FP7-ICT-2013.3.4

Project acronym: **AMADEOS**

Project full title: Architecture for Multi-criticality Agile Dependable Evolutionary Open System-of-Systems

Grant Agreement no.: 610535

Partners:

[1] Università degli Studi di Firenze

[2] Technische Universitaet Wien

[3] Université Grenoble Alpes

[4] ResilTech S.r.l.

[5] Thales Nederland BV

[6] European Network For Cyber Security Cooperatief Ua

D4.5 – FINAL RESULT AND LESSON LEARNED FROM PROOF OF CONCEPTS

(01.03.2016-30.09.2016)

Revision of the document:	4.1 (final)
Responsible partner:	RES
Contributing partner(s):	all
Reviewing partner(s):	TUW, UNIFI
Document date:	30/09/2016
Dissemination level:	PU

Change Records

Revision	Date	Changes	Authors	Status[1]
1.0	01/07/2016	First draft	RES	NV
2.0	01/08/2016	Contribution from pertners integrated	RES	NV
3.0	01/09/2016	Doc ready for internal review	RES	NV
3.1	20/09/2016	Internal Review comment addressed	RES	NV
4.0	29/09/2016	Ready for submission	RES	V
4.1	30/09/2016	Final editing for D4.5 submission	UNIFI	V

[1] V - Valid, NV - Not Valid, R - Revision

TABLE OF CONTENTS

LIST OF FIGURES

LIST OF TABLES

Executive Summary

This deliverable reports the outcomes of Task 4.3 “Assessment of AMADEOS methodology”.

In details, this deliverable is organized in the following way:

- Section 1 describes the scope of the AMADEOS assessment summarizing the aims of the AMADEOS qualitative assessment;
- Section 2 describes the general reference methodology used for the AMADEOS project assessment and the AMADEOS assessment plan derived from the methodology;
- Section 3 reports the results of the application of the assessment plan defined in Section 2;
- Section 4 describes the lesson learnt from the proof of concepts focusing on the benefits and limitations of the AMADEOS approach;
- Section 5 provides conclusion and possible future research directions.

1 INTRODUCTION

This deliverable reports on the assessment of the AMADEOS methodology which consists of the SoS SysML profile conceived in WP2 [2][3], and the Architectural Framework (AF) and prototype tools developed in WP3 [4][5]. In order to assess the influence of the AMADEOS methodology, on the properties of SoSs we will use quality metric concepts defined in WP2 (D2.3 [3], Section 2.12).

The assessment of the AMADEOS methodology makes use of methods based on the input of experts and realistic SoS use case scenarios in the energy domain to demonstrate the feasibility and applicability of the AMADEOS AF, prototype tools, the SoS SysML profile. As the AMADEOS methodology is based on the AMADEOS Conceptual Model (CM) our assessment will also indirectly validate the consistency, and demonstrate the applicability of the AMADEOS CM.

Further, this deliverable will – under the consideration of the assessment of the AMADEOS methodology – validate the achievement of the overall AMADEOS project objectives, present learned lessons, and conclude by proposing future research directions.

1.1 SCOPE OF THE AMADEOS ASSESSMENT

Starting from the outcomes of Task 4.2 (Design and implementation of the case study, see [7] and [8]) and Task 4.1 (Energy scenario and use case definition, see [6]), this task executes the use cases defined in Task 4.1, and analyses the results of the simulation execution.

The AMADEOS assessments process aims to evaluate the impact of the innovative concepts, novel formalism for SoS modelling and modelling tools on several different fields of application such as: research communities, academic professors, students and companies interested to the SoSs simulation potentialities of the AMADEOS prototype tools.

Especially in the academic and research fields, the AMADEOS methodology provides novel methods and techniques for exploring, examine in depth and find new design solutions for SoSs. Additionally, companies involved in design and development of new technologies could exploit the AMADEOS methodology, in particular the developed prototype tools, for simulating the designed SoSs and analyzing their behavior in order to observe possible problems and find the proper solutions. This particular exploitation of the AMADEOS methodology and its tools can contribute to improve the design of SoSs avoiding mistakes during design.

In the following the aims of the AMADEOS qualitative assessment are summarized.

1.1.1 <u>Overall aims</u>

- To qualitatively assess SoS properties (e.g., SoS evolution, timeliness, dynamicity, dependability and security) by defining and using measurable quality metrics, as well as, non-quantifiable quality indicators.
- To demonstrate the viability and efficacy of the AMADEOS Architectural Framework (AF) in a Smart Grid use case.
- To show that the overall AMADEOS methodology, incl. its underlying Conceptual Model, the AMADEOS Architectural Framework (AF) and developed prototype tools, is applicable and feasible by applying it in realistic use case scenarios.

1.1.2 <u>The Concept of Quality</u>

The concept of *quality* should be defined in order to plan the assessment process and strengthening qualitative research activities. Quality is perceived differently by different people. In a manufactured product, the customer as a user recognizes the quality of fit, finish, appearance, function, and performance hence, from this point of view, quality may be rated based on the degree of satisfaction by the customer receiving a product or a service and the ultimate test in this evaluation process lies with the consumer.

The customer's needs must be translated into measurable characteristics in a product or service. Once the specifications are developed, ways to measure and monitor the characteristics need to be found in order to provide the basis for continuous improvement in the product or service. The ultimate aim is to ensure that the customer will be satisfied.

Despite this, *quality* can be defined in different ways depending on the product or service and its purpose. In [25] several definitions of the term *quality* are given and explained. For the AMADEOS project, the definition of quality is the following: *The standard of something as measured against other things; the degree of excellence of something.* (see D2.3 [3] Section 2.12).

The qualitative assessment of AMADEOS is performed basing on this definition of quality researching a set of measures that represent *fitness for use* for all possible AMADEOS users including: researchers, professors, students or companies. Such measures are derived from a survey on the sectors in which AMADEOS exploitation will be performed, it foresees the identification of perceptions, resources, aspirations, relations of all possible users and stakeholders.

The definition of quality measures for AMADEOS represents a precise task of the AMADEOS assessment methodology application that is reported in Section 2.2.

2 ASSESSMENT

2.1 THE REFERENCE ASSESSMENT METHODOLOGY

This section describes the general reference methodology used for the AMADEOS project assessment and the AMADEOS assessment plan derived from these methodologies. The scope of this particular activity is to i) define a general assessment methodology ii) plan a set of activities for validating the AMADEOS methodology and ii) execute the plan and examine the outcome.

The AMADEOS activities will be assessed by qualitative evaluation of the outcomes and through a process of exploitation. Qualitative assessment collects data that does not lend itself to quantitative methods but rather to interpretive criteria.

The qualitative assessment approach focuses on describing a phenomenon in a deep comprehensive manner. This is generally done in interviews, open-ended questions, or focus groups. In most cases, a small number of participants participate in this type of research, because to carry out such a research requires many resources and much time. Interviews can vary from being highly structured and guided by open-ended questions, or be less structured and take the form of a conversational interview. Generally, qualitative methodology is the detailed descriptions of situations, events, interactions and observed behavior, direct valuation of people related to their experience and attitudes.

The assessment methodology (derived from [16], [17], [23]) relies on seven fundamental tasks aimed in principle to understand the subjects under assessment, define the objectives of the assessment and methods of the assessment, collects outcomes and comparing them to the objectives. The diagram in Figure 1 depicts the general assessment methodology above described highlighting its task as different logical layers of the whole assessment process.

In particular the methodology fundamental tasks are described in the following.

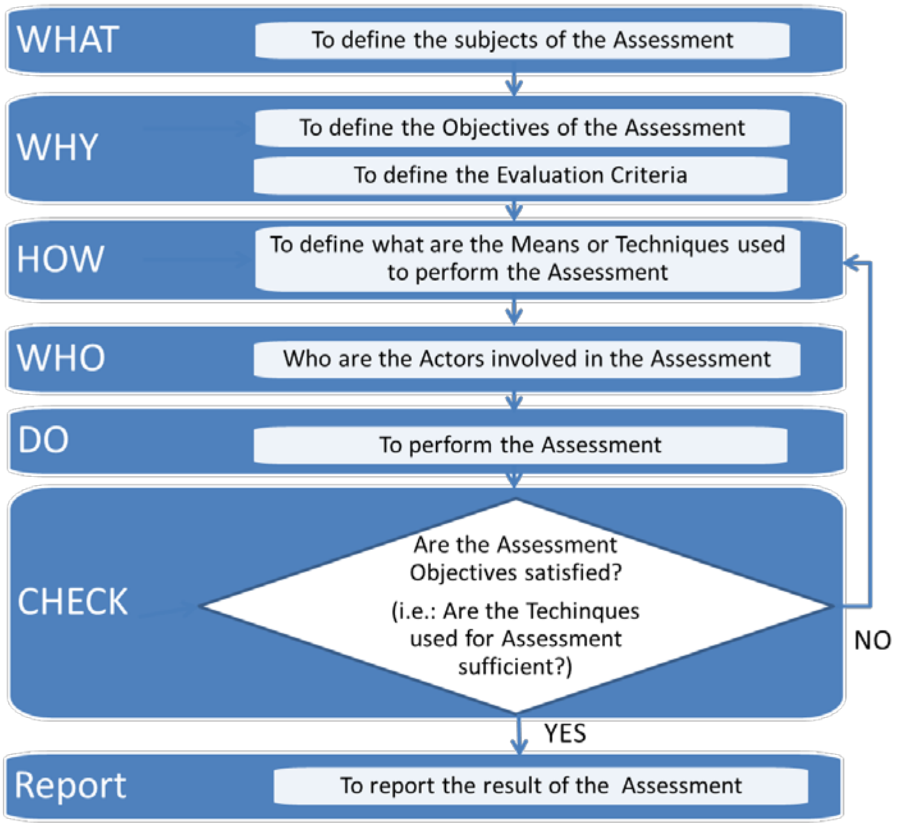

Figure 1 - The Qualitative Assessment Methodology

- The WHAT task is aimed to define the subject of the assessment. This task increases understanding of the activities and the systems under assessment analyzing any limitations or complexities and hence assisting in their interpretation.
- The WHY task defines the objectives and reasons of the assessment and its evaluation criteria in terms of qualitative measures.
- HOW defines techniques and methods to perform the assessment of the subject. This task can use appropriate technologies and tools for supporting the quality evaluation.
- WHO contributes to understand who are the actors involved in the assessment. This task should also find disadvantages caused by the processes and issues of diversity or conflicts that affect the actors.
- DO executes techniques and methods to perform the assessment defined in the HOW task and collect outcomes.
- CHECK performs a final analysis of the outcomes from the DO task in order to understand if the assessment has been reached. If the objectives are not satisfied the assessment is not achieved and the assessment process has to be restarted from the HOW task.
- REPORT corresponds to the assessment documents compilation

The techniques to perform the assessment (in the HOW task) can be different and they are defined ad hoc for the specific subjects of the assessment.

As described in [17], [18], [16], [19], [20], [21], [22], qualitative methodology exploits key methods for performing the assessment. They are defined in the task HOW and applied in the task DO. Key methods are based on the following:

- Direct Observation
- Case Studies
- Qualitative Interview

Direct Observation can involve experiments, related documents, images or videos for giving to the observer the correct perception of the activity under assessment. Observation is useful for getting a better understanding of the context, cross-checking information and possible differences between what people do and what they say they do, assessing the quality of relationships between individuals or groups.

Case Studies combine different methods to compile a holistic understanding of individuals, households, communities, market or institutions. Case studies combine investigation of: context, aspirations and perceptions, resources and relations, institutions and development interventions. They may be of many different types or described with different formalisms.

Qualitative Interviews aims to individuate issues and possible solutions by means of questionnaires or specific conversations. Both questionnaires and conversations include qualitative questions from low-level details to macro-level details.

In order to formalize and interpret data collected during the assessment phases it is necessary to consider objectives and goals established in the initial phase of the project. The key methods provides necessary information for deriving qualitative assessment of the activities comparing the results of the key methods with the objectives of the activities and requirements established during the design phase of the activities.

Open-ended questions and interviews allow researchers and practitioners to understand how individuals are doing, what are their experiences, and recognize important antecedents and outcomes of interest that might not surface when surveyed with pre-determined questions. Although qualitative research can be thought of as anecdotal, when pooled across a number of participants it provides a conceptual understanding and evidence that certain phenomena are occurring with particular groups or individuals. Hence the benefits of the qualitative approach can be summarized as follows:

- Allows identification of new and untouched phenomena
- Can provide a deeper understanding of mechanisms
- Gives a one-on-one and anecdotal information
- Provides verbal information that may sometimes be converted to numerical form
- May reveal information that would not be identified through pre-determined survey questions

2.2 ASSESSMENT PLAN FOR THE AMADEOS METHODOLOGY

In the previous section a general methodology for assessments has been defined. Starting from this methodology it is possible to define an AMADEOS project assessment plan for instantiating the general methodology in the AMADEOS activities. AMADEOS activities are related to provide basic SoS concepts, measurable SoS quality attributes, and propose a SoS design methodology, supported by extended mainstream UML-based tools, for the representation, modelling, development, evolution and validation of Systems-of-Systems.

Note that part of the AMADEOS qualitative assessment has been performed as self-assessment, indeed results from *Direct Observation* and *Case Studies* have been provided mainly by the AMADEOS test bed and by modelling of the case studies described in [7]. On the other hand, *Qualitative interviews* have been executed in collaboration with different observers (e.g., academic researchers, academic professors, students) and users analyzing discussion and questionnaires output.

In the following the assessment methodology is applied to AMADEOS step by step.

2.2.1 <u>WHAT</u>

According to the methodology defined in Section 2.1 we assess the following WHAT clauses reported in Table 1.

#	WHAT
1	Fielding industrial SoS, where the handling of time will be in the center of the analysis, in the domain of transport, smart grid application and a summary of the common findings.
2	A Conceptual Model for generic SoS
3	An architecture framework for generic SoS
4	Formalization of evolvability and dynamicity in SoS
5	A methodology for the design of SoSs that puts a focus on timeliness, complexity management, evolution, agility, dependability, security and emergent properties of the developing artefact.
6	Consideration of emergence in modelling SoSs.
7	Impact of introduction of global, synchronized time into SoS model & SoSE.
8	Supporting prototype software tools for the above-mentioned methodology based on SySML.
9	Building blocks and algorithms to improve the dependability, security and performability of a SoS.
10	Modeling and simulation results of case studies from the domains of smart grid automation.

Table 1 – Subjects of AMADEOS assessment

2.2.2 WHY

According to the methodology defined in Section 2.1, the Table 2 shows the objectives of the Assessment in relation to the WHAT clause.

#	WHY	# WHAT
1	With such an approach, each CS, interface and elements of a SoS has access to a precise synchronized global time base. This global time enables new solutions that are expected to result in guaranteed responsiveness, in reduced cognitive complexity, higher dependability and a simplification of the certification of safety-critical services.	**1, 7**
2	To reduce the cognitive complexity needed to comprehend the behaviour of a SoS by the application of appropriate simplification strategies. It is this considerable cognitive effort needed to understand the operation of a large SoS that is the main cause for the substantial engineering (and monetary) effort required to design and maintain many of today's system of systems.	**2, 3, 5**
3	Evolution of a SoS is necessary for the adaptation to environmental changes such as new business cases, legal requirements, compliance, changing safety regulations, evolving environmental protection rules, etc. On the other hand, unforeseen events can occur in a SoS, such as a disaster or a failure of a constituent system. In this case, widely applicable control protocols to tackle dependability, security and performability of SoS will be provided, allowing these systems to adapt to the dynamic environment. In fact, mitigating the effects of dynamicity and evolution will become necessary to the success of many SoSs.	**4, 5**
4	Understanding the mechanisms of emergence will help in the composition of constituent systems, especially in predicting the effects of composition on dependability, safety, security, availability. In close relation to the introduction of global synchronized time, we will investigate the role of time in the appearance of emerging properties and the required actions to predict or mitigate their effects on the SoS	**6, 5**
5	Over the years in many of the systems complexity has increased enormously and reached a point, where such system complexity is difficult to manage any further. Hence methodology is needed focusing on timeliness, complexity management, evolution, agility, dependability, security and emergent properties of the developing artefact.	**5**
6	To ease the adoption of the proposed methodology, supporting facilities are needed to guide the designer to model an SoS.	**8**
7	Guarantee dependability and security aspects in a SoS are important issues. For this purpose are necessary building blocks and algorithm to assess dependability and security of a SoS. AMADOES MAPE building block play a central role in this context. An important challenge raised during the AMADEOS MAPE architectural design has been the selection of technology able to implement the MAPE Analysis function for faults and failures detection. According to the AMADEOS architectural framework[4], Analysis should be able to perform the following actions: • Detect all possible types of fault and failure that can happen in an ICT SoS • Detect faults and failures in time for being able to manage a critical	**9**

<table>
<tr><th>#</th><th>WHY</th><th># WHAT</th></tr>
<tr><td></td><td>situations with timely reactions
• Adapt itself to the possible changes of the SoS

In order to detect and locate faults timely for allowing the proper recovery action, the MAPE modules should be connected online to the CSs. In addition, the MAPE Analysis solution should be applicable in every SoS configurations and every MAPE architectural pattern. Hence, adopting a suitable technology for Analysis means use methods a techniques that satisfy the above features.

Another challenge related to the MAPE Analysis implementation concerns measures used for fault detection. Generally, fault detection can be performed analysing indicators provided by probe sensors located in the CSs. Such indicators provide information related to the internal statuses of the physical components in which software programs runs. Indicators values are measured by CSs internal probes and propagated out of the component by means of RUMI.

• Adaptive and scalable service-level monitoring system is useful in anticipating of potential threats (both accidental and malicious) and emergent behaviours in order to generate detailed data, which is required for effective anomaly detection, failure prediction, and preventive and corrective maintenance.
• Cognitive and predictive models are useful to detect malicious attacks and operational failures by exploiting the monitored data of the infrastructure, and aiming to capture possible emergent system behaviours.
• Reaction strategies are useful to monitor results and subsequent failure predictions and on their impact on the trustworthiness of the system</td><td></td></tr>
<tr><td>8</td><td>To simulate the algorithms and interfaces that are required to model and simulate SoS architectures in the Smart Grid domain, and to use the results to validate the model.</td><td>10</td></tr>
</table>

Table 2 – Objectives of AMADEOS assessment

2.2.3 HOW

According to the methodology defined in Section 2.1, the Table 3 shows the techniques adopted to assess each of WHAT clauses.

# HOW	# WHAT	Key method	Term of evaluation
1	**1, 7**	Direct Observation Case Studies	Formalization of time in Conceptual Model and Resilient Master Clock. By observing the Case Studies results consisting of time viewpoint and usefulness of Relient Master Clock

# HOW	# WHAT	Key method	Term of evaluation
2	**2,3**	Direct Observation	By comparing the proposed conceptual model and architectural framework with SOTA.
3	**4**	Direct Observation Case Studies	Formalization of evolution and dynamicity in Conceptual Model. By observing the Case Studies results consisting of evolution and dynamicity viewpoints.
4	**5**	Direct Observation Case Studies	Formalization of dependability and security in Conceptual Model. By observing the Case Studies results consisting of dependability and security viewpoints. By observing reduction in cognitive complexity using the architectural framework, conceptual model and viewpoints. Case studies such as Intrusion and failure detection system in smart grid.
5	**6**	Direct Observation Case Studies	Formalization of the emergence in Conceptual Model. By observing the Case Studies results consisting of emergence viewpoint.
6	**8**	Direct Observation Case Studies Qualitative Interviews	By observing the Case Studies results, interviews from industry, INCOSE tutorial, academics or students, assessing the supporting facilities based on its usability, portability, expressiveness and Relevance/Utility[2].
7	**9**	Direct Observation Case Studies	Formalization of Building Blocks in Architectural Framework. By observing the results of Case Studies based on *Adaptive and scalable service-level monitoring, Cognitive and predictive models, Reaction strategies.* By observing the effectiveness of building blocks like MAPE in case studies.
8	**10**	Direct Observation Case Studies	By observing the effectiveness of the simulation tools and results of simulation on smart grid model with different inputs in various scenarios. By observing of supporting facility tool modelling and simulating the SoS with legacy system.

Table 3 – AMADEOS assessment techniques.

[2] Usability is directly connected with the ease of use of the supporting facilities. Portability is an attribute related to the possibility to execute the tools in different platforms. Expressiveness is intended as the capacity to express clearly the models and clearly exhibit output. Relevance/Utility gives the idea of how much people consider useful the subject of the assessment.

The following diagram shows the key methods used for AMADEOS in the HOW task. Each key method is represented with a rectangle while its relative input is represented with an ellipse. Outcome of the methods are used for checking with the assessment objectives if the assessment has been reached.

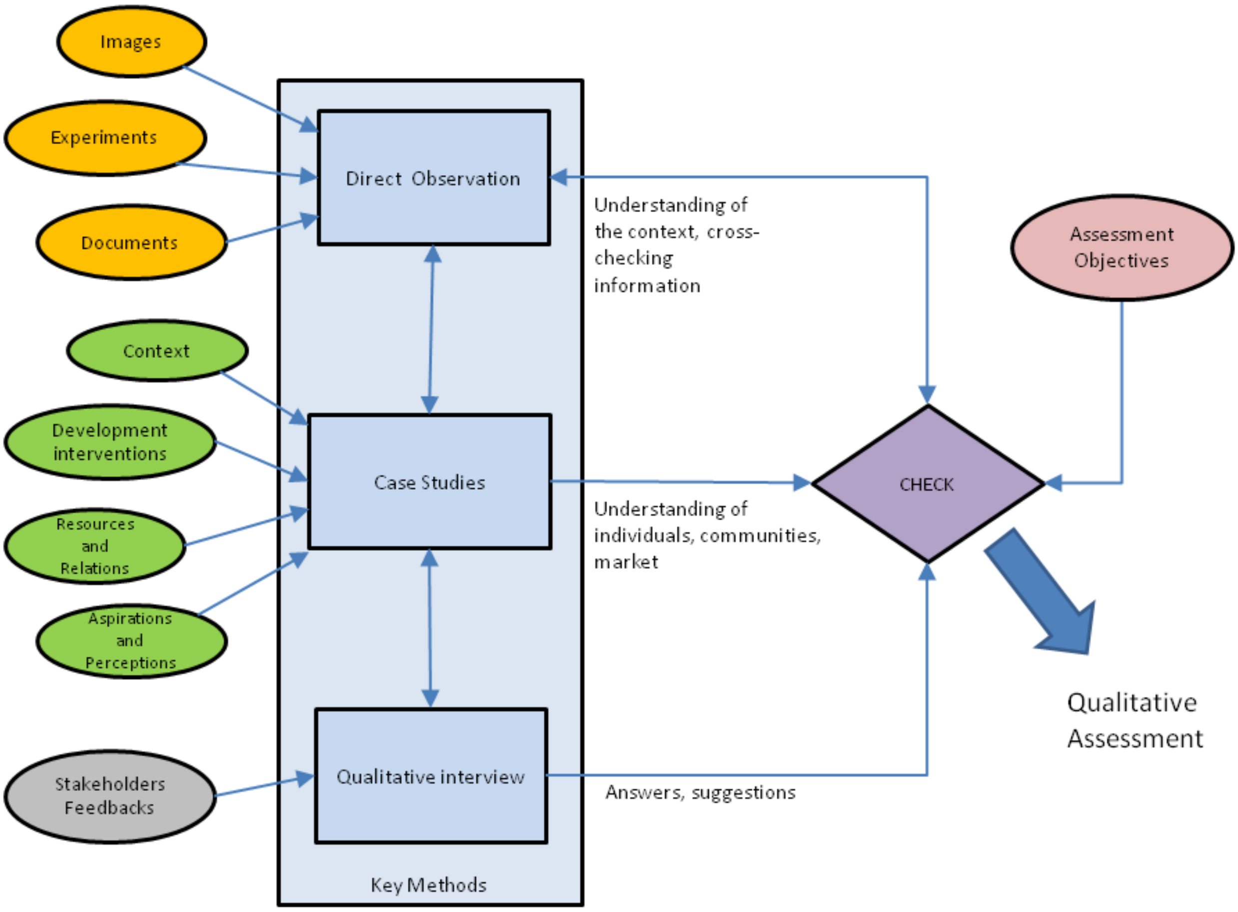

Figure 2 - The AMADEOS HOW task and its interactions

2.2.4 WHO

According to the methodology defined in Section 2.1 the Table 4 shows the actors involved in the assessment.

# WHAT	Actor
1	AMADEOS partners who have defined and modeled the case studies. (Thales, Resiltech, ENCS, UNIFI)
2, 3, 4, 5, 6	AMADEOS academic partners (UNIFI, TUW, UJF)
7	AMADEOS partners involved in the Resilient Master Clock (UNIFI, TUW)
8	AMADEOS industrial partners involved in modeling SoS (Thales, Resiltech, ENCS)
9	AMADEOS partners involved in the development of specific architectural building blocks (Resiltech, UNIFI, TUW,)
10	AMADEOS partners involved in SoS simulations (Thales, Resiltech, ENCS)

Table 4 – Actors involved in the AMADEOS assessment.

2.2.5 DO

According to the methodology defined in Section 2.1 the following Sections shows the task DO which uses the results of simulation on the AMADEOS test bed as the execution of the *Direct Observation* method in which different scenarios have been designed in order to address the most common problems that can happen in ICT SoSs also simulating the presence of common faults (i.e. malicious faults caused by a malicious user with the intention of prevent services and accidental faults caused by software/hardware malfunctions). The *Case Study* method has been performed in the DO task designing and implementing the AMADEOS Case Study described in [7].

2.2.5.1 DO #HOW 1

# HOW	# WHAT	Key method	Method of assessment
1	**1, 7**	Direct Observation Case Studies	Formalization of time in Conceptual Model and Resilient Master Clock. By observing the Case Studies results consisting of time viewpoint.

Cross domain common findings:

Many reference domains **(**i.e., Automotive, Railway, Energy, Banking (ATM), and Crisis Management) are analyzed in order to identify common core aspect relevant to SoSs.

In [1] the results of this analysis are presented, also following the Time viewpoint.

The concept of time is a key aspect for all considered domains.

- Synchronization requirements, he handling of time is a critical aspect in terms of:
- Checking real-time requirements.

The concept of a Global Time Base is often required to guarantee a proper functioning of the system. In particular, in the in-vehicle perspective within the Automotive domain and in the Railway domain, the concept of global time is not widely adopted, because the requirements of real-time assume greater importance than the synchronization. Conversely, in the case of V2X Communication in the Automotive domain, in the Energy and in the Crisis Management domains, synchronization requirements and, thus, the global time play a key role. In the Global Automated Teller Machine Network domain both the concept of real-time and the synchronization are important.

In all domains, the definition and the use of methodologies and tools are needed in order to guarantee a tight synchronization to the global time base and estimate resilient distance from the global time base, assuring the concept of time awareness.

It is worth noting that the time source should be reliable as much as possible. Time source redundancies can also be desirable as well. For example, as pointed out in a recent investigation performed by GAO (U.S. Government Accountability Office), GPS provides positioning, navigation, and timing data to users worldwide and is used extensively in many of critical infrastructures sectors, such as communications and transportation. However, sectors' increasing dependency on GPS leaves them potentially vulnerable to disruptions. A failure in the GPS service which makes it unavailable constitutes a common cause failure for all the sectors.

Time formalization:

In an SoS, external clock synchronization is the preferred alternative to establish a global time, since the scope of an SoS is often ill defined and it is not possible to identify a priori all CSs that must be involved in the (internal) clock synchronization. A CS that does not share the global time established by a subset of the CSs cannot interpret the timestamps that are produced by this

# HOW	# WHAT	Key method	Method of assessment

subset.

The SoS SysML profile expresses the time-related concepts by adopting the MARTE standard (OMG). MARTE is an UML profile that provides support for non-functional property modelling, defines concepts for software, hardware platform modelling and concepts for quantitative analysis (e.g. scheduling, performance).

Time is measure through clocks by defining a clock stereotype which extends the one defined in the MARTE profile. A MARTE Clock Stereotype is considered as a means to access to time, either physical or logical. The MARTE Clock is an abstract class and it refers to a discrete time.

The preferred means of clock synchronization in an SoS is the external synchronization of the local clocks of the CSs with the standardized time signal distributed worldwide by satellite navigation systems, such as GPS, Galileo or GLONASS.

All the details of the time basic concepts and clocks have been reported in [3] Section 2.2.1 and Section 2.2.2 respectively.

The Resilient Master Clock acts as a *dependable, accurate global time base* that includes local clock correction techniques, self-estimation of time awareness and fault tolerant synchronization solutions. The ultimate objective is to guarantee a consistent global time view across the SoS infrastructure. All the details related to RMC have been reported in [9].

Case study observation:

Following the recommendations of the EAB, the Time viewpoint aspects described in D4.2 haven't been further developed in a simulation, so their utility can only be assessed qualitatively by experts in the field of medium voltage network management. However, such study was not planned as part of the project.

2.2.5.2 DO #HOW 2

# HOW	# WHAT	Key method	Method of assessment
2	**2,3**	Direct Observation	By comparing the proposed conceptual model and architectural framework with SOTA.

We performed a SOTA analysis concerning the related ADL design approaches presented in the literature of SoSs. The analysis was not meant to be exhaustive but it was based on some of the most representative related works on designing SoSs. Its objective was to determine at what extent viewpoints-based SoS concepts have been already captured in the literature. All the details of this analysis have been reported in D2.3, Section 4.1.1.

2.2.5.3 DO #HOW 3

# HOW	# WHAT	Key method	Method of assessment
3	**4**	Direct Observation Case Studies	Formalization of evolution and dynamicity in Conceptual Model. By observing the Case Studies results consisting of evolution and dynamicity viewpoints.

Evolution and Dynamicity formalization:

Large scale Systems-of-Systems (SoSs) tend to be designed for a long period of usage (10

# HOW	# WHAT	Key method	Method of assessment

years+). Over time, the demands and the constraints put on the system will usually change, as will the environment in which the system is to operate. The AMADEOS project studies the design of systems of systems that are not just robust to dynamicity (short term change), but to long term changes as well.

Evolution: Process of gradual and progressive change or development, resulting from changes in its environment (primary) or in itself (secondary).

In the conceptual model and SysML profile, the viewpoint of Evolution shows the SoS from the evolution perspective that deals with long-term adaptation. Evolution of an SoS is necessary for the adaptation to environmental changes such as new business cases, legal requirements, compliance, changing safety regulations, evolving environmental protection rules, etc. SoS evolvability includes necessary modifications that are required to keep a system services relevant in the face of the ever-evolving society (e.g., new legal requirements, business cases, etc.).

In the conceptual model and SysML profile, the viewpoint of dynamicity describes the SoS from the dynamicity perspective. SoS dynamicity includes the adequate reconfiguration of SoS in specific situations, for example in case of an emergency management situation, such as a disaster or a failure of a CS after the occurrence of a fault.

All the details of the formalization of Evolution and dynamicity have been reported in [3] in Section 2.9 and Section 3.

Case study observation:

As stated in D4.2, the EV SoS must be designed to provide a friendly and convenient service to the users and, at the same time, profitable to the provider. Planning and scheduling is of paramount importance for both energy providers and users: as an example, on one side, if the charging requests are spread during the day, there will be limited and/or controlled load peaks on the grid to be handled, thus the energy price may not vary abruptly over time and prioritized consumers (e.g., police and fire-fighter vehicles, ambulances, etc.) will be easily handled by the charging station operators. On the other hand, knowing the energy prices and available time slots, the users will be able to carefully plan the recharging operation while keeping the service affordable.

A typical scenario would be as follows: EVs travel through a wide area, where several charging station operators provide recharging services, by means of charging points. Drivers in need of power for their EV can provide the expected charging context (duration, power, etc.) to the e-mobility service in order to receive information regarding recharging time slots and associated energy prices of each charging station operator. A load management optimizer that cooperates with the charging station operators carries out planning and scheduling activities. The interested driver will then choose one of the slot-price pair possibility for recharging its vehicle, will be allowed to plug-in its EV at the charging point of the chosen charging station operator during the reserved time slot only, and the amount due will be based on the energy consumption times the booked price. At the end of the recharging operation, the driver receives a billing invoice.

# HOW	# WHAT	Key method	Method of assessment

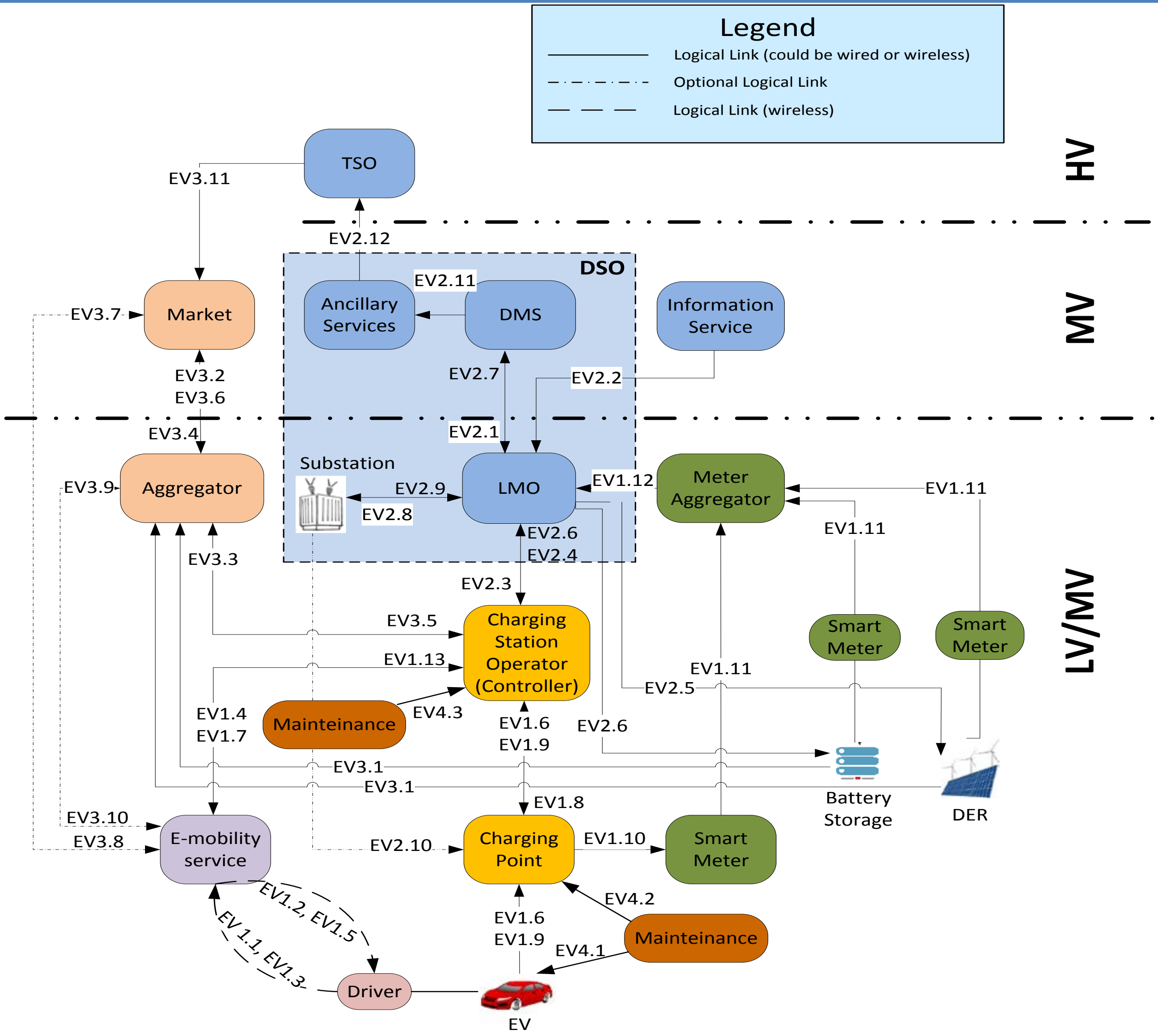

Figure 3 - EV Charging SoS (from D4.1)

With this scenario in mind, and using the AMADEOS tooling, we designed a SoS (shown in Figure 3, and described in D4.1) based on a typical EV rollout, in particular based on the desired situation in the Netherlands, based on interviews and workshops with experts, both in the EAB and Grid operators. This SoS was modelled using the Blockly tool (this is described in D4.2) and a simulator was generated from the tool. We combined this simulator with a simulation toolkit called SimPy and performed a number of experiments based on the scenarios defined in D4.2. This served to both validate the simulator as well as determine if such a simulation could be used to determine and validate possible future designs of the EV charging network, based on varying user and SoS behaviour.

The simulation is built using "SimPy" and uses fixed time slots for reservations. At the moment these slots are 15 minutes long and start on 0, 15, 30 and 45 minutes past each hour. Each Driver that wants to charge will attempt to make a Reservation for a Charging Point (via the E-Mobility Service) at a CSO. Because of these timeslots, a Reservation will only begin at the start of the next timeslot and will last for a number of time slots as calculated by the CSO. After the reserved time has passed, the Driver is expected to Plug Out its EV from the Charging Point, though he can do so at an earlier point in time (if the EV is already fully charged, or the driver otherwise decides to

# HOW	# WHAT	Key method	Method of assessment

do so).

Results from the EV Charging Scenarios:

The first scenario that we will discuss is a usage case where EV drivers do not immediately remove their vehicles from the charging points for a period of time. This means that those drivers who do not remove their EVs are acting badly – blocking charge points from other users. We chose a fixed period of four hours for this behaviour in this scenario, and the simulation took place over a period of 24 hours (simulated time via SimPy). There were 1000 EVs present and 500 Charging Points (CPs). In this simulation, market costs did not cause a significant change in behaviour. The EVs were set to desire a charge of 70%, with a “must charge” threshold at 20%. Four states are defined in the simulation: Charging, Driving, Idling and Waiting. Charging and Driving are self-evident. Idling was periods of time when the EV was not in use (due to lack of driver need). Waiting was an undesirable state where an EV was waiting for a CP to become available. In order to stress the Grid and the SoS as a whole, Idling was the least desired state of the three normal states – the chance of an EV idling was set to 10%. Finally, note that the drivers, cars and initial state were randomly assigned and a period of fifteen minutes was allowed to let the state settle. This can be seen in an initial jump in all of the graphs.

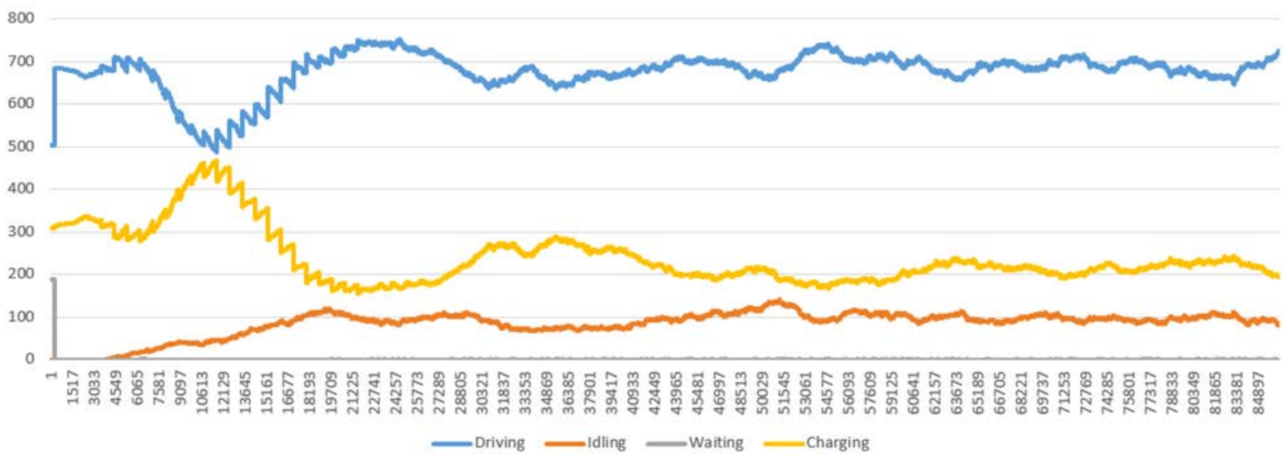

Figure 4 - Initial state where no drivers act badly

In Figure 4, you see four basic types of behaviour: Driving, Idling, Waiting (for a free Charging Point) and Charging. The numbers on the X-axis reflect the number of EVs in that state. In this scenario, (and in all our tests, due to global variables that were set) the initial state was close to optimum for each of the EVs, with around 70% driving and 30% charging. The initial spike in charging (around 10000 seconds into the simulation) is due to the initially driving EVs falling below the 20% lower threshold and requiring a charge. You can then see that after around 20000 seconds, the system reaches a steady state with around 70% driving, 20% charging and 10% idle. Note also that there is an insignificant number of EVs in the waiting state.

# HOW	# WHAT	Key method	Method of assessment

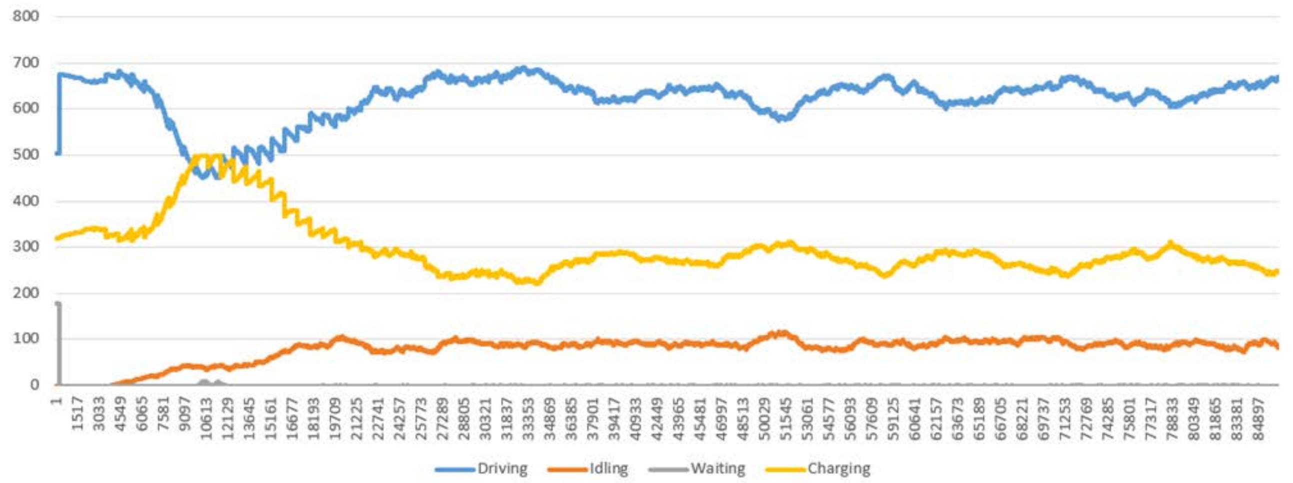

Figure 5 - 20% of EVs acting badly

Figure 5 shows the first set of EVs acting badly – in this case, 20% delay their disconnection from the CPs and block other drivers from using them. This scenario shows the resilience of the SoS to such behaviour – there is again the same spike in charging after around 10000 seconds, and a minor amount of EVs waiting, but again after around 20000 seconds, the SoS reaches a relatively steady state, although with more EVs charging (around 5% more) at any given time (and consequently 5% less driving) that the optimal case. One last thing to note is that all CPs are in use during the initial spike.

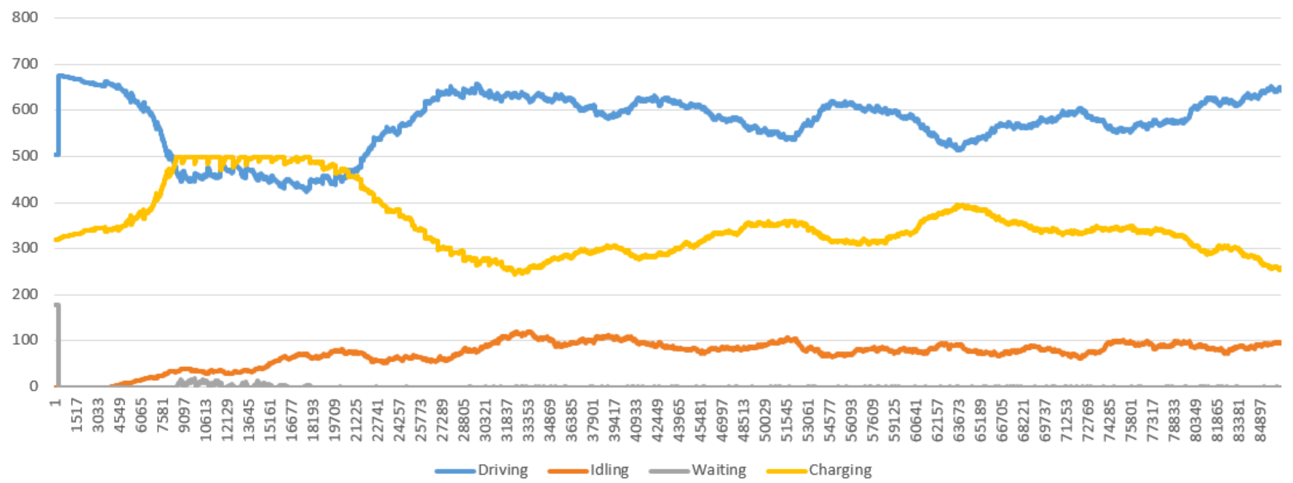

Figure 6 - 40% of EVs behaving badly

In Figure 6, the consequence of bad actors is more apparent. Now, there is a noticeable issue for EV drivers during the initial spike in usage, with a period of time when all of the CPs are in continual use (for around 15000 seconds) and there are some drivers constantly waiting for service. However, the SoS is still very resilient to this issue – the number of waiting drivers is still very low (maximum was 16 drivers). Furthermore, the average wait time was around 600 seconds (ten minutes). Again, this shows the resilience of the SoS to malicious events. However, again there is a drop in number of active drivers when the SoS reaches a steady state of another 8%, with the number of charging drivers up by the same amount.

# HOW	# WHAT	Key method	Method of assessment

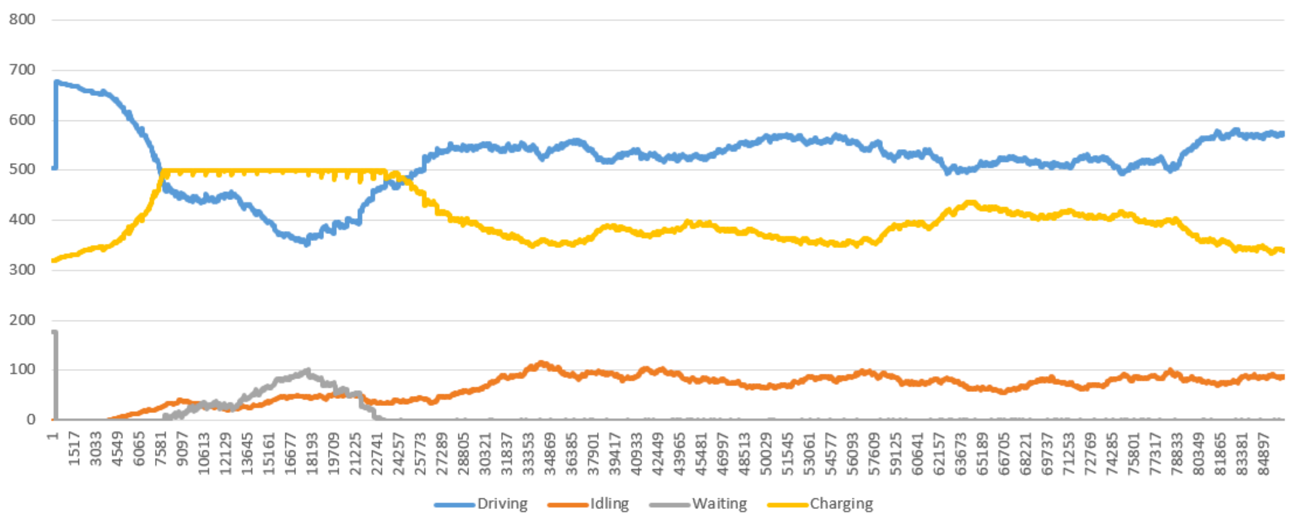

Figure 7 – 60% of EVs behaving badly

Finally, in Figure 7, the first significant effects of bad behaviour can be recognised, while 60% of drivers are acting badly. First, the period where the initial spike causes full usage of all CPs, and consequently up to 100 EVs waiting for service at the worst point. Despite this, there is another drop in active drivers when the simulation reaches the steady state – down to around 58% from a high of over 70% in the control simulation.

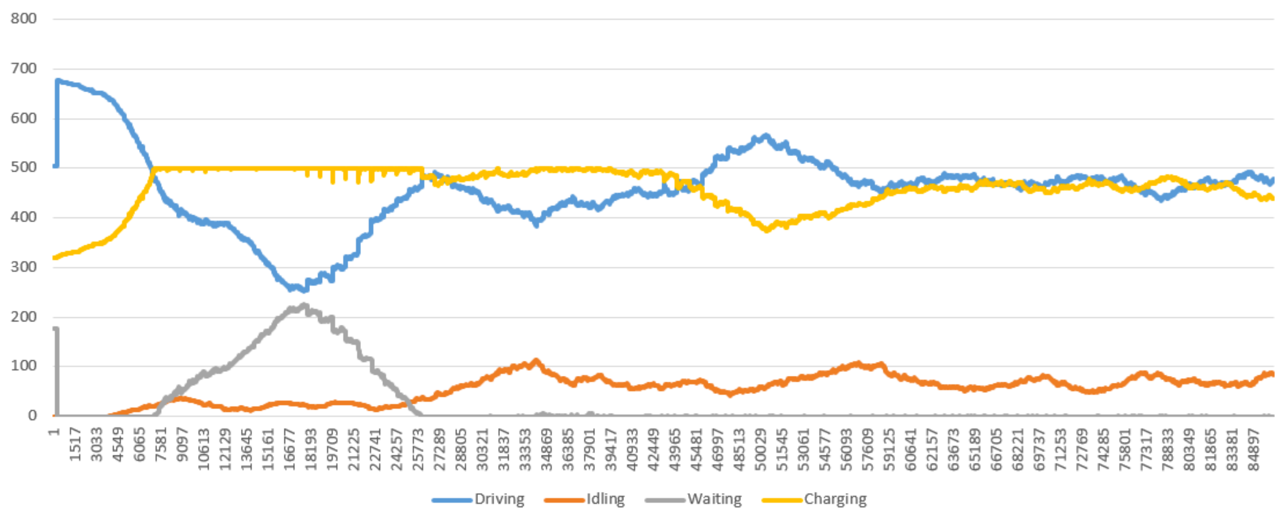

Figure 8 - 80% of EVs behaving badly

In Figure 8, we can see some significant changes in the usage profile of the SoS due to the bad behaviour of the EV drivers. In this scenario, during the initial spike, there is now a period of around 35000 seconds (nearly 10 hours in total) where all of the CPs are in constant use. This is also reflected in the more than 200 EVs waiting at one point. In this scenario, we now see less than 50% of the EVs driving when the SoS reaches a steady state, with roughly the same number charging as driving. This means that 20% of the EVs have changed from driving to charging since the control simulation. However, this is taking place where only 20% of drivers are behaving

# HOW	# WHAT	Key method	Method of assessment
correctly and yet the SoS remains (for the most part) available for use.			

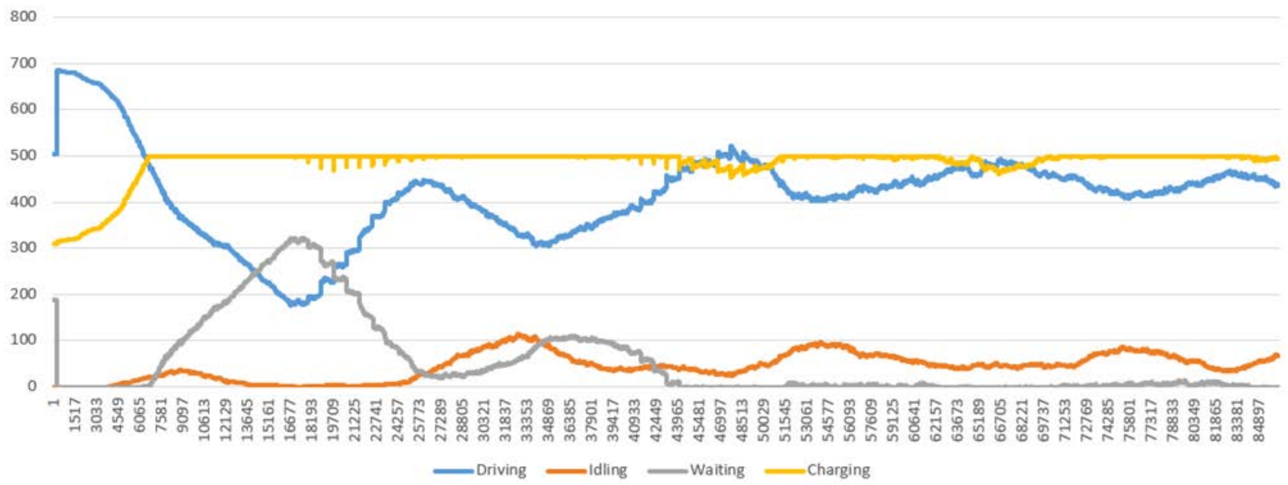

Figure 9 - 100% of EVs behaving badly

In the final simulation, shown in Figure 9, every driver is now acting badly. In this case, the effect on the SoS is dramatic and relatively catastrophic. In this simulation, for essentially the entire day, all of the CPs are in constant use, the number of drivers is down to less than 50% and for large periods of time, EVs are waiting to charge (more than 30% at the peak).

2.2.5.4 DO #HOW 4

# HOW	# WHAT	Key method	Method of assessment
4	**5**	Direct Observation Case Studies	Formalization of dependability and security in Conceptual Model. By observing the Case Studies results consisting of dependability and security viewpoints. By observing reduction in cognitive complexity using the architectural framework, conceptual model and viewpoints. By observing the results of case study such as Intrusion and failure detection system in smart grid.
Dependability formalization: In the conceptual model and SysML profile the Dependability viewpoint provides “dependability guarantee” block which can be achieved through different techniques. These techniques are used to reduce the occurrence of faults. The following techniques are provide by the profile: “fault prevention”, “fault tolerance”, “fault removal”, and “fault forecast”. The “dependability guarantee” is measured quantitatively for the following measures : availability, reliability,			

# HOW	# WHAT	Key method	Method of assessment

maintainability, safety, integrity, and robustness.

Case study observation:

The Smart grid model with IFD is modeled using dependability and security viewpoints of AMADEOS framework (more details can be found in 2.2.5.7).

In addition, Market simulation and Energy Usage simulations were performed to determine the reaction of the SoS to malicious behaviour on the part of the drivers. The results described above shows how the SoS behaves from the perspective of the EV drivers. However, there is another important aspect that can also be studied: the reaction of the SoS from the perspective of the Grid. This was calculated from the perspective of the TSO (see Figure 3). The mission of the SoS is, basically, to ensure the stability of the Grid, and to ensure that large changes in energy generation are not required. In order to achieve stability, a number of measures were enacted. First, the CPOs (there are 5 independent CPOs in this simulation) received a wholesale price from the DSO, via the market, based on:

1. Forecasted demand (by the DSO) for the next 24 hours, in fifteen minute intervals, and
2. How much energy they predicted that they required also in next 24 hours, in fifteen-minute intervals.

The goal of a CSO was, using the market price, to ensure they did not stray too far from their forecasted need. Prices were set by the DSO based on five energy bands – the price per unit was set based on the band that the CSO requested, regardless of the actual energy used in reality. The ideal situation for a CSO was to get as close to the top of a band, without exceeding it, as the cost per unit would jump, and make the per unit cost more expensive. Therefore, if a CSO discovered that (based on EV reservations) that they were going to jump up to the next band, their behaviour would be to attempt to get many more customers, by reducing the customer price. We used this aspect to drive competition and the response from the domain experts was that this is indeed a desired future scenario. This aspect proves the evolutionary promise of an AMADEOS SoS design.

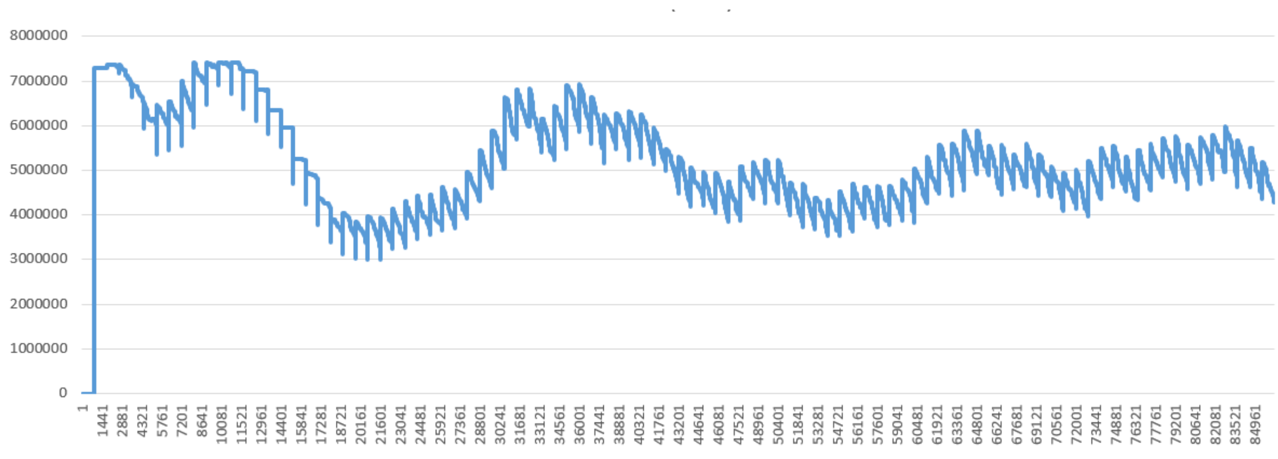

Figure 10 - Total network load with 0% bad behaviour.

Based on the same simulations as shown in Section 2.2.5.3, Figure 10 shows the reaction of the Grid to the EV charging scenario where no bad behaviour is present. The requests come in fifteen-minute intervals, and this can be seen in the jagged lines that are present in the graphs. In this instance, there is again an initial jump, after the 15 minute settling down period and then a peak and trough after around 18000 seconds. This is again due to the jump in number of users charging

# HOW	# WHAT	Key method	Method of assessment

at the start of the simulation based on their initial charge state and desire not to fall below 20% / attain 70% charge. The interesting outcome is the narrow band (between 40MWh and 60MWh) that the Grid eventually stabilises towards. Future research will definitely consider running such experiments over several weeks of simulated time and integrating a typical Grid usage pattern into the demand[3].

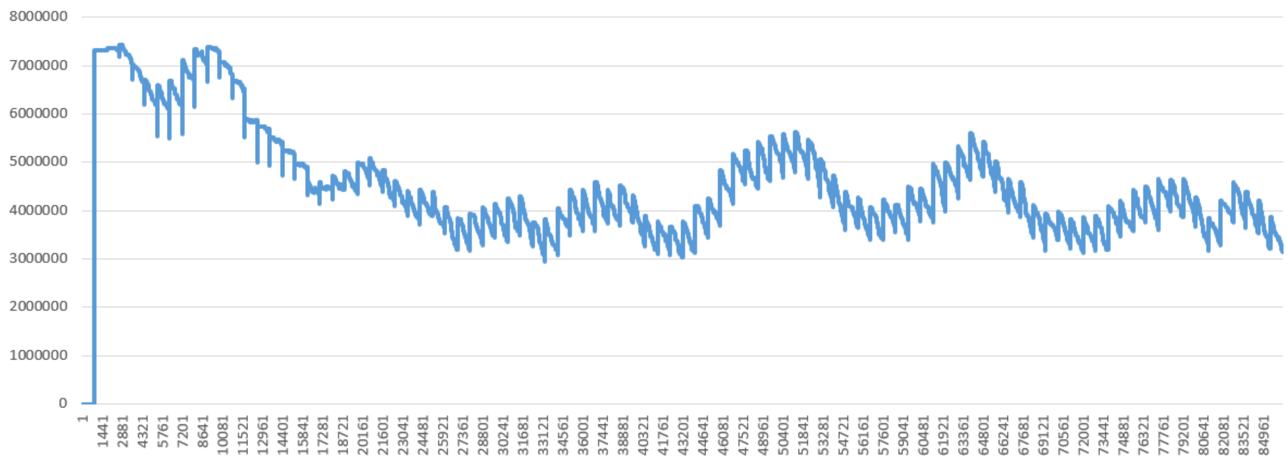

Figure 11 - Total network load with 40% bad behavior

The 20% (not shown) and 40% bad behaviour results (see Figure 11) show how the bad behaviour is increasingly reflected in the variations of load over time. The intermediate results (60% and 80% bad behaviour) show increasingly wide variations, cumulating to the wide swings shown in Figure 12, where all of the drivers are again behaving badly. These variations once again show how malicious actors will cause significant issues for Grid operators.

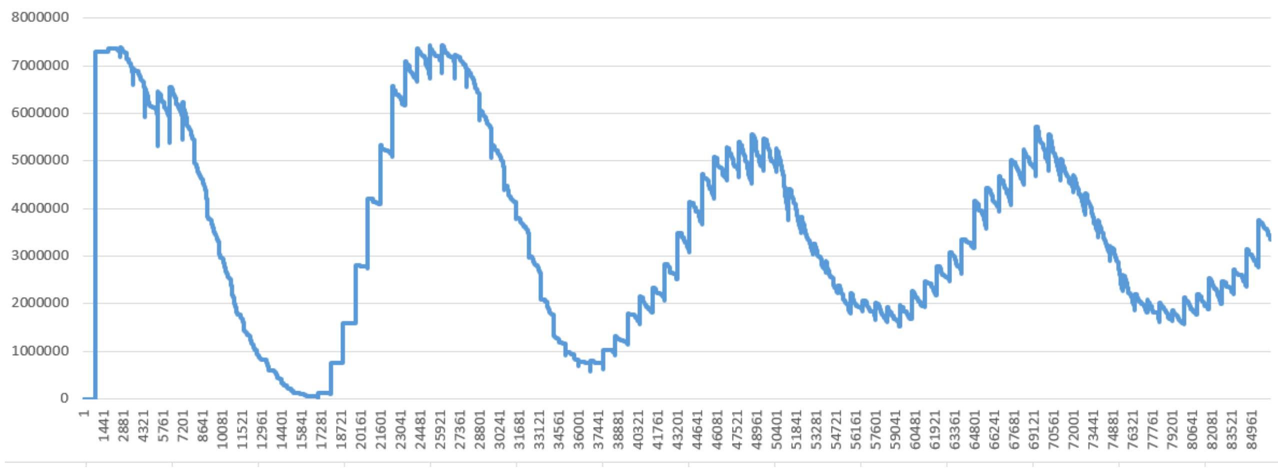

Figure 12 - Total network load where 100% of the drivers are acting badly

[3] As discussed in D4.2, the simulation was not run in concert with the MV case study.

2.2.5.5 DO #HOW 5

# HOW	# WHAT	Key method	Method of assessment
5	**6**	Direct Observation Case Studies	Formalization of the emergence in Conceptual Model. By observing the Case Studies results consisting of emergence viewpoint.

Emergence formalization:

Emergence phenomena has been formalized in conceptual model and the SysML profile consisting of emergent phenomenon such as explained, unexplained, and resultant phenomenon. The phenomenon are caused by behaviors which are characterized as expected or unexpected behavior. Also, behaviors of beneficial and detrimental types have been considered.

Emergence phenomenon are explained by transordinal law.

The simulations carried out in relation with the EV scenario have allowed the measurement of a wide range of parameters related to the dynamics of the EV charging actions, and of the Charging Point, CSO, DSO, Market, and Aggregator. The scenario assumes that each EV that that needs to charge will attempt to make a Reservation for a Charging Point (via the E-Mobility Service) at a CSO. The scheduling is performed on discrete time slots, so reservations can only begin at the start of a timeslot and will last for a number of time slots as calculated by the CSO. The EV is expected to disconnect from the Charging Point at or before the end of the reserved time.

The CSO keeps track of all incoming charging point reservations and running charging actions. The CSO can schedule all reservations according to a number of strategies. For the purpose of these experiments two strategies have been implemented: (i) Greedy, where each EV is charged and the maximum rate, and (ii) Balanced, where the charge rate ensures the full charging by the end of the reservation time.

For the DSO and Market a number of functions have been implemented to simulate the load balancing for multiple CSOs, and the price variations as function of charging requests and the amount of produced energy.

By varying these different simulation parameters different emergent phenomena can be observed, such as oscillations of the total charge of the running vehicles and of the price. However, the number of simulations was not sufficient for extracting more interesting relations between e.g. planning and pricing strategies and other aspects of the simulation. Nevertheless, the AMADEOS Emergence viewpoint model and the developed tool seem to be very effective in modeling and simulation of complex distribute processes.

2.2.5.6 DO #HOW 6

# HOW	# WHAT	Key method	Method of assessment
6	**8**	Direct Observation Case Studies Qualitative Interviews	By observing the Case Studies results, interviews from industry, INCOSE tutorial, academics or students, assessing the supporting facilities based on its usability, portability, expressiveness and Relevance/Utility.

SoS such as Smart Grid consisting of EV-Charging, Household Management and Medium Voltage Control; interface with Intrusion and Failure Detection System case studies are used as assessment for the supporting facilities tools.

Feedback on the supporting facilities tools from AMADEOS industrial partners involved in SoS modeling, AMADEOS academic partners, students, and INCOSE tutorial participants is collected for assessment. Feedbacks from users have confirmed that the Blockly concepts are suitable for modeling SoSs and the Blockly tool functionalities allow to model SoSs in easy way. This means that the Blockly modeling tool is user friendly and easy to use. In addition users did not have problems to install the tool on different operating systems hence it is possible to conclude that the tool reached a good *Usability* level.

Portability is guaranteed by the possibility to install the AMADEOS facility tools in different operating systems, both Linux and windows are included.

Expressiveness quality of the Blockly formalism is represented by two main aspects: i) the design power of the formalism that allows modeling any SoS in any context and ii) the capacity of the tool to express by intuitive graph representation the systems designed by humans. On the other hand the IFD tool features allow the user to be continually informed about the status of the SoS every time and in (near) real time with the possibility to access to a huge quantity of data and information using the real time data plotting feature. These features have assigned to the IDF tool ah high level of *Expressiveness* as confirmed by the feedbacks received form users.

From the *Relevance/Utility* point of view the Blockly tool has proved capable of modeling very different systems belonging to different contexts and automatically producing the relative code for the simulation of the SoS. This feature has been appreciated by expert and academic users that experimented the Blockly tool and hence they have proposed software extensions for improving the tool and apply it in the context of energy distribution for retrieving useful information during the design phase of smart grids.

The IFD tool, even if it is still in an experimental phase, has received positive opinions mainly thanks to its great versatility and applicability in different contexts (e.g. Service Oriented Applications, market applications, energy distribution, etc.).

Collected Assessment Results for the Blockly Tool					
Source	**Method**	**Usability**	**Portability**	**Expressiveness**	**Relevance/Utility**
Resiltech	Direct Observation	The tool is simple and intuitive to use for design SoS. Some improvements have been suggested as future work to enhance user-interaction.	The tool is completely portable. It works across the Operating Systems. It has also been known to work on tablets and hand held devices.	The tool is very intuitive to use and is able to express clearly the models and clearly exhibit output. Behaviors for each component could be easily expressed in Python. Constraints could be expressed in JavaScript which is executed in design time to warn against constraint violations. Models were easily searchable using the model query and to have custom viewpoints.	The tool is helpful for the designer to avoid mistakes in the design phase of a SoS. The tool is fast and requires no installation and less resources to run. This makes it ideal for rapid modeling and simulation of SoSs.

	Case Studies	The tool it was very intuitive and easy to use for the early stage of design the following case studies: EV-Charging SoS, Household Management SoS , and Medium Voltage Control SoS	The tool is very portable. The Python code automatically generated is easy to configure and run on various systems.	The EV Charging, Household Management and Medium Voltage control models designed with the tool are easily understandable. The tool provides the viewpoint mechanism to highlight several properties of the above mentioned SoS models. (e.g Architectural viewpoint, Interface viewpoint, Dependability viewpoint, Security viewpoint, etc.)	The experience of using the tool with case studies was very positive, similar design done in SysML without viewpoints was complex and difficult to manage due to too many lines in the diagram. The filter feature of the tool helped concentrate on a particular viewpoint of a large case studies.
Thales	Direct Observation	Generally usability is good, especially as the tool is in early stage of development. Specific improvements were suggested.	The tool is very portable – it works across operating systems and the outputted code was easily configured and run.	All relevant ideas could be expressed. The challenge expressed was where to express some concepts. That is, there are multiple possibilities where a concept could be modeled.	Utility of the Blockly tool was clear to all users. The concepts required some explanation, but expert
	Case Studies	The EV-Charging case study was very complex and entailed linking to a simulation tool (SimPy). While the outputted simulation code was used, some aspects of the interface need improvement: e.g. allowing links to files rather than embedding the code into	Again no issues were reported and the portability of the tooling relied upon Python. The architect experts saw this as a good choice of technology.	The case study was expressed with sufficient ease to allow real results to be generated and real conclusions to be drawn. The case study experiments showed that the different aspects (security, dependability etc.) could be expressed and experiments run within these aspects.	The utility and relevance of the tooling was clear within the case study. One comment (paraphrased) from a domain expert "This is exactly how we would like to design the electricity grid in the future in the Netherlands". In particular domain experts were very impressed with the expressiveness, relevance and utility of the tool.

		the Blockly tool.			
ENCS	Direct Observation				
	Case Studies				
UNIFI	Direct Observation				
	Case Studies				

Table 5 - The Blockly Assessment

Collected Assessment Results for the IFD Tool					
Source	**Method**	**Usability**	**Portability**	**Expressiveness**	**Relevance/Utility**
UNIFI	Direct Observation				
	Case Studies				
Resiltech	Direct Observation	The IFD GUI is intuitive and simple to use. Through the IFD GUI it is possible to select the detection methods.	The IFD can be installed in several operating systems because it is implemented in java. The IFD can be equipped with a standard communication protocol module (i.e. SNMP)	The graphical functionalities of the IFD provide a clear real time view of the monitored SoS probe indicators and their relative adaptive thresholds.	Even if the IFD is still in an experimental phase, it provides several features for faults detection respect of many other COTS detectors.
	Case Studies	The detection rules can be customized to detect several malicious and accidental faults because the core of the IFD is based on ESPER engine. This technology allows the	Not Applicable	The graphical functionalities of the IFD provide a clear real time view of the detected fault. Furthermore the CS involved in the faults are displayed in real time.	The developed IFD can find applicability in several scenarios and application domain due to its adaptability and extensibility.

		recognizing of several type of faults.			

Table 6 - The IFD Assessment

2.2.5.7 DO #HOW 7

# HOW	# WHAT	Key method	Method of assessment
7	**9**	Direct Observation Case Studies	Formalization of Building Blocks in Architectural Framework. By observing the results of Case Studies based on *Adaptive and scalable service-level monitoring, Cognitive and predictive models, Reaction strategies.* By observing the effectiveness of building blocks like MAPE in case studies.

Formalization of Building Blocks:

The AMADEOS conceptual model formalizes the following building blocks (i) Resilient master clock, (ii) MAPE design patterns.

An appropriate infrastructure whose aim is to support a set of highly-dependable services, which are essential for an SoS architecture, namely Monitoring, Predicting and Reacting services. In order to implement the support to the above services we adopted the well-known MAPE-k cycle to implement the above services through Monitoring, Analyze, Plan and Execution components.

All the details related to MAPE building block have been reported in [5].

The Resilient Master Clock acts as a *dependable, accurate global time base* that includes local clock correction techniques, self-estimation of time awareness and fault tolerant synchronization solutions. The ultimate objective is to guarantee a consistent global time view across the SoS infrastructure. All the details related to RMC have been reported in [9].

Direct observation:

In AMADEOS simulation environment, each component is simulated as a black box whose input and output are connected to other components composing the overall SoS. Each CS is simulated by a process that emulates both nominal and faulty behaviour without the possibility to trace the internal states of the corresponding real component. For this reason CSs indicator values are not available in the simulation.

Possible solutions have been examined as the following:

- The CSs simulation is performed using the complete real code of the corresponding real component. Thereby software probes can measure realistic indicator values
- Indicators values can be dynamically evaluated using simulation algorithms

The first solution is not feasible because it leads to a terrible increment of the simulation complexity and furthermore the implementation of the components may be very difficult.

The second solution does not affect the complexity of the simulation but the implementation of the probe algorithms provides non-realistic indicator values.

The challenges related to the MAPE Analysis have been completely resolved with the adoption of Complex Event Processing (CEP, [10]) technology that allows an efficient management of the pattern detection process in the huge and dynamic data streams and as such it is very suitable for recognizing complex events and situations online. CEP consists of the processing of events generated by the combination of data from multiple sources and aggregated in *complex-events* representing situations or part of them. Common event processing operations include reading, creating, transforming, and deleting events.

CEP is the means to allow correlating indicators incoming from CSs, aggregate them reducing redundancy, computing complexity, and applying detection rules in real time allowing timely faults and failures detection for its management.

The AMADEOS MAPE Analysis function has been implemented relying on the Esper [11] CEP engine. The Esper engine works a bit like a database turned upside-down. Instead of storing the data and running queries against stored data, the Esper engine allows applications to store queries and run the data through. Response from the Esper engine is online when conditions occur that match queries. Esper queries represent detection rules for faults detection. They are expressed through the Event Programming Language (EPL) [11] syntax. EPL has been designed for similarity with the SQL query language but differs from SQL in its use of views rather than tables. Views represent the different operations needed to structure data in an event stream and to derive data from an event stream.

In the AMADEOS MAPE architecture, Esper implements the core of the AMADEOS Intrusion and Failure Detector *(IFD)* tool in charge of indicators correlation. Esper has been also selected because it offers the following benefits: i) EPL rules can be modified at runtime without requiring to halt the IFD component, ii) Esper is easy to configure by using EPL rules and iii) Esper is totally developed in Java, for this reason it has been integrated easily in the IFD component.

Analysis processes measures provided simulating the presence of probes in the CSs and using indicator values incoming from implemented real systems. In particular, the performed experiment have been executed taking input data from the measures provided by the tests executed in SmartC2Net project [13] and Secure! project[12]. In SmartC2Net project aspects related to ICT security in smart grids have been considered and relevant indicators have been used as input in AMADEOS test bed. On the other hand the Secure! project has provided data exploited for the simulation of energy management services. This solution has allowed a realistic simulation of the target SoS, the Medium Voltage Control case study described in [7], based on smart grid architecture.

Given the complexity of the interactions in the considered use case, the implementation and testing has been restricted to a subset of the components and steps described in the following section. The focus is on those interactions that can be tested in a simulation environment.

Figure 13 shows how a cyber attack (UDP flooding Attack) is simulated in the SoS simulation. Considering only those CSs directly involved by the attack, the *Load Management Optimizer* (LMO) and the *Charging Station Operator* (CSO) described in[7], probes read indicator values stored in datasets called *trends* and provide them to the CS RUMI. The dataset corresponds to the behaviour of the specific indicator under test, including the presence of the attack that happens at a certain time. In this case, the CSO probe reads the udpInErrors indicator trend while the LMO probe reads the RTT indicator dataset.

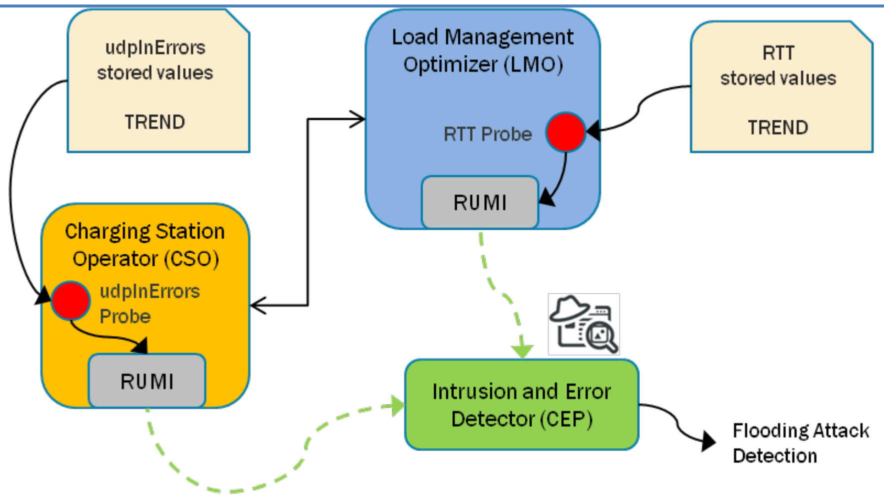

Figure 13 - The Flooding Attack Simulation

The trend is analysed by the Analysis and the attack is detected immediately after the trends change from the nominal to the faulty behaviour.

Adaptability of the MAPE Analysis has been achieved using an auto adaptive threshold algorithm applied on the indicator values. The algorithm executed to perform anomaly detection is called Statistical Predictor and Safety Margin (SPS, [15], [14]) algorithm already described in [5]. We present only few details on the SPS algorithm for brevity as they are not relevant for the considered discussion. SPS computes a prediction of the behavior of an indicator, based on a statistical analysis of the past observations. The prediction produces an interval, given by a minimum and a maximum value, in which the value of the indicator is expected to fall. If the value of the indicator is outside the interval, SPS signal that an anomaly is suspected for such indicator.

2.2.5.8 DO #HOW 8

# HOW	# WHAT	Key method	Method of assessment
8	**10**	Direct Observation Case Studies	By observing the effectiveness of the simulation tools and results of simulation on smart grid model with different inputs in various scenarios. By observing of supporting facility tool modelling and simulating the SoS with legacy system.

Data from simulation of RUMIs, services, Intrusion and Fault Detection in Smart Grid were taken by simulating attacks such as reset and flooding attacks are taken for assessment.

The results of the test beds on Failure and Intrusion detection in Smart Grid can be found in Deliverable [7]. The main results of the test beds are summarized in the following. Even if the test bed highlights cyber attack detection and accidental faults, however the IFD tool provides an high adaptability to all types of attacks and faults detection basing on the customization of detection rules.

The operator can observe the experiment through a real time graph visualization provided by the LiveGraph (see Figure 14 and Figure 15) open source tool that has been integrated in the IFD and can be activated clicking on the Graph View control button on the IFD GUI.

Figure 14 depicts the plots related to the indicators values received in input by the IDF tool. The horizontal axis represents time (in seconds form the experiment start) while vertical axis represents the values of the indicators. The operator can selects or deselects the Indicators to be shown in the graph using the Data series settings panel.

Figure 15 shows the RTT indicator with its relative adaptive threshold evaluated by the SPS algorithm. Examining this graph it is possible to recognize the anomalies detected in RTT occurring in correspondence of the peaks of the RTT values in which the limit of the upper bound is exceeded.

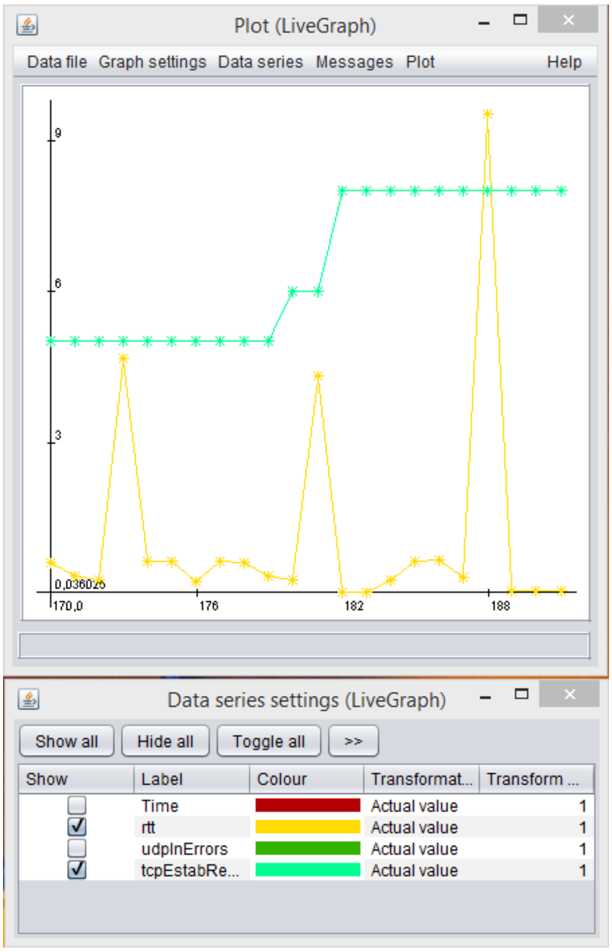

Figure 14 - Indicators values Graph

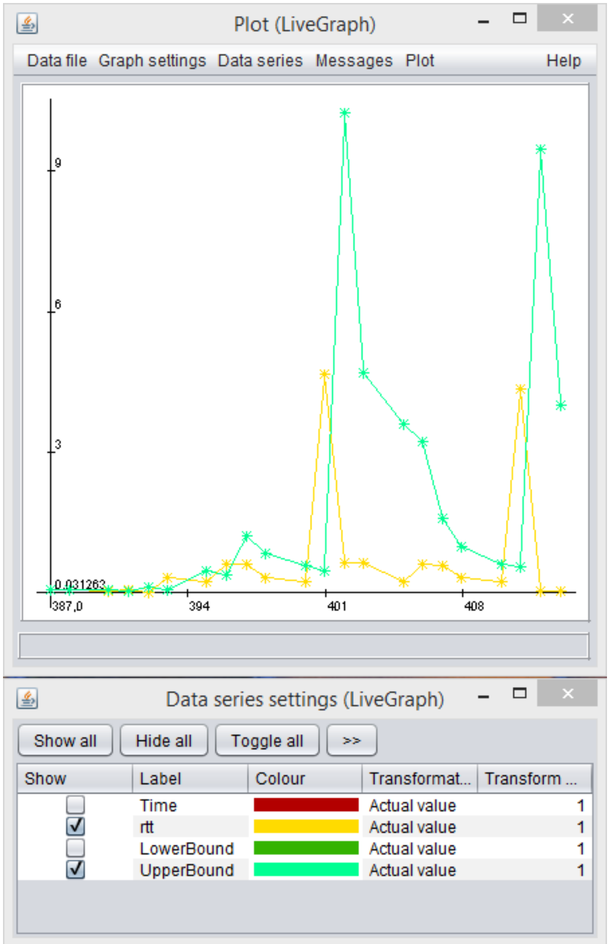

Figure 15 - Indicator RTT with relative threshold

2.2.6 <u>CHECK</u>

# WHAT	# DO	Check condition if the Assessment objectives are satisfied	Result of check
1,7	<u>DO #HOW 1</u>	Does the profile consider a clock with time granularity, accuracy, and stability. Does the case study achieve global time using the resilient master clock.	Satisfied
2,3	<u>DO #HOW 2</u>	As detailed in D2.3, Section 4.1.1, in the literature different attempts exist to apply SysML approaches to specific viewpoints that we deemed essential in supporting the design of SoS. These approaches	Satisfied

# WHAT	# DO	Check condition if the Assessment objectives are satisfied	Result of check
		have shown the utility of adopting SysML formalisms to model architectural aspects of SoSs thus supporting different types of analysis and a first step towards executable artifacts that can be automatically derived. Although these approaches provide detailed insights for different viewpoints aspects, it was still missing (i) an homogeneous synthesis at a more abstract level of key design-related SoS concepts and (ii) and a viewpoint-based vision. With the AMADEOS SysML profile we brought this perspective in one single consistent reference model, thus providing solutions to specific design problems while still keeping the required interconnections among viewpoints.	
4	DO #HOW 3	Does the case studies use blocks from AMADEOS evolution and dynamicity viewpoint.	Satisfied
5	DO #HOW 4	Does the case studies use AMADEOS viewpoints such as Dependability and Security	Satisfied
6	DO #HOW 5	Does the case studies use blocks from AMADEOS emergence viewpoint.	Satisfied
8	DO #HOW 6	Have participants used the supporting facilities tool for designing SoS on different platforms (browsers, Operating Systems)	Satisfied
9	DO #HOW 7	Does the building blocks in architectural framework cover Monitoring, Analysis, Planning and execution. Can the effectiveness of the MAPE building block in case studies be observed	Satisfied
10	DO #HOW 8	Is the IFD tool able to detect cyber attack detection and accidental faults. Is the EV-charging simulation able to detect attacks (passive or active) to the charging network	Satisfied

2.2.7 REPORT

This Section describes the structure of the Assessment Report. The complete Report of Assessment plan is reported in Section 3.

The Assessment Report is represented in the Table 7.

# WHAT	Description of WHAT	Type of Assessment	Results	Remarks

Table 7 – Assessment Report Structure.

In the follow will be described in details the fields of the Table 7:

- Column **# WHAT**: contains the number corresponding of the id of each item of WHAT clause reported in Table 1
- Column **Description of WHAT**: contains the description of the corresponding # WHAT of each WHAT clause reported in Table 1
- Column **Type of Assessment**: contains the key methods used for performing the qualitative assessment (Example: Direct Observation, Case Studies, Qualitative Interview, see also Section 2.1)
- Column **Result**: contains the final assessment of WHAT clause
- Column **Remarks**: An optional field to provide any extra information regarding the WHAT clause and its assessment.

3 ASSESMENT RESULTS OF AMADEOS METHODOLOGY

Here we will describe the results of assessment (all evidences, HOWs in detail as mentioned in earlier section).

# WHAT	Description of WHAT	Type of Assessment	Results	Remarks
1	Fielding industrial SoS, where the handling of time will be in the centre of the analysis, in the domain of transport, smart grid application and a summary of the common findings.	Qualitative/Quantitative	Time has been well formalized in the conceptual model, and having a global time has proven to be very beneficial in maintaining synchronization between various CS and has simplified the design of Smart Grid SoS. Having a global time has proven to be very beneficial in maintaining synchronization between various CS and has simplified the design of Smart Grid SoSs. TNL has performed an internal qualitative assessment of the entire AMADEOS framework. The specific viewpoints of Time and Dynamicity were recognized as very relevant extensions	
2	A Conceptual Model for generic SoS	Qualitative	Conceptual model of generic SoS involving various viewpoints has reduced cognitive complexity and simplified the design of SoS.	
3	An architecture framework for generic SoS	Qualitative	The proposed architectural framework was found to be effective in guiding the SoS designer to follow a process and use "meta-requirements" to design an AMADEOS based SoS.	
4	Formalization of evolvability and dynamicity in SoS	Qualitative	Evolution has been well formalized in the conceptual model and the SoS profile. It covers evolutions (managed and unmanaged),	

# WHAT	Description of WHAT	Type of Assessment	Results	Remarks
			which complies with goal and improves business value. Evolution may be modified by a CS/SoS and is affected by environment.	
5	A methodology for the design of SoSs that puts a focus on timeliness, complexity management, evolution, agility, dependability, security and emergent properties of the developing artefact.	Qualitative	While designing the case studies the Conceptual model, the SoS profile, and the supporting tools have been found to simplify and SoS design and focus on viewpoints such as time, evolution, dependability, and security properties of SoS.	
6	Management of emerging properties in SoS.	Qualitative	Emergence has been well formalized in conceptual model and using the supporting tools it was possible to specify various kinds of emergence phenomenon (explained and unexplained) and its causes using various kinds of behaviours.	
7	Impact of Introduction of global, synchronized time into SoS model & SoSE.	Qualitative and Quantitative	The use of RMC as a building block in case studies has been very effective in maintaining global time	
8	Supporting prototype software tools for the above-mentioned methodology based on UML extensions.	Qualitative and Quantitative	Supporting facility tool developed for AMADEOS has been found to be very useful and effective in modelling and simulation of SoSs.	
9	Building blocks and algorithms to assess the dependability, security and performability of a SoS.	Quantitative	The building blocks used in the case studies involving IFD has been found to detect failures and attacks.	
10	Modeling and simulation results of case studies from the domains of smart grid automation.	Qualitative and Quantitative	Using the AMADEOS concepts and supporting facility tools, the Smart Grid case study consisting of Ev-Charging, Household Management and Medium Voltage Control has been designed and simulated.	

4 LESSONS LEARNT FROM THE PROOF OF CONCEPTS

Modeling complex and pervasive infrastructures as the one used as case study clearly highlights how the support of a precise conceptual model and of specific tools for its instantiation is fundamental for a sound and comprehensive codification of the various properties of the whole.

At design time the identification of causal loop in the lower levels of the hierarchy, enabled by the support for simulation through model execution, is a mandatory step to identify possible emergent behaviors at the higher levels that may lead, also in future evolution of the system of systems, to a violation of system requirements. A correct representation of the environment is necessary.

Global time Awareness and monitoring are fundamental to early detect and to contain the effect of detrimental emergence phenomena at run time.

4.1 THE USE CASE APPROACH AND THEIR ANALYSES

From the simulation of the EV-Charging scenario, extensive amount of data was collected and analyzed. In general, we found the tooling to be extremely useful, and, for example, several emergent behaviours were discovered in the data that, if proven true, would lead to significant challenges to the electrical grid. Furthermore, the aspects that the SoS was designed to test (primarily security and dependability) were successfully tested and validated. We plan to exploit these results in two manners: First to follow up on the results and perform more experiments after the end of the project, leading to publications, and secondly both Thales and ENCS are planning to make the simulator code available online, so other researchers can validate and use the same simulation code.

4.2 BENEFITS OF THE AMADEOS APPROACH

The main benefit of the AMADEOS approach can be easily seen in the results from the simulations: The AMADEOS architectural framework and associated tools allow an SoS architecture to be comprehensively designed and a simulation extracted that can be tested. This allows system architects to quickly test hypothesis regarding future systems and determine what attributes will lead to advantageous or poor results.

4.3 LIMITATIONS OF THE AMADEOS APPROACH

A number of minor problems and limitations were discovered during the modelling and testing of the testbed. However, most of these issues, such as the best manner to model aspects, were solved via communication with the tool developers. Other aspects, such as how to ease development of the constituent systems, via source control, are in development and future versions of the tools will be substantially easier to manage. The single limitation regarding the AMADEOS tooling that was not (directly) addressed was the lack of a simple method to support simulation-stepped runs, to allow faster than real time simulation runs. However, the SimPy toolkit provided this functionality and, due to the ease of integration into the AMADEOS generated code, as the generated code is written in Python, as is SimPy. Furthermore, the combination of SimPy and the AMADEOS Blockly tool provides a comprehensive, yet easy to use, simulation system.

5 CONCLUSION AND FUTURE RESEARCH DIRECTIONS

The importance of the areas of the often time- and safety-critical embedded and cyber-physical systems (CPS) will continue to grow with the increasing pervasiveness of ICT and the development of the Internet of Things.

5.1 CHALLENGES

The challenge is to design, program and implement highly distributed and connected digital technologies that are embedded in a multitude of increasingly autonomous physical systems, with various dynamics and satisfying multiple critical constraints including safety, security, power efficiency, high performance, size and cost.
Such combination of several cyber-physical systems in SoS gives rise to unpredictable behaviour and emergent phenomena.
A significant improvement in design and programming of CPS is therefore needed including a "science of system integration and evolution".

5.2 FUTURE RESEARCH ACTIVITIES

Based on the AMADEOS concepts and project results the future research activities should address:

- Implementing time-triggered virtualization and hypervisor based technology in CPSoSs for improved performance w.r.t. temporal predictability, adaptability, scalability, enhanced composability, complexity management, evolution, security, safety and the management of hardware obsolescence.
- Establishing a new approach for designing and developing evolving CPSoSs, which considers novel architectural building blocks (such as the TTVM) and combines them with legacy parts.
- Providing a holistic model-centric approach based on the combination of models supported by experimental fault injection for predicting and assessing major CPSoSs properties, both during design/development phase and at runtime, supporting system evolution while guaranteeing key CPSoSs properties such as safety and security (at the same time).

The above-mentioned activities should be fundamental for enabling technologies that:

- reduce the cost of development
- cost of ownership
- cost of managing evolution

While AMADEOS project focused on the integration of “dynamicity”, “evolution” and “time-awareness” into CPSoS architecture for dependable multi-criticality level system, the introduction of real-time virtualization, provides a way to a paradigm shift in the design for near autonomous or autonomous SoS/CPSoSs controlling emergence, evolution, safety and security. The real-time virtualization providing both flexibility and complete isolation among different applications shall be a robust and verifiable element of the design.

6 REFERENCES

[1] Project AMADEOS. Deliverable D1.1 “SoSs, Commonalities and Requirements”.

[2] Project AMADEOS. Deliverable D2.2 “AMADEOS conceptual model”.

[3] Project AMADEOS. Deliverable D2.3 “Conceptual model - Revised”.

[4] Project AMADEOS. Deliverable D3.1 "Overall Architectural Framework".

[5] Project AMADEOS. Deliverable D3.2 “Amadeos methodology, mechanisms, and implementations”.

[6] Project AMADEOS. Deliverable D4.1 "Case Study and Use Cases Specifications".

[7] Project AMADEOS. Deliverable D4.2 “Case Study Realization”.

[8] Project AMADEOS. Deliverable D4.3 "Amadeos simulator framework".

[9] Project AMADEOS. Deliverable D4.4 “Design of Resilient master clock".

[10] O. Etzion and P. Niblett, “Event Processing in Action”, MANNING Greenwich (74° w. long.), 2011, pp.68-73.

[11] Esper Team and EsperTech Inc. [http://esper.codehaus.org], “Esper Reference version 4.9.0”, 2012.

[12] Secure! Project, http://secure.eng.it/.

[13] SmartC2Net Project, http://www.smartc2net.eu/.

[14] R. de Lemos, «Software engineering for self-adaptive systems: a second research roadmap.,» chez *Software engineering for self-adaptive systems. Lecture notes in computer science, Vol. 7475*, Springer, 2013, pp. 1-32.

[15] H. Kopetz, O. Hoeftberger, B. Froemel, F. Brancati et A. Bondavalli, «Towards an understanding of emergence in systems-of-systems,» chez *System of Systems Engineering Conference (SoSE)*, 2015.

[16] Adler, P. A. and P. Adler (1994). Observational Techniques. Handbook of Qualitative Research. N. K. Denzin and Y. S. Lincoln. London, Thousand Oaks, New Delhi, Sage: 377-392.

[17] Patton, M. Q. (1990). Qualitative Evaluation and Research Methods. Newbury Park CA, Sage Publications.

[18] Woodhouse, P. (1998). People as Informants. Finding Out Fast: Investigative Skills for Policy and Development. A. Thomas, J. Chataway and M. Wuyts. London Thousand Oaks New Delhi, Sage: 127-146.

[19] Stake, R. E. (1994). Case Studies. Handbook of Qualitative Research. N. K. Denzin and Y. S. Loncoln. Londo, Thousand Oaks, New Delhi, Sage: 236-247.

[20] Thomas, A, J. Chataway and M. Wuyts eds (1998) Finding Out Fast: Investigative Skills for Policy and Development. London, Thousand Oaks, New Delhi, Sage: 307- 332.

[21] Roche, C. (1998). Organizational Assessment and Institutional Footprints. Finding Out Fast: Investigative Skills for Policy and Development. A. Thomas, J. Chataway and M. Wuyts. London, Thousand Oaks, New Delhi, Sage: 173-200.

[22] Roche, C. (1999). Impact Assessment for Development Agencies: Learning to Value Change. Oxford, Oxfam Novib.

[23] Denzin, N.K and Y. S. Linconl, Eds (1994) Handbook of Qualitative Research. Thousand Oaks, London, New Delhi, Sage.

[24] M.Mori, A.Ceccarelli, P.Lollini, A.Bondavalli, B. Fromel « A Holistic Viewpoint-Based SysML Profile to Design System-of-Systems», 2016 IEEE 17th International Symposium on High Assurance Systems Engineering.

[25] Joseph M. Juran, A. Blanton Godfrey « Juran's Quality Handbook », 1998, Fifth edition, McGraw-Hill.