engine.Analyzer] Scenario '6_InsertionSCADA': 386 structural and 77 emerging threats
[Info][scenario.EvolutionStep] Evolution Step '7_InstallingMicroGrids' read: 0 added and 0 deleted components
[Info][engine.Analyzer] Scenario '7_InstallingMicroGrids': 386 structural and 77 emerging threats

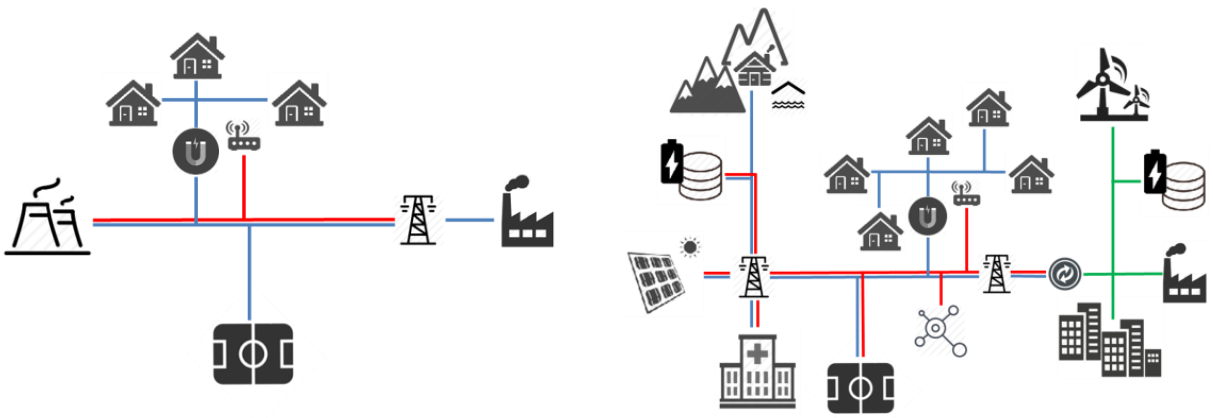


Figure 3-2: Evolution steps from [6]: 0_InitialScenario and 5_BuildingIndustrialDistrict

[Info][scenario.EvolutionStep] Evolution Step '8_ImprovingDecarbonisation' read: 2 added and 0 deleted components
[Info][engine.Analyzer] Scenario '8_ImprovingDecarbonisation': 414 structural and 80 emerging threats

In the examples, 9 evolutions of the grid are considered: an initial definition of the grid ("*0_initialScenario*") and 8 temporal evolutions, leading the grid being exposed to 494 threats (414 structural and 80 emerging) in its final state. In Table 3-2 we can observe the textual summary of the threat analysis execute on the scenarios mentioned above. In the first scenario all the components - and, consequently, the threats - are new, while other steps add or remove components from the previous scenario. The threats related to the scenarios changes accordingly: considering "*5_BuildingIndustrialDistrict*" we can observe that 2 components are added and 4 are removed, totalizing 384 structural and 81 emerging threats. 8% of this threats are new with respect to the previous "*4_InseringStorages*" scenario due to the addition of 2 components, while 17% of threats that affected the "*4_InseringStorages*" where eliminated due to the removal of 4 components.

Table 3-2: Textual Threat Analysis Summary

Scenario	Components						Structural			Emerging			Statistics (%)			
	old	new	del	all	build	conn	all	new	del	all	new	del	Str.	Em.	new	del
0_InitialScenario	0	15	0	15	8	7	172	172	0	17	17	0	91	9	100	0
1_DiscoveringResources	15	4	0	19	10	9	225	53	0	30	13	0	88	12	26	0
2_GrowingNumberOfPeople	19	6	0	25	13	12	304	79	0	51	21	0	86	14	28	0
3_AddingKeyBuildings	25	4	2	27	14	13	327	44	21	58	11	4	85	15	14	6
4_InseringStorages	27	6	0	33	17	16	413	86	0	94	36	0	81	19	24	0
5_BuildingIndustrialDistrict	33	2	4	31	16	15	384	29	58	81	8	21	83	17	8	17
6_InsertionSCADA	31	2	2	31	16	15	386	24	22	77	14	18	83	17	8	9
7_InstallingMicroGrids	31	0	0	31	16	15	386	0	0	77	0	0	83	17	0	0
8_ImprovingDecarbonisation	31	2	0	33	17	16	414	28	0	80	3	0	84	16	6	0

Together with this file and the *xml* file representing the input for the *ConExp Colibri-FCA* Tool, the tool prints more detailed summaries with the listings of all the threats for each scenario. The xml file can be opened with the *ConExp* tool as depicted in Figure 3-3. The threat analysis can be seen as table, showing which threat affects which component (if structural) or components (if emerging) in which scenario.

Otherwise, the lattice representation shows a graph where big nodes are scenarios, giving a hierarchical point of view on the entire threat analysis through different evolution steps. In the lattice reported in Figure 3-3 we can see how scenarios 6 and 7 are represented in the same point. The only change here is the substitution of the BDC component with a SCADA: since they are threatened by the same items, the results of the threat analysis do not change, leading to represent both scenarios in the same point of the graph.

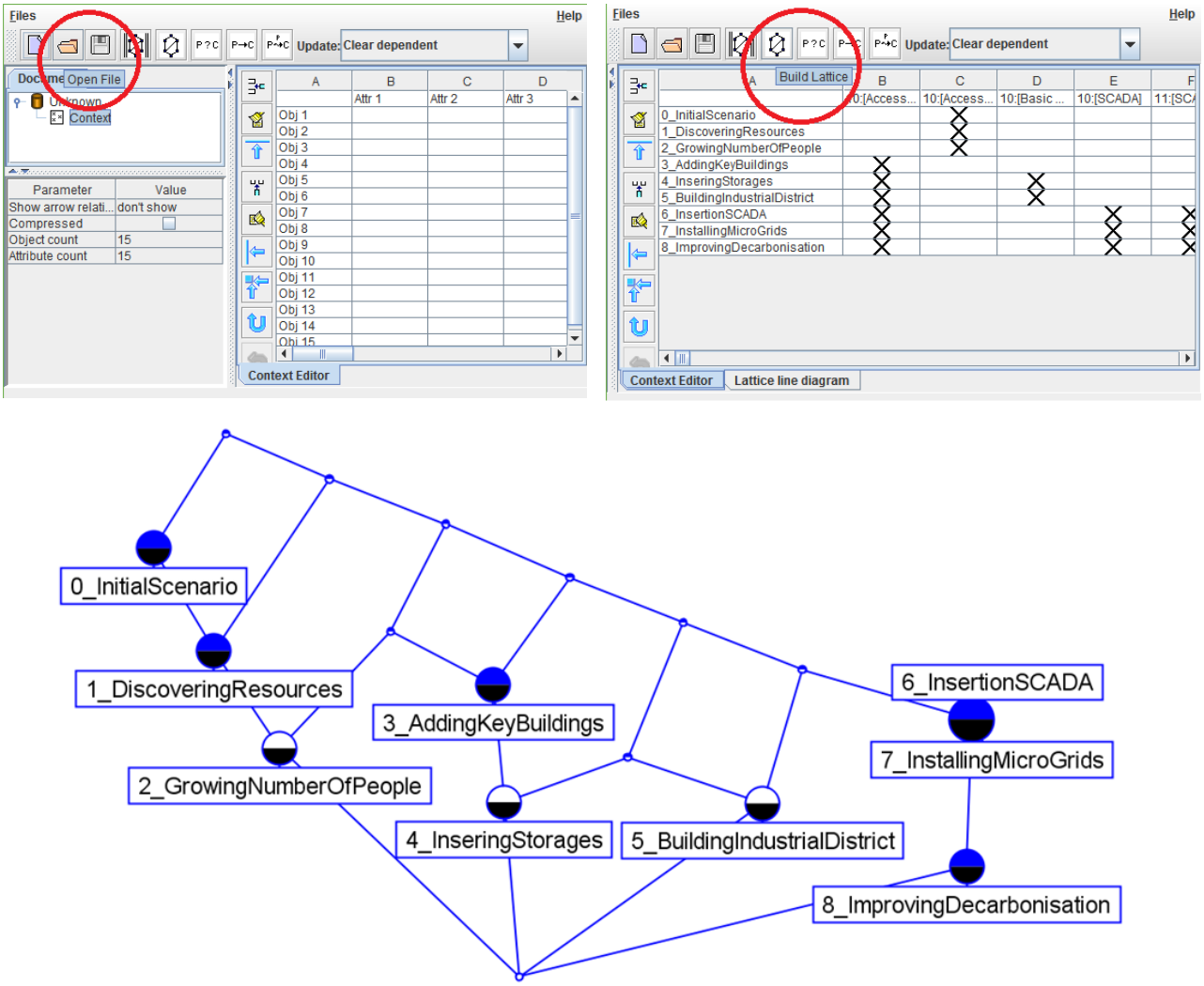


Figure 3-3: Steps to build a graphical lattice with *ConExp* FCA tool

Moreover, we can observe that a hierarchy is established between two or more scenarios if the evolution step only adds component (as for scenarios 0, 1, 2); otherwise, some threats can be removed, changing the base threat set and not simply extending it.

4 THE BAYESIANFAIR TOOL

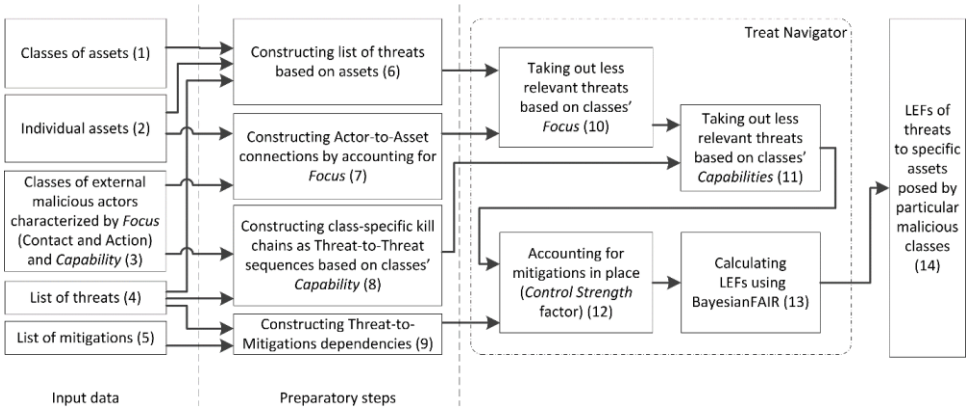
The outcome of the WP2 tool (a number of threat-component relations) needs to be structurally and traceably considered to concentrate on specific threats. This can be done by filtering out less relevant threat-component relations. Then, the tool users might providing a number of input values for calculating the Loss Event Frequencies (LEFs) of IRENE threats.

BayesianFAIR is a module that allows numerical threat assessment based on the states of the four FAIR factors [4], including *Contact, Action, Threat capability*, and *Control Strength*. The tool is used to assess LEFs of smart grid threats. The numerical outputs given by BayesianFAIR can help to further rank threats in the same category (e.g. High or Very High), which is an extension from the FAIR framework [4]. This will be helpful to prioritise threats to assign the constraint security resources, especially in cases where many threats are considered in the network. Moreover, BayesianFAIR allows the fuzzy inputs, instead of the common fixed enters like Very High, High, Medium, Low, Very Low (VH, H, M, L, VL). BayesianFAIR calculations help to highlight the impact of a single factor on the overall assessment, and from that, help to point out the most influential factor[2] that a system operator should focus on to build effective mitigation plans[3].

To calculate LEFs one should consider 'Threat Event Frequency' and 'Vulnerability'. The first construct includes *Contact* and *Action* factors of a particular threat. Threat Event Frequencies are thus constructed by relating probabilities of contacts between threat sources and the system, complemented by the attackers' incentive to engage (*Intent*). Vulnerability deals with *Control Strength* and *Threat Capability*. More information about LEFs is provided in D2.2 [3].

The steps to obtain LEFs of individual threats from the overall list of threats to a system can follow the steps described in D2.2 [3] and in [15]. These steps, named the Threat Navigator method (

Figure 4-1), takes a large set of input data and outputs the *LEF*s of relevant threats.



[2] The most influential factor (of the four input factors) for a threat is the factor that will decrease this threat's severity the most, given the same improvement on the input's security level [3].

[3] A detailed case study to illustrate that point can be found in [3].

Figure 4-1: Proposed *Threat Navigator* method.

The input data are the following: 1) "Classes of assets" are grid assets, 2) individual grid components reflect the categories and individual assets, 3) "Classes of external malicious actors" define which actors apply to individual threats, 4) "List of threats" represents the generic list of threats to be considered; it is later used to identify all threats for the given assets, 5) "List of mitigations" includes controls that can be implemented against the threats listed in D1.1 [1]. These data are pre-processed to construct 6) a list of threats to be refined, 7) *Actor-to-Asset* and 8) *Threat-to-Threat* connections, and 9) *Threats*-to-*Mitigations* links. These pre-processing steps take place within WP2 tool.

Subsequently, the constructed relations can be studied to remove threats less relevant to specific classes of attackers based on their *Focuses* (10) and *Capabilities* (11). Next, one should account for implemented mitigations (12) and finally calculates *LEF* for threats (13). The LEF calculation can be performed by using either FAIR tables (see p.17 of D2.2 [3]) or employing a fuzzy input (see p.46 of D2.2 [3]) for calculations. Either of these ways is performed in block 13 of the

Figure 4-1.

A way to concentrate on a particular threat can be as follows. Let us consider that the application of the WP2 tool resulted in a number of threats to the system that are linked to specific components. These threats correspond to different categories (for details please see Table 19 of D2.1 [8]). The AR "Achieve results" category (one of several categories outlined) includes the following threat events: Cause integrity loss, Obtain unauthorized access, and Obtain information.

The less relevant threats can be filtered out according to some assumptions about the focus and the capability of attackers. This approach can follow the suggested classification of attackers described in sub section 6.2 (Root-cause analysis of adversarial threats) of D2.2 [3]. Noteworthy, for natural disasters decisions about the context of the grid can be informed by the suggestions provided in section 10 of D2.2 [3].

As a result of this (or alternative and plausible) process the users can decide to further analyse one specific threat in connection to a number of components, e.g., the AR threat "Obtain unauthorized access". The users need to consider what values should be assigned to LEF constructs to enable LEF calculation. Some assumptions are connected to the properties of the city under analysis, for instance to the political and geographical localization of the city. Examples of the assumptions include (see p.70 of D2.2 [3]): 1. the city has an important strategic relevance and is consequently exposed to terrorism; and 2. the city is NOT in a seismic zone. Existing controls and vulnerability of individual component (e.g., the height of the electric equipment with respect to the expected flood high) should be also accounted for. The values of 'Contact', 'Action', 'Threat Capability', and 'Control Strength' can be used to calculate the overall LEF value of the treat to a component.

The BayesianFAIR tool is implemented in the Overall Grid Modelling (OGM) tool. The next section describes the OGM tool, along with the tool implementations of both OGM and BayesianFAIR.

5 THE OVERALL GRID MODELLING (OGM) TOOL

5.1 DESIGN OF THE OGM TOOL

In this case, an open OGM tool is developed. The tool development is based on the agile process, where the processes of specification, design, implementation and testing are concurrent, and as an iterative approach. The open tool is developed in a series of increments where the user will evaluate each increment and make proposals for later increments. GUI are usually developed using an interactive development system. The functional of the OGM tool is shown in Figure 5-1 below. A**s** there is no detailed specification of the tool development and therefore the design documentation is minimized. Such agile-based tool allows the concurrency (non-sequential based) of inputs to be added from previous IRENE WPs.

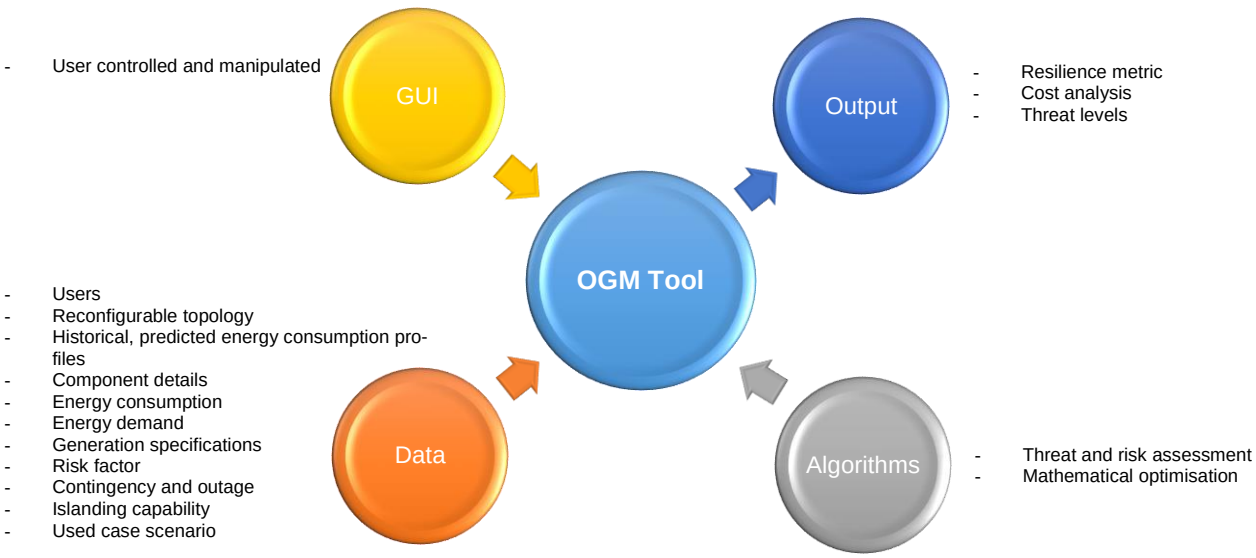


Figure 5-1: The OGM functional open modelling tool.

Based on Figure 5-1, the OGM tool consists of a *GUI* that will facilitate ranges of services based on the requirements by Stakeholders as defined in WP1. The GUI is also designed that is user-friendly, easily controllable and manipulated. Then, *Data* should be provided by users or based on methodologies and policies as developed within WP2-WP3, depending the degree of knowledge that the user wishes to provide into the tool. Once information is gathered sufficiently, prediction and threat and security assessment are performed using the formulated *Algorithms* within WP2 – WP3. The *Output* demonstrates the analysis of results of the overall performance of the network (i.e., resilience metric, cost savings and threat impacts). If the user wishes to implement several used case scenario (or what-if analysis) in order to observe the output changes directly (i.e. the addition/removal of particular consumption profiles, critical loads, generators, storages, renewables, topological changes, outage simulation and islanding analysis), this can be added/removed with high flexibility,

without having to remove and reload the previously modelled information. Henceforth, the user does not need to redefine the network topology and all the components from the beginning of the process. This demonstrates the added agility concept of the tool by allowing concurrency in updating new trends of input information provided by the user using the existing model.

As the OGM tool is aimed for Stakeholders (Municipal authority planner, DNO, Developers, Critical Infrastructure Operator, Business and Citizen Representative) with various technical/conceptual background, the tool is designed that is easily-interpretable for fellow Stakeholders, without incorporating complex power-flow model and analysis. The tool is designed through the literature research obtained from WP5 [4], where the tool is expected to provide the standard functionalities and also the functionality requirements from WP1. In addition, new features such as Resilience analysis, security and threat assessment (BayesianFAIR) as developed within WP2-WP3 are integrated into the toolset, along with added flexibility to adjust the scalability of the tool for simulation of bigger urban city wide area.

5.2 OGM TOOL INTERFACE

5.2.1 GUI Input window

At first instance, a network topology presenting the integration of the grid and Microgrid-connected is necessary. At outlined in IRENE D3.1 [3], the IEEE-inspired bus tree is used as the fundamental representation of grid architectural topology. Such topology is implemented in this case as the main user interface for grid modelling and simulation, where the user can manipulate the whole integration of the grid (without altering the nodes/buses as constructed).

Overall, the GUI is developed using IntelliJ IDEA, the Java IDE software. The GUI of the tool at the first instance is shown in Figure 5-2. Figure 5-2 shows the clear input window of GUI as before the user loads the network topology and configure the component specifications.

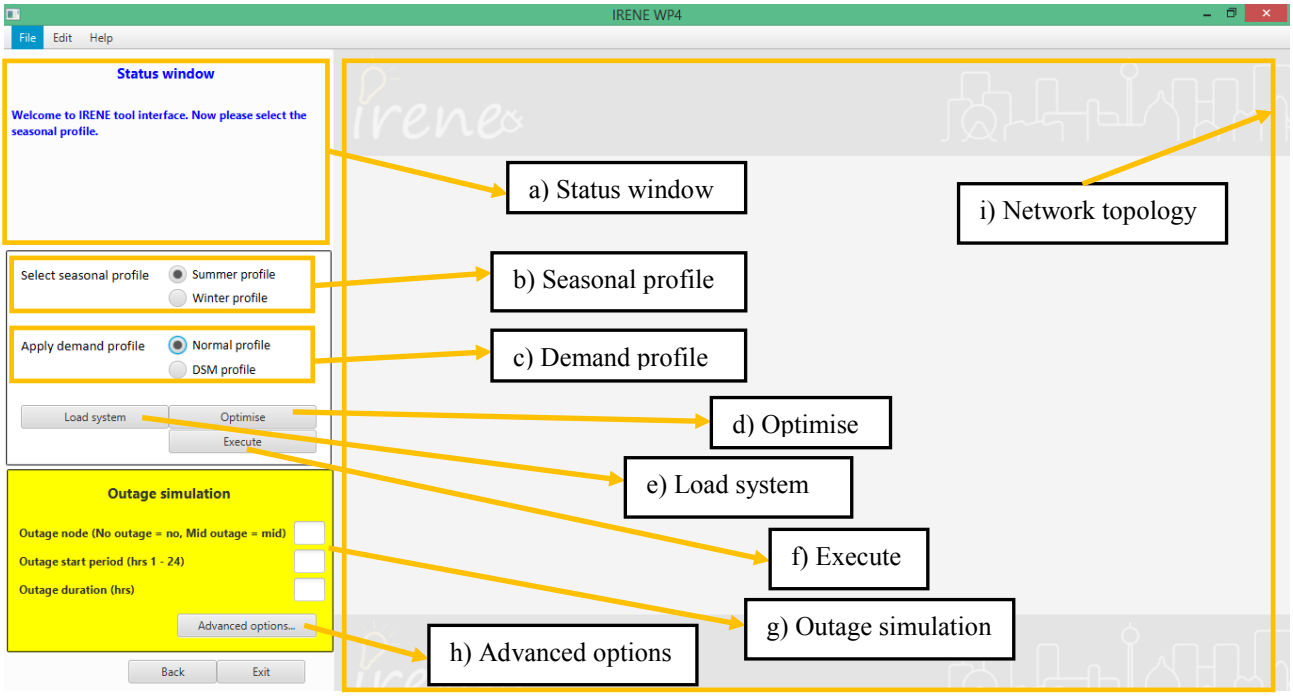


Figure 5-2: The GUI input window of OGM tool.

Based on Figure 4-3, the input GUI of the tool consists of:

a) **Status window** – The status window explaining the progress of the overall simulation configured by the user;
b) **Seasonal profile –** The toggle selection of summer or winter seasonal demand profile;
c) **Demand profile –** The toggle selection of demand profile either the demand data adopted from the public domain, or the demand profile through the AIT's optimised profile with DSM capability;
d) **Optimise –** The action button to perform the optimisation simulation;
e) **Load system –** The action button to load the network topology;
f) **Execute –** The action button to perform the threat and criticality assessment;
g) **Outage simulation –** The outage window for simulation of outage events;
h) **Advanced options –** The advanced features and additional input options for advanced users;
i) **Network topology –** The input display for network topology to be loaded.

5.2.2 Network topology

In the input window, the user in initial has to select the type of seasonal and demand profile as before loading the network topology. By default, the summer seasonal and normal type of demand profile will be selected if the user do not select any of the options. By loading the network topology (through clicking on 'Load system' button), the IEEE-inspired tree network topology is loaded as shown in Figure 5-3.

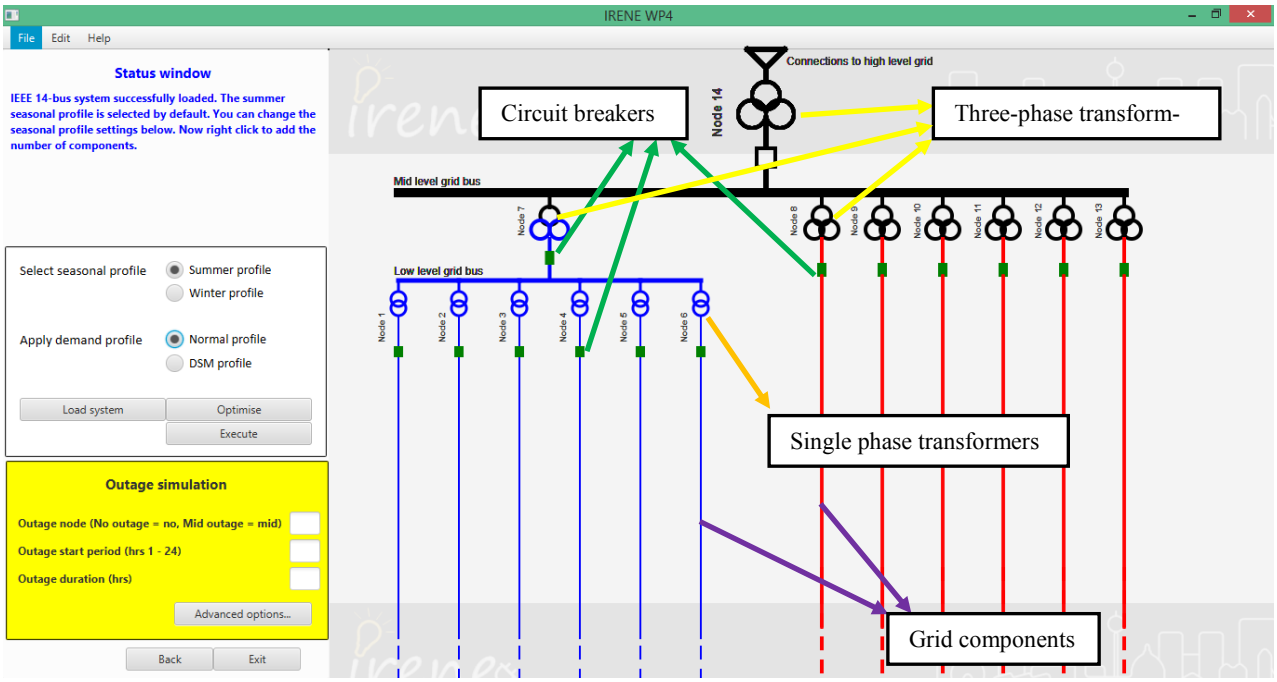


Figure 5-3: IRENE GUI input of IEEE 14 network topology.

5.2.3 Adding components

In the next step, in order to insert the component respective to each node, the user has to identify the number of component to be inserted in each component, as shown in Figure 5-4.

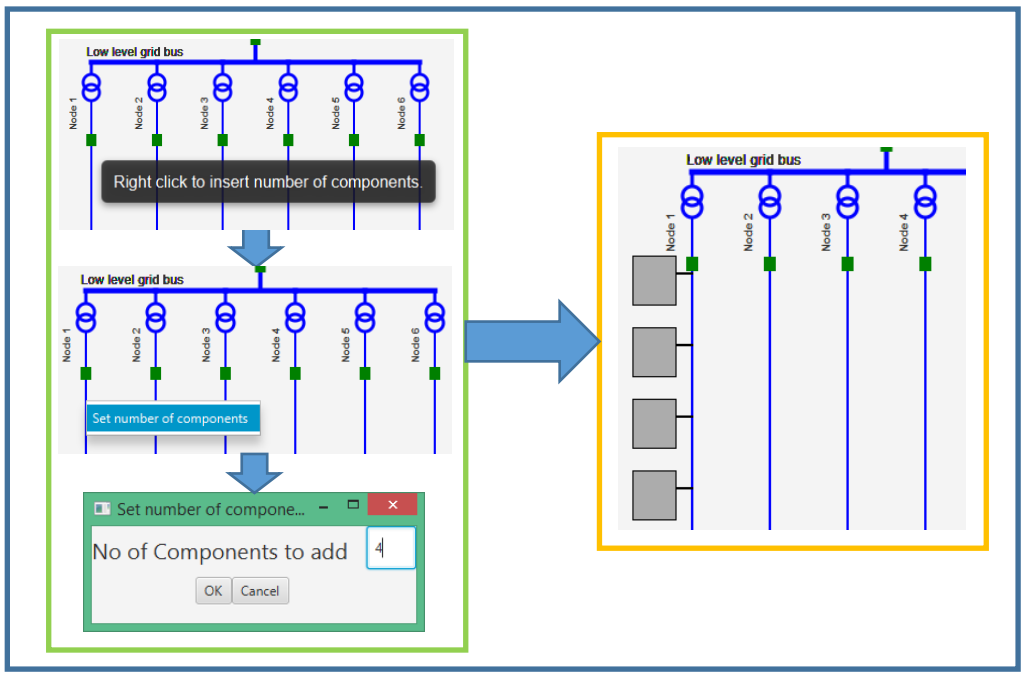


Figure 5-4: IRENE GUI to insert number of components.

Based on Figure 5-4, the user will be prompted to inserted the number of components to be added into the respective node (as shown in the left panel of Figure 5-4), right after the number of component is entered by the user, the component will be added to the input network topology model, as shown in the right panel of Figure 5-4.

When the number of compoenents are successfully added, the next step involves the addition of grid components (the generating and consuming components). Figure 5-5 demonstrates an example of adding the 'Household' profile component into the input model.

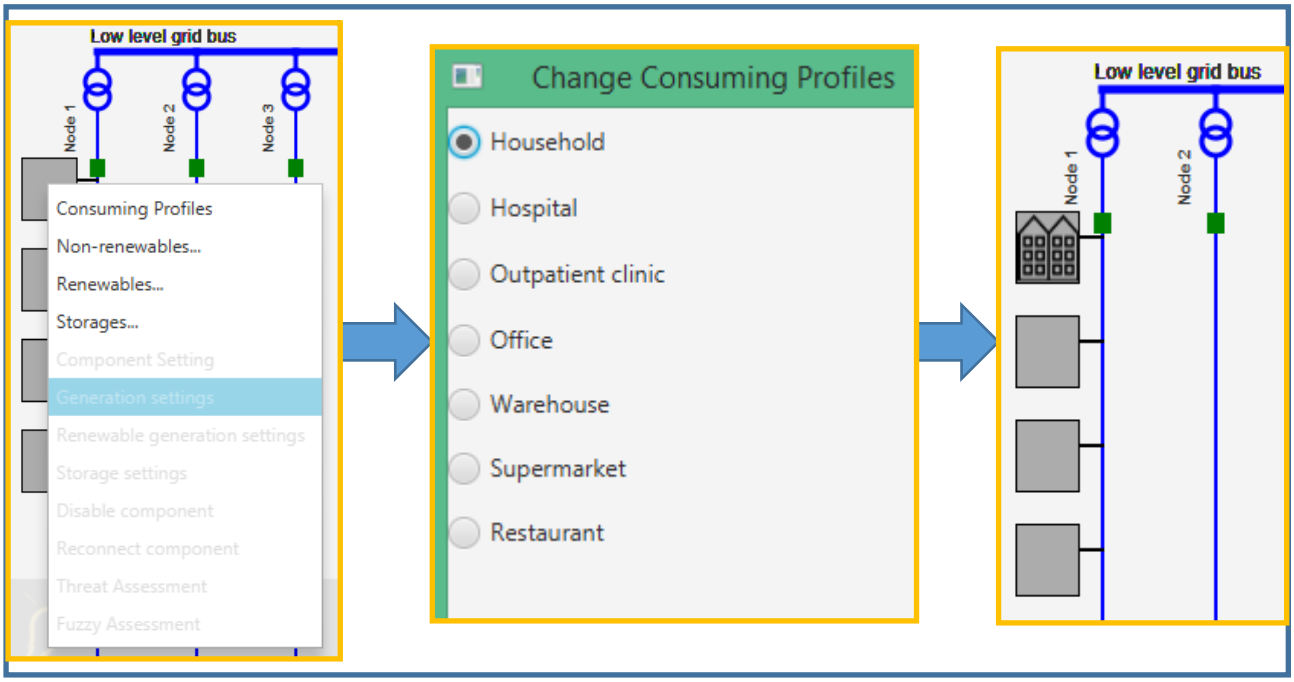


Figure 5-5: IRENE GUI to insert grid component.

5.2.4 Component setting configurations

In order to configure the component setting for the 'Household' profile in this example, such functionality is available in the input model as demonstrated in Figure 5-6. Based on Figure 5-6, in the right panel, the number of attributes represent the population/frequency of the selected consuming profile. The default attribute is set as 15,000 and if the user wishes to apply such setting then no further modifications are needed, as well as the other settings relating to the criticality analysis and assessment. Those settings can be changed as if the user wishes to apply their own knowledge for the component settings.

Similarly, Figure 5-7 illustrates the configuration of the generation settings for generators, renewables and energy storages. Similar to the case of consuming-component settings, the default generations specifications are provided by default and the user can apply the specifications provided for further simulations. Those settings can be modified based on users' knowledge on generation settings, and also to fit the scalability (for instance, the generating capacity fit the overall demand, when a new network topology is added with demand variations compared to the ordinary predefined grid settings) of the grid architecture.

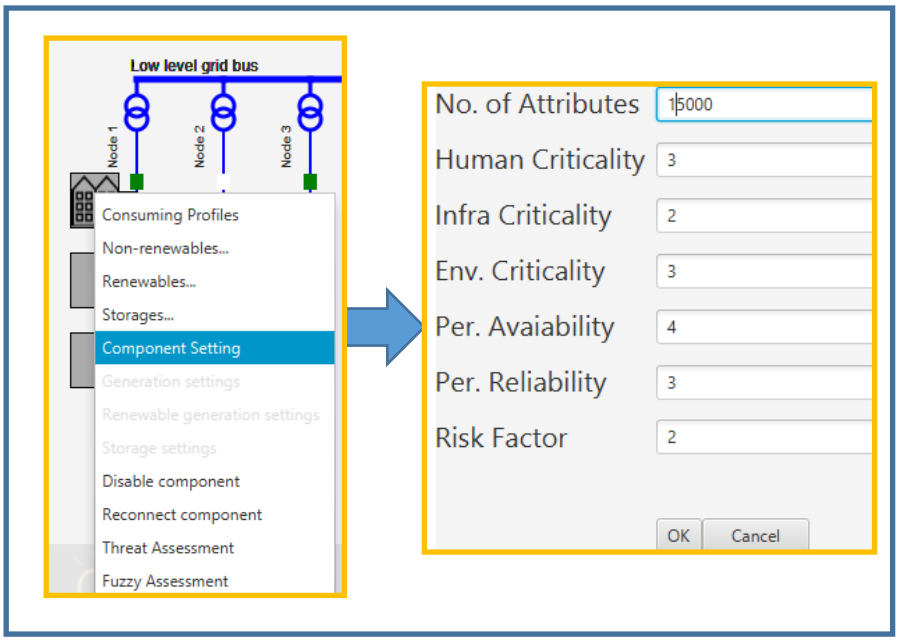


Figure 5-6: IRENE GUI to configure the component settings.

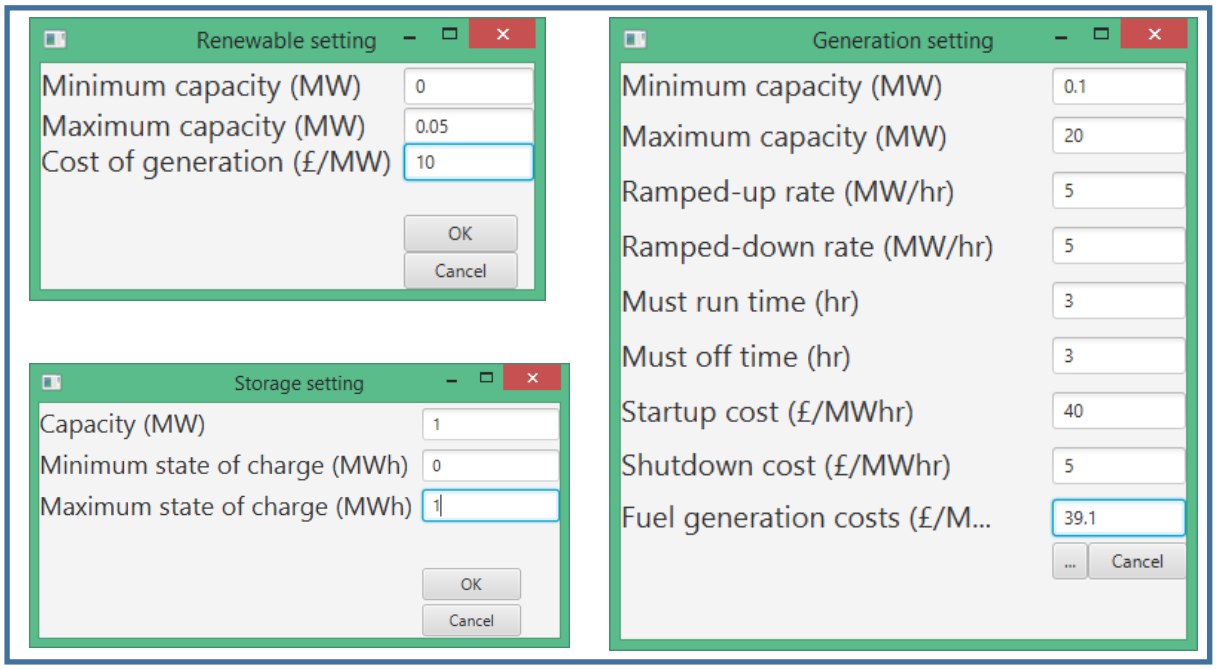


Figure 5-7: IRENE GUI to configure the generation, renewable and storage settings.

5.2.5 Outage simulation

In default configuration, if no outage is perturbed in the input model, the tool will apply the normal mode of simulation without outage. For the case of single node of failure/branch in a grid (N-1 compliance), the configuration is adjusted as illustrated in Figure 5-8, where the outage is perturbed in Node 2 by opening up the Circuit Breaker in Node 2. The outage period and duration are set in the input outage simulation window, where outage start period at 900 morning is triggered with six hours of outage duration. Additional display window will inform the user on the outage simulation that is triggered in a particular node.

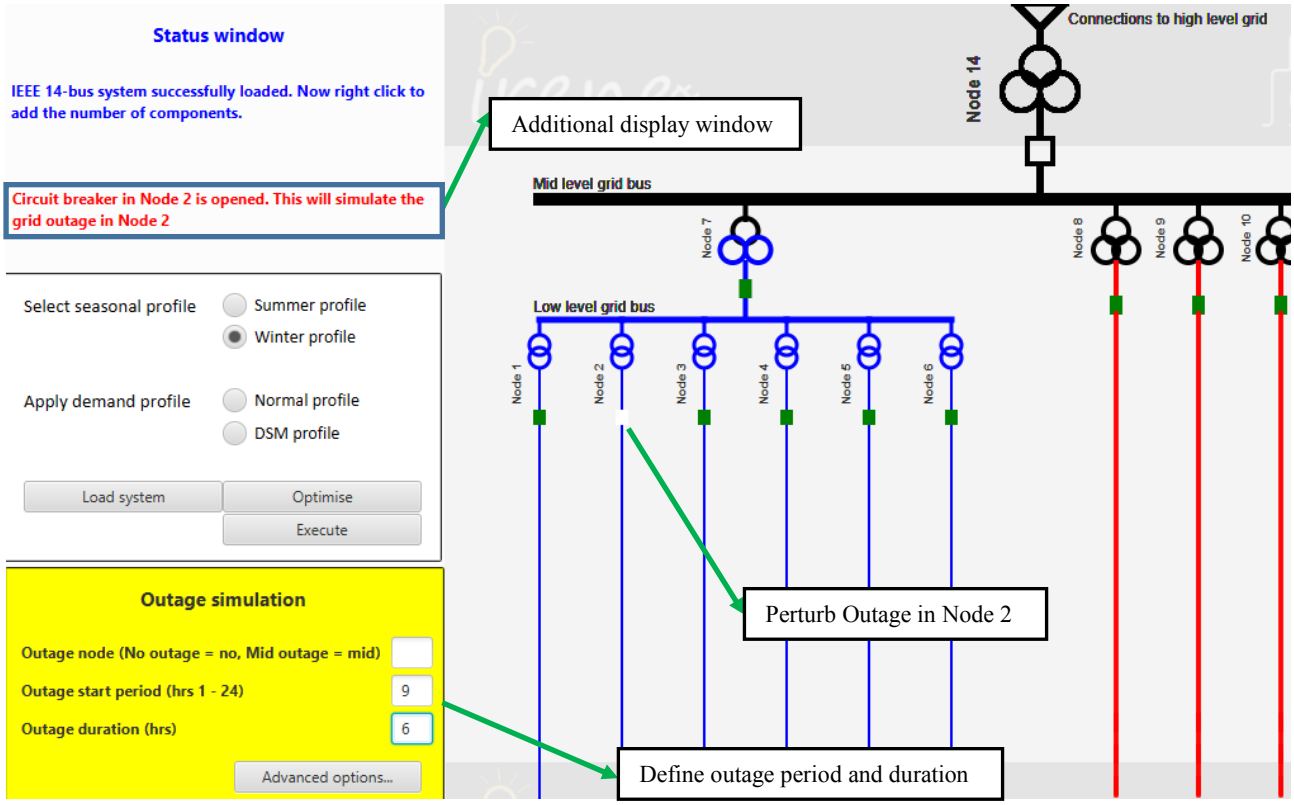


Figure 5-8: IRENE GUI to configure the outage node.

Another option is to disable the operation of local generations in individual nodes. Figure 5-9 illustrates such example, where a Generator is disabled in order to perform an outage simulation for the generator.

The GUI demonstration of the output simulation window is postponed to the next section.

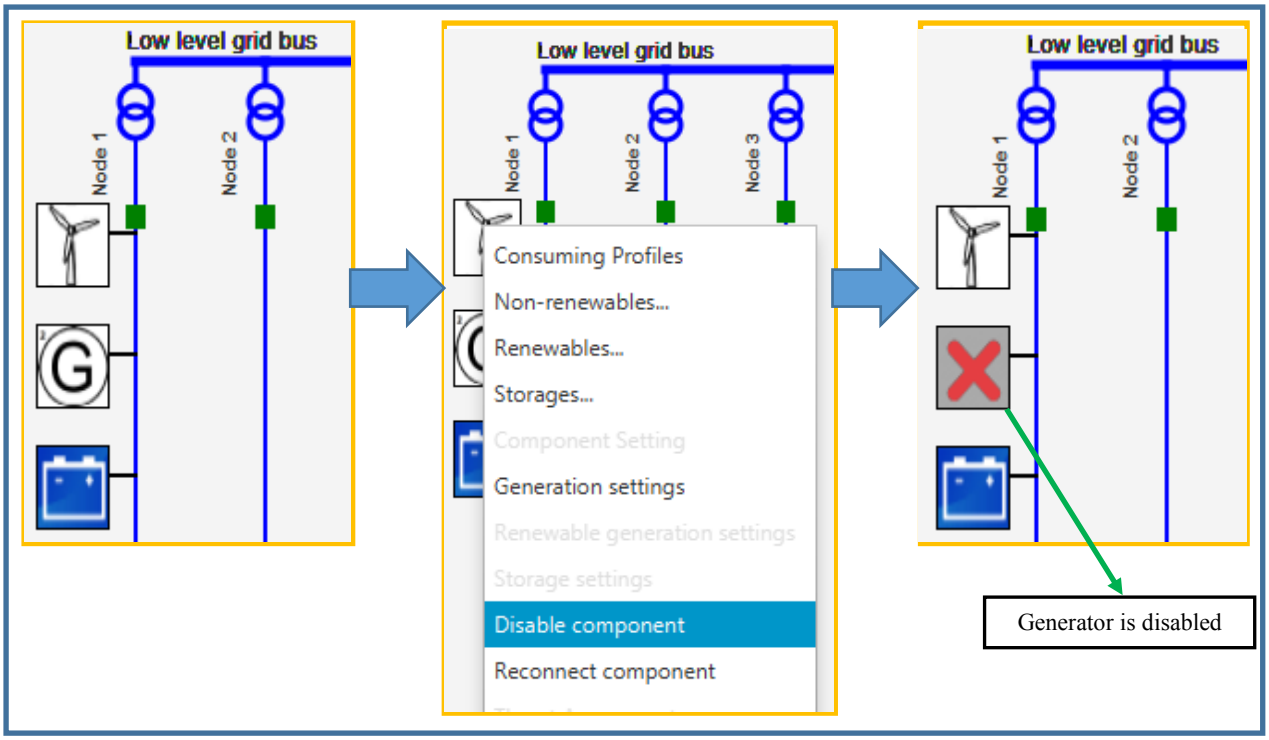


Figure 5-9: IRENE GUI that disables a component.

5.3 OGM TOOL ANALYSIS EXAMPLE

This section explains the implementation of the security assessment, prediction and optimisation model developed in WP2 and WP3 in order to perform the simulation for each used case scenarios.

Figure 5-10 shows the example of toolset with components and settings added into the network topology. Such example is adopted from the earlier defined samples of consumer, generations and load distributions of the grid in WP3 [3]. From

Figure 5-10, Nodes 1-6 consist of the main distributed generation buses and the loads, while buses 1-8 contain the energy storages for reserving purposes, Nodes 12 – 13 contain empty generation and load sources. Finally, Node 14 is the connection to the main grid. The distribution of load profiles in this case is not intended to include the profiles of commercial services (e.g. hospitals, offices) and domestic households within the same bus. However, the variety of commercial services within a same bus is still possible. Additionally, most of the commercial services are connected with their own substation due to the huge amount of loads required.

In this example, outage event is triggered in Node 3 at 900 in the morning, with six hours of outage duration.

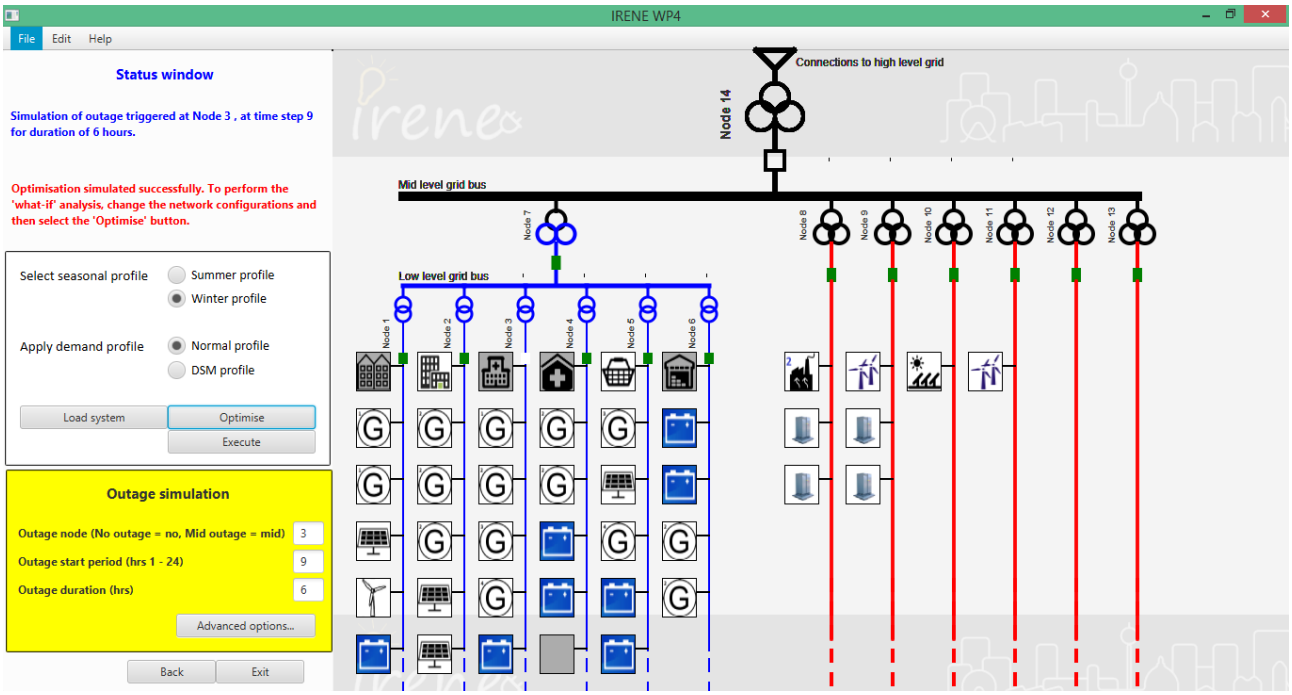


Figure 5-10: An example of the OGM tool with added components and settings into the network topology

The numerical optimisation algorithm is initially is solved using the Matlab software in WP3. The dual-simplex algorithm is applied for such LP problem of the grid optimisation. However, as it is intended to employ Java-based software environment for open-based modelling tool, the lp_solve 5.5.2.3 [17] is applied as the library file for Java that is called to perform the optimisation algorithm in WP4. The lp_solve is basically a Mixed Integer Linear Programming (MILP) solver with no limitation on model size (variable size), and also, it is absolutely free and with sources.

The configuration as defined in Figure 5-10 is simulated and after the simulation, the output window will demonstrate the simulation results. Total of four output windows tabs are created to display different types of output.

5.3.1 Output window

Figure 5-11 shows the output window that demonstrates the distribution of consumption profiles during the winter season. Such output enables the user to gain better understanding the trend of consumption pattern, and also to determine the peak demand periods.

Figure 5-11: An output window showing the distribution of consumption profiles (24-hr) cycle in winter season.

Figure 5-12 shows an another output window displaying the distributions of consumers and generators within network nodes as defined by the user in the input model. Additionally, the bottom panel presents the overall resilience metric of the grid during the outage period, and also as well as the cost savings for the simulation runs (having to add/ remove local generations, dispatching of energy storages into the grid system).

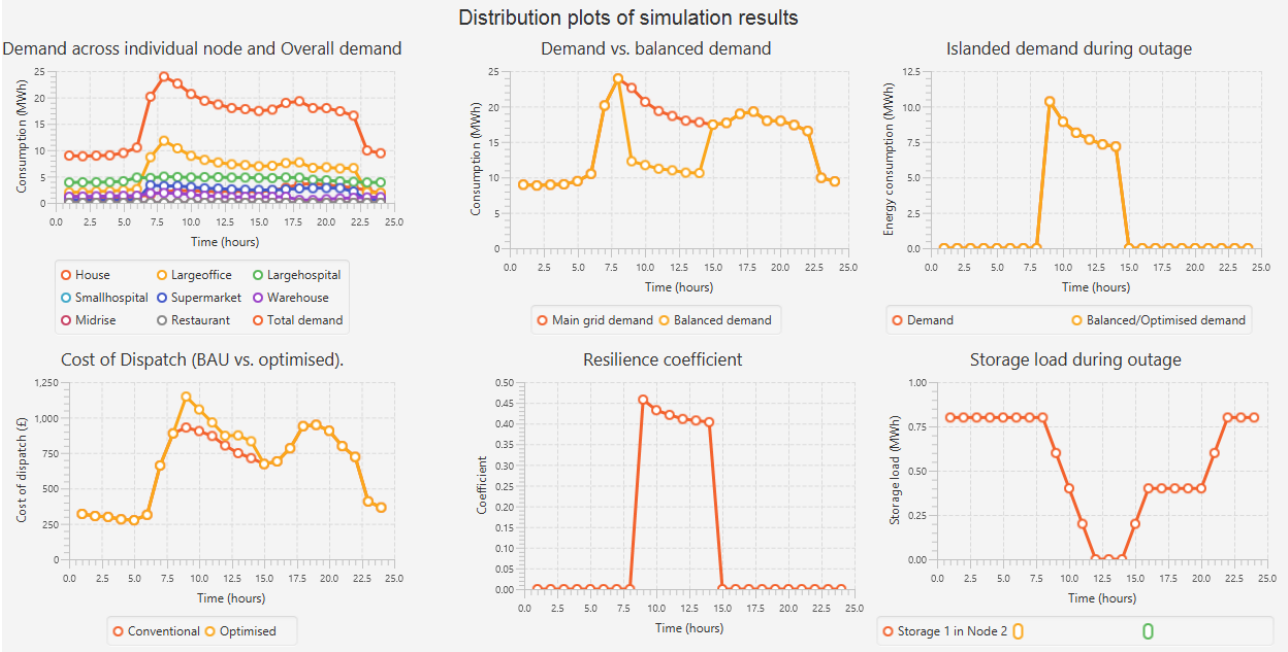


Figure **5-13** shows the output window that demonstrates the distribution plots of simulated results. Such distribution plots employed earlier concept of plotted results from WP3, where the top left presents the plot of individual consumer demand profile and the overall demand profile across the grid, the top middle demonstrates the main grid and the balanced/optimised demand, the top right panel displays the balanced/optimised demand which is isolated from the main during during the outage operation. On the other hand, the bottom left displays the business-as-usual versus the optimized solutions of costs, the bottom middle panel demonstrates the resilienc coefficient with respect to the time of simulation, and lastly, the bottom right panel shows the energy storage load distributions during the outage operation.

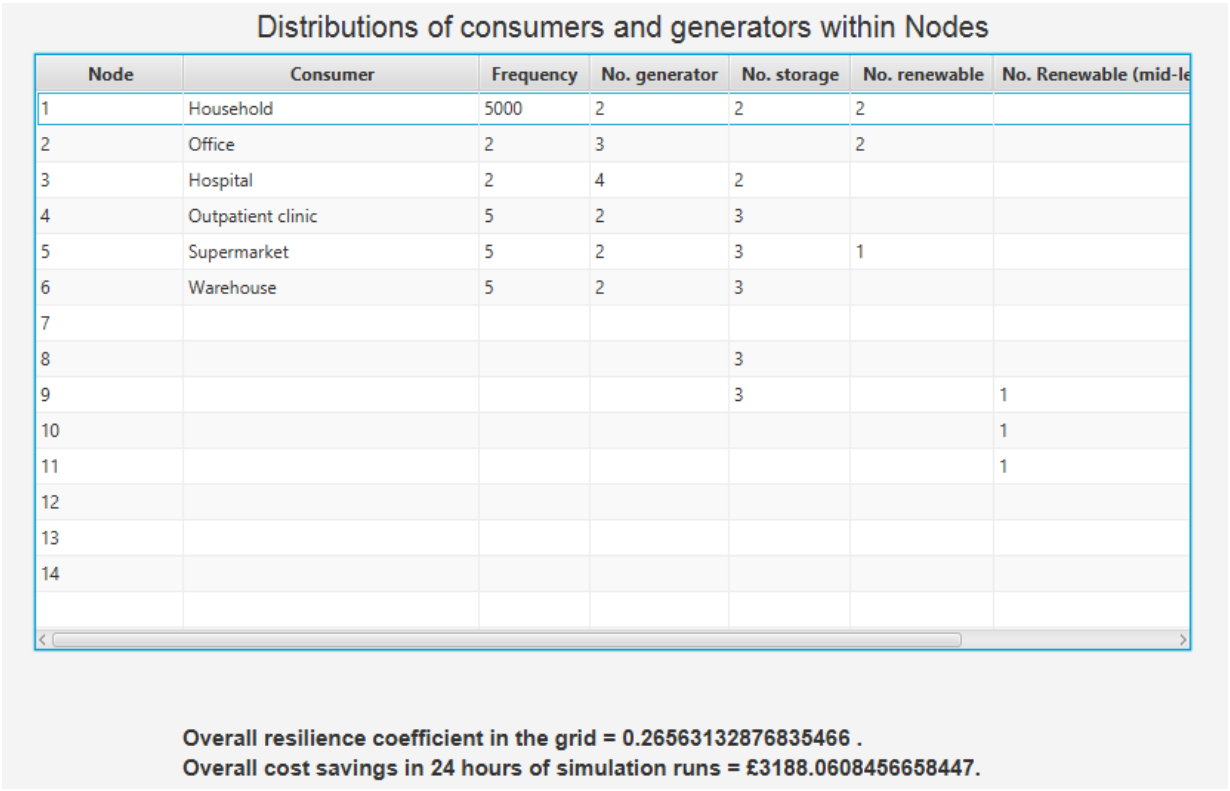


Figure 5-12: An output window showing the distribution of consumption profiles (24-hr) cycle in winter season.

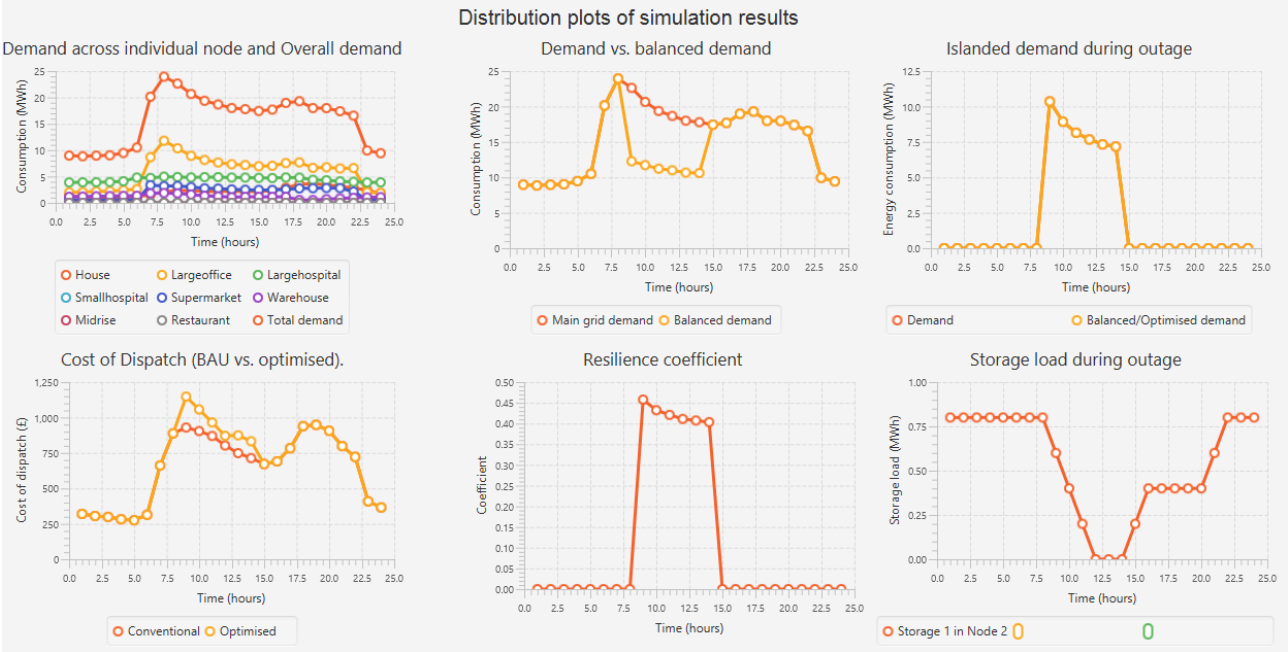


Figure 5-13: An output window showing the distribution plot of simulation results.

Figure 5-14 shows the results of EnKF assimilation and forecast with the number of ensembles used = 1000 in order to forecast the day ahead electricity demands. The result of EnKF forecast in this case performs relatively well compared to the actual demand data, where in this case the ensemble size of 1000 is sufficiently enough to produce the required demand forecast, with the root mean squared error (RMSE) of 0.275. For different EnKF realisations, please refer to the results of different EnKF realisations in D3.1 of WP3 [5].

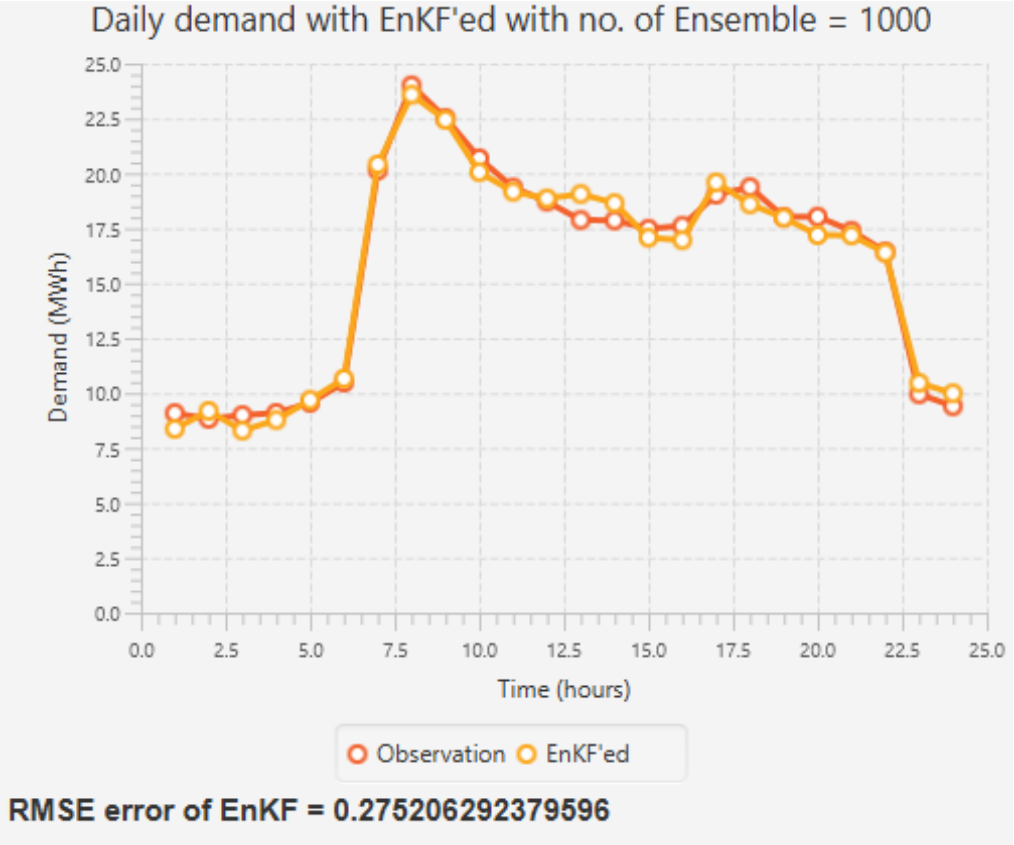


Figure 5-14: An output window showing the result of EnKF forecasts.

5.3.2 Threat assessment

This section illustrates the interface for users to provide input for LEF calculations. In the current software implementation, two ways are possible: 1. By using values like (Very Low/Low/Medium/High/Very High) and 2. By providing input in a more fuzzy way, e.g., [40%VL 15% L 15%M 15%H 15%VH]. More information on it how to form a fuzzy input is provided in section 7 of D2.2 [3].

After several LEF values for different components (and for different threats, if needed) are calculated, these values can be compared to each other. The goal of this step is to relate individual threat-component relations to each other. This can help in prioritizing how to spend limited resources to improve some controls.

The threat assessment is implemented in the OGM tool, where the user selects the component, apply the tuning and examining the overall threat level in the component. Figure 5-15 and

Figure 5-16 show the result of threat and fuzzy assessment for the component household correspondingly.

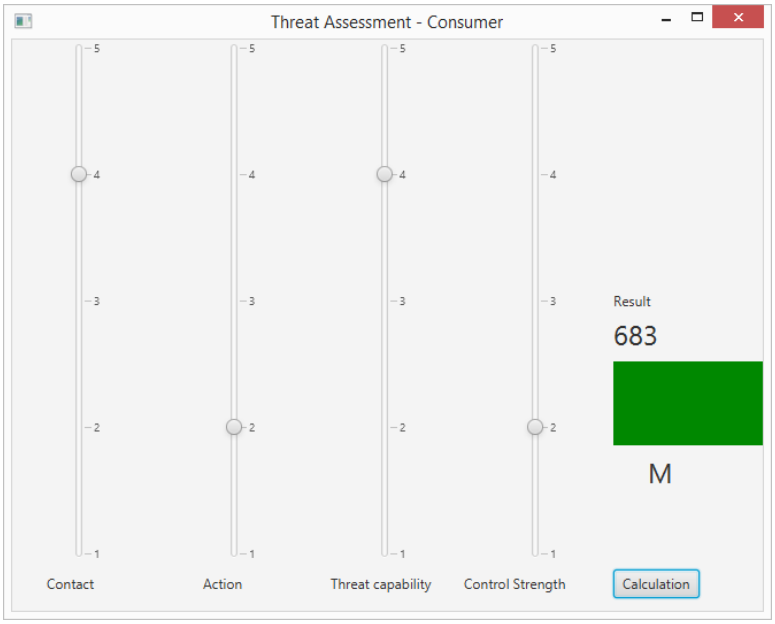


Figure 5-15: The threat assessment for component 'household'.

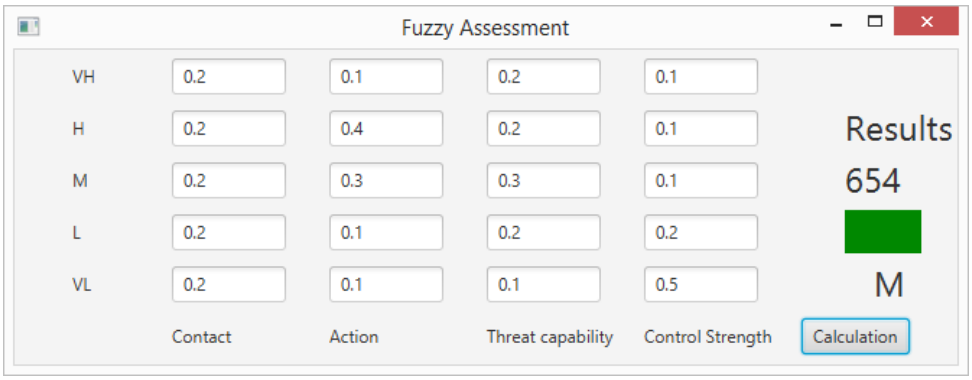


Figure 5-16: Fuzzy assessment for component 'household'.

Table 5-1 shows the numerical results of traditional FAIR, SE through BayesianFAIR, and the overall ranking of the components.

Table 5-1: Numerical results of BayesianFAIR and FAIR method on some components.

Component	Input state	FAIR	BayesianFAIR	Rank (overall)
Houses	[H,L,H,L]	M	683	7
Offices	[H,VH,H,M]	VH	863	2
Supermarkets	[H,M,L,VL]	VH	825	4
Outpatient clinics	[H,VH,VH,M]	VH	865	1
Generators (DGs)	[H,L,M,H]	M	656	8
Carbon plants	[H,H,VH,M]	M	844	3
Battery storages	[H,L,VL,M]	M	583	9
Photovoltaic stations	[H,M,M,M]	H	757	5
Wind-farms	[H,M,M,H]	H	754	6

Based on **Fehler! Verweisquelle konnte nicht gefunden werden.**, the Outpatient clinics has the highest threat severity due to *High* probability of large scale damages (i.e., power failure of the connected power lines to Outpatient clinic), and the Low resistance to the damages. Henceforth, such SE value informs the city planner that actions need to be taken in order to reduce the SE level of Outpatient clinics, e.g., installing backup-generation to supply the emergency power during the earthquake. The threat severity for all components can be updated easily if users wish to assess the updated input states. Such allows the Stakeholder to perform further mitigation plans in order to reduce the threat severity of the grid components.

5.3.3 What if analysis

The user can easily readjust the settings (e.g. outage node, removing/adding generators, pro-long/reduced the outage duration, and set a new outage period) based on the input model and per-form the optimisation instantaneously. There is no need to reconfigure the overall input network topology. This is to enhance the functionality of the tool by enabling the sensitivity analysis (what-is analysis) to be performed that allow the user to compare the overall performance and resilience of the grid with respective to modified configurations within the same network topology.

6 THE MICROGRID EVALUATION TOOL (MGE)

In order to use this tool, it is assumed that the studied smart grid is composed at its periphery of microgrids - local grids with loads between a few hundreds kW and 10 MW. Microgrid architectures have been proposed for advanced control, DER control and optimization.
The MGE Tool is an event based simulation of the interacting Customer Energy Management System (CEMS) controllers and the microgrid (MG) controller. The load models, load predictions, flexibility are updated and the optimization models produce new local control actions, the MG controller creates new setpoints for the respective building controllers, etc.
The control system covers only the microgrid, see Figure 6-1, consists of the microgrid (MG) controller and a number of customer controllers (CEMS) associated mostly to each of the buildings in the microgrid. In contrast to the passive control of distribution grids conceived at planning time, in the current approach controllers use flexibility, demand management and scheduling to cope with changes in the power supply, caused by an outage.
For the realisation of the control loop, the Model Predictive Control (MPC) technique is used, meaning that the power consumption (and generation) is predicted for a certain time horizon (e.g. six hours), however the actuation is performed only for the next period.
The MPC mechanism is combined with a novel exchange of flexibility information. Energy flexibility models exist in this work for HVAC (heating, ventilation, air conditioning), electric vehicle charging and battery storage. Each CEMS controller aggregates the flexibility of its assets and reports the resulted profile (during the time horizon) together with the planned consumption profile. The latter is the result of an optimization step, taking into consideration local goals, MG setpoints and constraints from all the local assets.
The MG controller reads the latest flexibility and consumption plans from the CEMS and computes updated setpoints (for the whole time horizon). In case the CEMS proposed consumption is too high, it sheds certain demands within their flexibility limits.

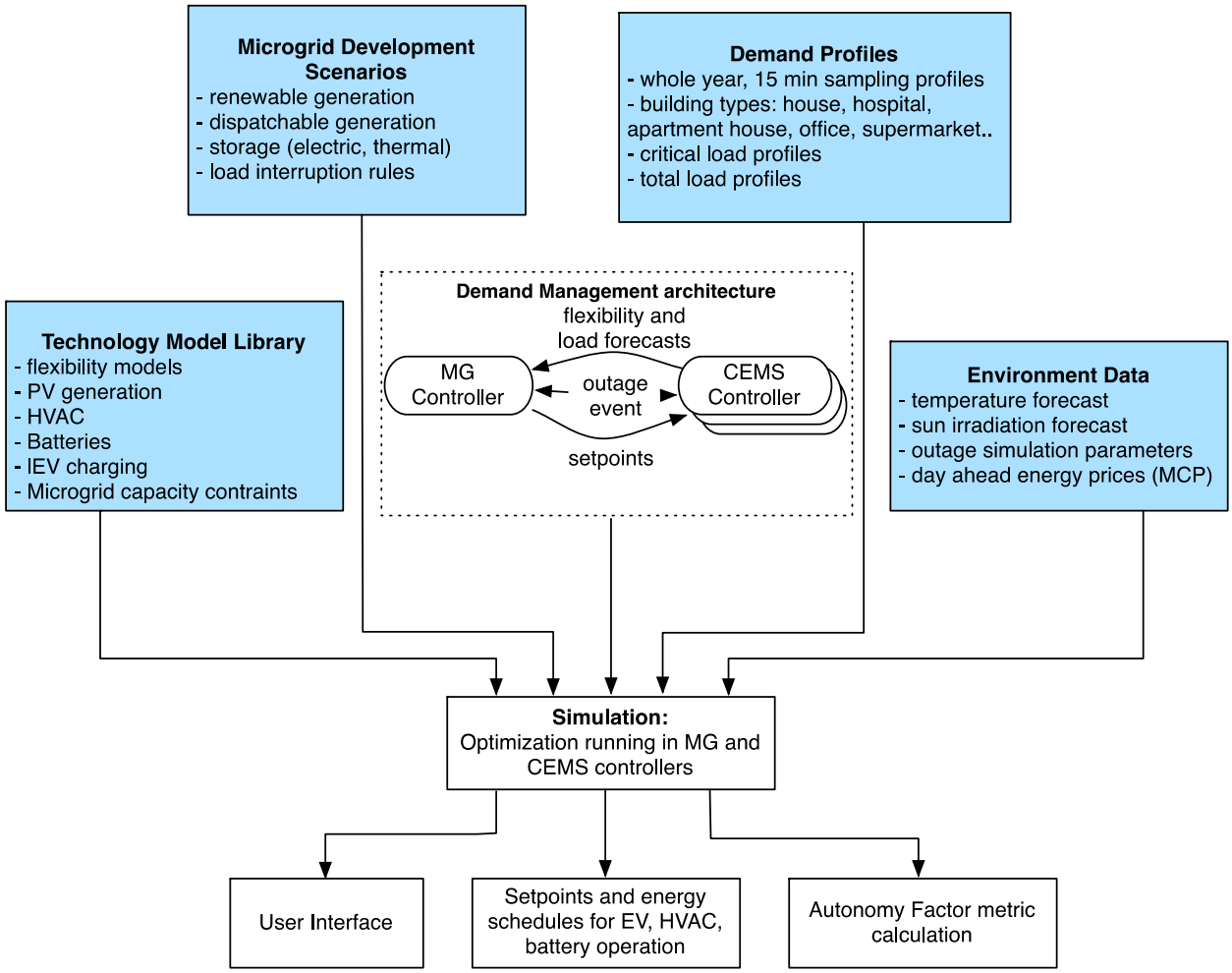


Figure 6-1 Overview of Microgrid Tool operation

6.1 POWER AND ENERGY FLEXIBILITY

Demand Management by using only the historical load profiles is not possible. Moreover, data on flexible loads and energy storage devices such as EV charging, home batteries is not available. Therefore, we have developed models that predict the consumption and calculate the flexibility information. Power flexibility (minimum and maximum power in each time period) and energy flexibility (minimum and maximum) models are provided for the following appliance types:

- HVAC (heating, ventilation air conditioning)
- EV charging
- (home) battery

In Figure 6-2 the EMS controller that solves periodically an optimization problem, becomes information from the different models and static load profiles. The results are different controls x, y, z for the local flexible loads and PV generation as well as new predicted consumption and flexibility.

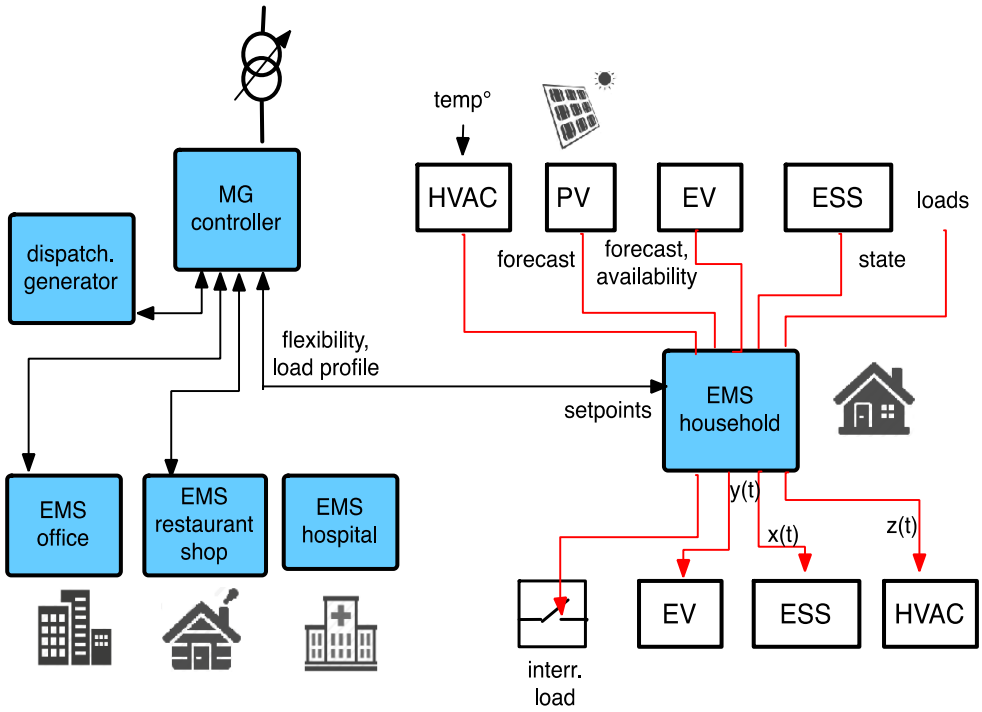


Figure 6-2 Building model with its inputs and outputs

6.2 DEMAND CHARACTERIZATION

For the purpose of planning the consumption of a microgrid, both in normal operation mode and during an emergency (external outage), load profiles and their aggregation are the main source of information.

The EIA (Energy Information Administration) provides high qualitative annual consumption data on an hourly basis, for different climatic regions in the US. The Chicago area has been selected, as it seems to be most likely to northern Europe.
Fortunately, the consumption data of residential and commercial buildings in has been de-aggregated in the categories ventilation, cooling, heating, lights and equipment. To these categories we added model-based loads such as EV charging, home battery storage, as well as PV generation. We focused on the summer months, in which air conditioning operates, the thermostat limits were set to 22-25 degrees Celsius. The HVAC models have been calibrated accordingly.

The demand during an outage, called critical demand is based on the selection in advance of the type of load that has to be maintained. The rest is interruptible load and will be discontinued during the outage. This type of configuration has to be done for each building type.
Interruptible loads are disconnected during an outage. Examples of interruptible loads in the household are: loads in the kitchen, entertainment, washing machine, vacuum cleaner, air conditioning, EV charging.

In the EIA dataset and also throughout Europe, (depending on building type and climate region), heating and water heating is often done with natural gas. Therefore, the visible electricity consumption during the summer due to air conditioning is higher than in winter.
We illustrate in Figure 6-3 the obtained profiles (including flexible HVAC, EV load and PV generation during two days of normal operation for residential house and a small office

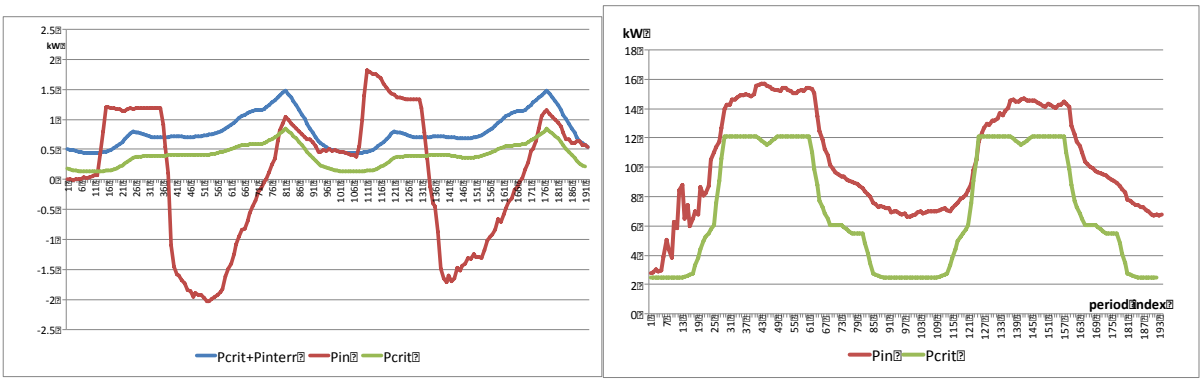


Figure 6-3: Total load (including PV generation) interruptible and critical load for residential house (left) and small office (right).

The "smartness" of a microgrid could materialize in a number of rules that are activated, once the outage event is received by the controllers in the microgrid. The rules can be more restrictive or more relaxed, depending on the energy balance, i.e. the amount of dispatchable generation available and the societal needs in the different building types. We used following rules for the outage mode:

- economic, price dependent criteria are disabled in the CEMS optimization, the load profiles still must follow the setpoints and keep the strict balance between supply and demand
- Shedding the PV generation is not allowed, the PV output is maximized.
- interruptible loads are disconnected
- the air conditioning/heating may be switched off in certain buildings to save energy. In any case thermostat limits are relaxed to increase flexibility.
- EV charging is either disabled or may use only local renewable energy.

6.3 MGE SIMULATION

A tool prototype has been built in Java. Extensive configuration is available:

- Main properties: simulation start date and time, outage start and end time
- profilemaster files : lists the buildings in the microgrid and their types (includes charging stations, external batteries)
- Building type profiles: load profile used, thermodynamic parameters (needed for HVAC model), battery and PV switches and parameters.
- Environment profiles: sun irradiation and outside temperature
- Energy prices: day ahead market clearing prices (hourly)

The output is available both via a user interface and through extensive logging and filtering. The latter can focus on charging profile of a certain EV, temperature in a certain building or state of charge of a certain battery.

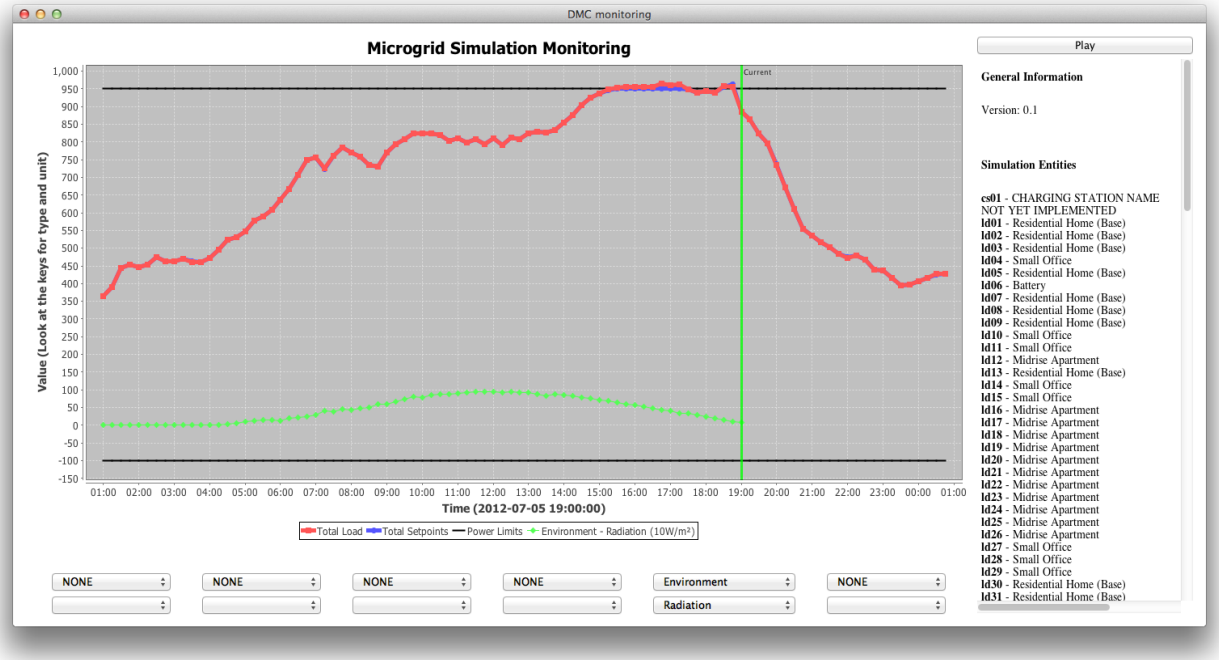


Figure 6-4: MGE Tool output: total load of the microgrid, total setpoint value (blue line), MG load limit (black).

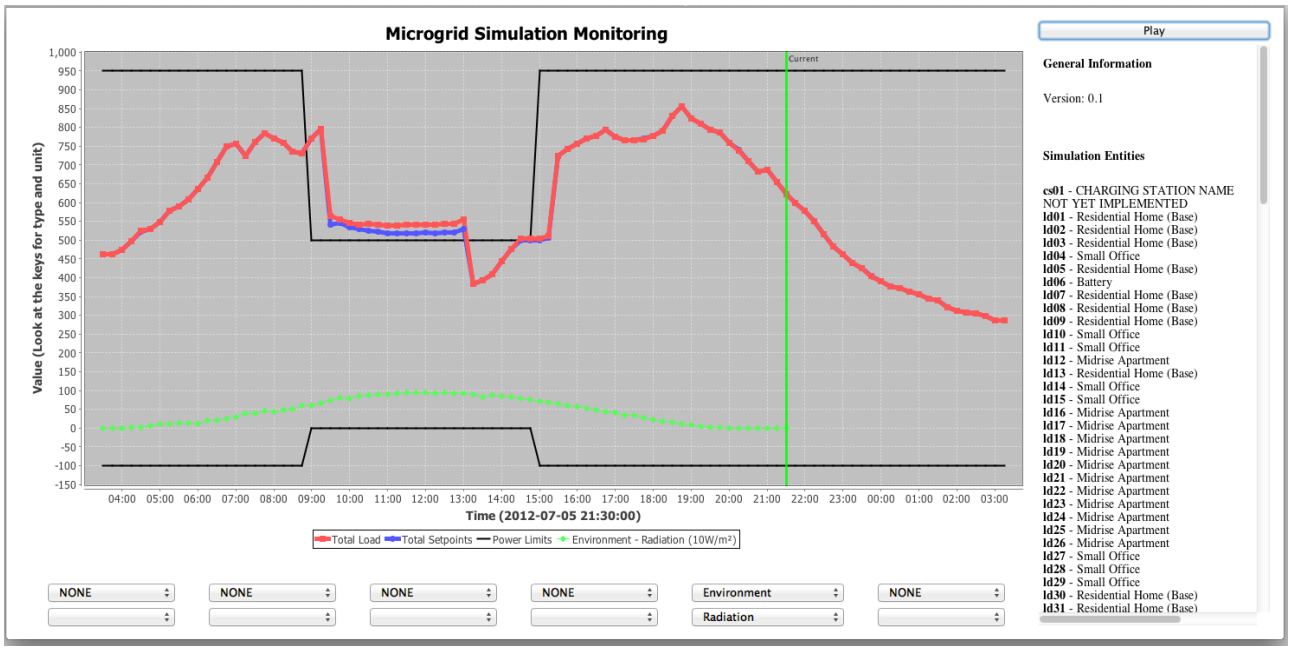


Figure 6-5 MGE simulation output, outage scenario. The reduced load is approx. 550kW

Figure 6-4 shows a snapshot of the run of a microgrid consisting of about 40 buildings. The limit at the transformer has been set to 950kW. The run was done on the 5th of July. The same configuration has been simulated in Figure 6-5, with an outage between 9am and 3pm. Due to the critical demand definition and the other rules mentioned above the reduced demand for this outage situation is down to 550kW. This value is used in the SILFAST tool to evaluate the remaining overloads in the higher level grid during this particular type of outage.

6.4 THE SINGLE LINE FAILURE TOOL (SILFAST)

This tool takes a meshed grid topology as input and performs a power flow calculation. The topology can be automatically modified by temporarily removing one branch (link) and recalculating the power flows. The resulting overloaded lines are reported (lines that have currents above their nominal values. Used together with the MGE Tool, it can be tested whether using the reduced microgrid loads calculated by MGE in each node of the high level topology still create overloaded lines.

7 CONCLUSION

This deliverable presents the IRENE toolset of supply demand prediction, threat identification and security classification. The lists of Stakeholders within the collaborative framework that maintain the grid resilience include Municipal authority planner, Distribution Network Operator (DNO), Developers, Critical Infrastructure Operator, Business and Citizen Representative.

This deliverable further presents the present design, modelling simulation and analysis of IRENE toolset, which include the Evolutionary Threat Analysis (ETA) tool, BayesianFAIR tool, Overall Grid Modelling (OGM) tool, Microgrid Evaluation (MGE) tool, and Single Line Failure Simulation Tool (SILFAST). The toolset is integrated with the previously defined methodologies and policies in WP1-WP3 and are further simulated in order to provide the analysis results that enable users to evaluate the efficiency of fault and attack mitigation measures, the energy resilience outcomes, and the impact on different critical infrastructures. Types of inputs to the toolsets are illustrated with IRENE individual tool simulations, and the update of methodologies and policies from fellow stakeholders to evaluate the overall grid resilience and also in the same instant, to evaluate the efficiency of the methodologies and policies as derived. The end of result analysis demonstrates the capabilities of the IRENE toolset that are able to investigate the threats and issues in the smart grid and are able to put into practice the identified mitigations.

The outcome of this deliverable will constitute the base for the open modelling framework in D4.2 [2] of the WP4, where the integration of the IRENE toolset for all components is validated internally.

8 ABBREVIATIONS

LP	Linear programming
GUI	Graphical user interface
EnKF	Ensemble Kalman filter
OGM	Overall Grid Modelling
ETA	Evolutionary Threat Analysis
FAIR	Factor Analysis of Information Risk
LEF	Loss Event Frequencies
CEMS	Customer Energy Management System
IRENE toolset	
ETA	Evolutionary Threat Analysis Tool
MGE	Microgrid Evaluation Tool
SILFAST	Single Line Failure Simulation Tool
OGM	Overall Grid Modelling Tool
BayesianFAIR	Bayesian Factor Analysis of Information Risk

9 REFERENCES

[1] IRENE D1.1, "IRENE scenario and baseline model," 2015.

[2] IRENE D4.2, "Open modelling framework," 2017.

[3] IRENE D2.2, "Societal impact of attacks and attack motivations," 2015.

[4] J. Jones, "An introduction to factor analysis of information risk (fair)," *Norwich Journal of Information Assurance,* pp. 67, vol. 2, no. 1, 2006.

[5] IRENE D3.1, "System architecture design, supply demand model and simulation," 2016.

[6] NIST, "Guide for Conducting Risk Assessments, NIST Special Publication 800-30," 2012.

[7] NIST, "Introduction to NISTIR 7628 guidelines for smart grid cyber security," 2010.

[8] IRENE D2.1, "Threats identification and ranking," 2015.

[9] H. A. S. A. a. A. B. Abbass, "MEBRA: multiobjective evolutionary-based risk assessment," *IEEE Computational Intelligence Magazine,* vol. 4.3, pp. 29-36, 2009.

[10] M. S. B. S. a. K. S. Lund, "Risk analysis of changing and evolving systems using CORAS," in *Foundations of security analysis and design VI. Springer Berlin Heidelberg*, 2011.

[11] H. A. Kopetz, "Conceptual Model for the Information Transfer in Systems of Systems," in *ISORC*, 2014.

[12] C. A. Z. T. B. A. Mori Marco, "On the impact of emergent properties on SoS security," in *SoSE*, 2016.

[13] L. H. a. L. E. H. Rosenberg, "Software quality metrics for object-oriented environ-ments," *Crosstalk Journal,* vol. 10.4, pp. 1-6, 1997.

[14] N. e. a. Ayewah, "Using findbugs on production software," in *ACM SIGPLAN conference on Object-oriented programming systems and applications companion*, 2007.

[15] A. Vasenev, A. L. Montoya Morales, A. Ceccarelli, A. Le and D. Ionita, "Threat navigator: grouping and ranking malicious external threats to current and future urban smart grids," in *1st EAI International Conference on Smart Grid Inspired Future, 19-20th May*, Liverpool, United Kingdom, 2016.

[16] IRENE D5.1, "State-of-the art in gaming simulations and stakeholder workshops for method evaluation (in progress)," 2016.

[17] lp_solve, “Introduction to lp_solve 5.5.2.5,” 2016. [Online]. Available: http://lpsolve.sourceforge.net/5.5/. [Accessed 19 October 2016].

[18] A. e. a. Najgebauer, “The prediction of terrorist threat on the basis of semantic association acquisition and complex network evolution,” *Journal of Telecommunications and Information Technology*, pp. 14-20, 2008.