

Extending GOReM through the RAMSoS method for supporting modeling and virtual evaluation of the Systemic Risk

Angelo Furfaro, Teresa Gallo, Alfredo Garro,
Domenico Saccà, Andrea Tundis

Department of Informatics, Modeling, Electronics and
Systems Engineering (DIMES), University of Calabria,
Via Ponte P. Bucci 41C, Rende (CS), 87036 Italy
{a.furfaro, t.gallo, a.garro, sacca, a.tundis}@dimes.unical.it

Simona Citrigno, Sabrina Graziano
Centro di Competenza ICT-SUD,
Piazza Vermicelli,
87036 Rende (CS), Italy
simona.citrigno@cc-ict-sud.it,
sabrina.graziano@cc-ict-sud.it

Copyright © held by the authors.

Abstract— Recently, due to the increasing complexity and wider adoption of heterogeneous systems, the management of security properties, vulnerabilities and risks of systems by integrating and structuring existing components, is becoming more and more crucial. A particular aspect to be considered is the Risk Analysis and, specifically, the analysis of the Systemic Risk. This risk derives from the interdependence of the system under consideration, from services provided by other systems and, in general, from the interactions among them. In fact, it may happen that an adverse event, which occurs at a certain system that is not properly controlled, can cause dangerous effects that, through its propagation to other interconnected systems, would/could compromise their operation. Thus, suitable engineering approaches need to be exploited to prevent and manage the risks arising from the integration of system components so as to increase the security of systems, data and even human life. In this context, the paper proposes specific extensions of a Goal Oriented methodology for Requirement Modeling, called GOReM, through the RAMSoS method, natively conceived for supporting dependability analysis. Such combination enables the modeling and the evaluation of the Systemic Risk centered on agent-based simulation techniques. The combination of RAMSoS and GOReM is experimented on a case study concerning an online payment service, by evaluating the impact of the failure of a single component on the overall system.

Keywords— Cybersecurity, Modeling and Simulation, Requirement Engineering, Systemic Risk Analysis.

I. INTRODUCTION

In recent years, the global crisis has shown that the benefits of globalization are increasingly accompanied by a growing interdependence and interconnection of systems and services, bringing out new vulnerabilities coming from unexpected directions. Global risk can cause a significant negative impact on a number of countries and companies, showing a systemic nature [14]. In this view, it is important to distinguish between the idiosyncratic shock which affects only a single institution or activity, respect to the systemic risk that can cause the rupture of an entire system (social, political, economic, technological, etc.), causing a damage of

remarkable entity. Its main features are: (i) small fragilities that combine to produce a more extensive failure; (ii) risk sharing or contagion, when a loss triggers a chain of other losses; (iii) hysteresis, when the system is unable to recover after a shock, [10]. The causes that lead to systemic events reside primarily in the influence that the various actors in the network have with each other; furthermore the systemic importance of the various actors is not determined by their size, but from the correlation degree among them. Similarly, it is not always true that a negative event of large dimensions can be always defined as systemic. In fact, the propagation mechanism can be realized not only through the direct exposure to a negative event caused by the shock, but also indirectly. In this context, it is interesting to understand how it is possible to modeling actors and factors arising from systemic risk in order to fully consider them in the different phases the of risk analysis.

In this context, the paper aims at investigating in such direction by exploiting engineering tools for representing relationships among systems/services and observing their behavior. Specifically, the adoption of the Systems Engineering approach combined with Modeling and Simulation techniques are used to catch how and which entities of the overall system influence the operation of the entire system and, as a consequence, the evaluation of the Systemic Risk. In particular, the combination of a Goal Oriented methodology for Requirement Modeling, called GOReM [4], with the RAMSoS method [8], natively conceived for supporting systems dependability analysis, is provided. Such combination enables the modeling and the evaluation of the Systemic Risk by exploiting an agent based simulator that has been ad-hoc implemented.

The rest of the paper is structured as follows: Section II presents the related work and highlights the main research challenges related to the systemic risk in the cyber-security domain; the combination of the GOReM and RAMSoS methods are presented in Section III. A case study concerning an online payment service is described in Section IV, whereas the simulation-based evaluation is presented in Section V. Finally conclusions are drawn in Section VI.

II. A PANORAMA ON THE SYSTEMIC RISK

A. Overview on the Systemic Risk

As mentioned above the Systemic Risk is intended as a risk deriving from the interdependence between the main system, object of the analysis, and the services provided by other systems and, in general, by the interactions between them. It is possible to define the systemic risk as “any set of circumstances that threatens the stability of or the public trust in the system” [2]. In this way, there is a strong link between systemic risk and operational risk and it is interesting to understand how it is possible to explicitly modeling factors deriving from systemic risk in order to fully consider them in the different phases of operational risk analysis and treatment.

Companies inadvertently expose themselves to risks outside of their structure, by outsourcing, interconnecting or divulging their data to an increasingly complex and inscrutable networks’ system. Some risk factors have been identified and published on the “Zurich Cyber Risk Report”, and, in particular, seven IT risks have been identified that could threaten a systemic shock: internal corporate network, outward counterparts and affiliates, supply chain and outsourcing contracts, disruptive technologies (IoT in the first place), critical infrastructure and external shocks [15].

These seven risks can be grouped in three areas “*Near*”, “*Everywhere*” and “*Distant*”. The “*near*” area is related to the usage of contracts, SLAs, internal corporate controls and resiliency within a company. The “*everywhere*” area includes all those companies that may have contractual relationships with other companies around the world, so the risks are not generally controlled by individual contracts, but by companies and governments through standards, regulations, global and national governance. The “*distant*” area is then related to all those external risks to which individuals or group of companies may not have any influence. Risk control coming from external shocks is almost entirely in charge of governments, intergovernmental organizations and transnational organizations [15].

B. Systemic Risk in the Financial field

Systemic risk in the financial sector can be thought as the probability that a failure of a significant portion of the financial sector can occur, which can lead to a reduction in credit availability. The materialization of such event is likely to generate negative effects on the real economy. Systemic risk in the financial sector is essentially related to the risk of infection among financial institutions, which could generate a potential destabilization of the entire financial system. Some negative externalities, or inappropriate behaviors, generating damaging effects on the financial market status, have great impact on the increasing of the systemic risk. Several preventing approaches have been proposed: making use of suitable financial stability or strength indicators; measuring the existing correlations between financial institutions; usage of legislative bodies aiming at regulating the activities of the actors in the financial sector to minimize such kind of risks.

Four main reasons determining negative effects on a system have been identified (the focus is on negative externalities, i.e. economic and financial behaviors which

affect the overall market trend, and influencing systemic risk growth):

Informational contamination. Rapid news propagation having influence on financial topics leading to considerable mismatches on assets and liabilities maturities. A striking example of the materialization of such event is the failure of Lehman Brothers, which led, from one side, Merrill Lynch to merge with Bank of America, and, on the other side, Goldman Sachs and Morgan Stanley to become ordinary banks, causing in this way the collapse of US real estate stocks. The involvement of important institutions in the crisis is relevant for the propagation of negative information.

Loss of specific and confidential information about the creditworthiness of the debtor. The failed credit bank customers will have greater difficulty in obtaining a credit to new banks. This is because new banks can apply more restrictive policies for granting credit to new customers since there is scarce information about them.

Debt-Credit relations between banks. Credit institutions and financial intermediaries are inclined to work more closely among themselves at commercial level. The risk of a crisis spreading in the whole financial system can be increased by the interactions between banks and intermediaries, which can be related not only to the interbank market, but also to a large sector of derivatives markets, included CDS (Credit Default Swap), guarantees, brokerage services, etc.

Liquidity spiral. This negative externality occurs when financial market operators, instead of selling financial assets for gaining liquidity, use different strategies to restrict the new credit extension, that means, for example, making a credit rationing having high-margin/cuts, or increasing the interest rate for the grant allocation. These activities can reduce prices and outputs and, can increase the possibility of failure in accessing the loan. This kind of problem is caused by an extreme exposure to risk of the liquidity shortage by financial institutions, which make use of high debt strategies.

In the end, the negative propagation effects can be greater when the failure is related to large institutions having different interconnections and in the presence of a not transparent market structure (OTC markets, not characterized by the typical requirements for regulated markets). Government institutions implicitly support and foster financial institutions to increase their size and interconnections, so that they can increase the possibility of being saved in time of crisis, since they are “too big to fail”.

C. Systemic Risk in the Information Technology field

Microsoft has proposed the creation of a G20+20 Cyber Stability Board, that means, 20 governments and 20 companies, operating in the information and communication technology, which should work in synergy to draw up a set of basic principles ensuring, from one side, an 'acceptable behavior' in cyberspace and, on the other side, some “guidelines” to improve IT risk management.

The following recommendations about potential systemic risk impact in IT, can be useful for both large and small organizations to survive to a potential cyber shock, and can be

considered as a kind of “shock absorber” that can potentially reduce the magnitude of the shock: (i) improving the resilience and incident response at system level; (ii) expanding security concepts aim at involving third-party suppliers as much as possible; (iii) providing targeted subsidies; (iv) considering other measures, such as “Stability Board” and the “G-SIFIs” requirements.

For small business enterprises there are three categories of recommendations: *Basic*, *Advanced* and *Resilience*.

Basic. The main 5 crucial recommendations of the 20 Critical Security Controls SANS, are taken in consideration: (1) Whitelist application - organizations should enable computers to perform only a limited set of pre-approved programs; (2) Standard system configurations usage - computers with a few standard configurations are less expensive and easier to defend; (3) Patch application software and (4) System software within 48 hours - large companies should check software on a regular basis looking for any bugs in order to drastically reduce the opportunities of vulnerabilities exploitation by hackers; (5) Reduction of the number of users having administrative privileges.

Advanced. Broadening risk horizon - taking in consideration counterparts, contracts and outsourcing agreements, and critical infrastructure, each part should be at least partially controlled by contracts, agreements on service levels, in-depth site visits and audits; Cyber Insurance usage - to transfer IT risks, particularly risks associated with third-party data breaches or business interruption; Requiring standard and more resilient and safe products to key suppliers; Acquiring at management level a broader view on IT risks.

Resilience (the ability of large companies to recover from interruptions in the shortest time as possible): Redundancy - redundant power and telecommunications suppliers, ISP alternately connected to the peering point, work-around with little dependence on IT in order to provide some alternative solutions when Internet access is off; well defined Response to incidents and business continuity planning - standard operating procedures, clear objectives based on metrics, quantification of the needed time to detect an accident or an intrusion in the system; Simulating scenarios and security training - analyzing the most likely and the most dangerous cyber risks and test their Security Response Team, together with the company management in order to build a historical memory for incident response.

III. COMBINING GOREM AND RAMSOS METHODS FOR MODELING AND SIMULATING SYSTEMIC RISK

A. GOREM Overview

GOREM (Goal Oriented Requirements Methodology) is a lean, easy to master methodology for capturing and maintaining up-to-date requirements of large systems operating in complex application domains. GOREM first definition [4] was done in 2014, for supporting the requirements engineering activities in an industrial research project [5, 6, 7] where numerous stakeholders, coming from several industrial and academic domains, with different goals, skills and languages had to cooperate. Since then, GOREM has

been incrementally improved through its actual exploitation for better supporting the requirements modeling aspects and it has been experimented in other real industrial research projects. Moreover, a set of lessons learned have found a response in the current proposal. The full-fledged version of GOREM methodology is described in this section. The GOREM method is centered on the UML notation, which is easy to use and it simplifies concepts sharing with a wide variety of stakeholders. The resulting requirements modeling activity is recognized by the actual users to be easier and more effective than their past requirements elicitation activities.

GOREM consists of three main phases, each of which is devoted to modeling specific aspects of a requirement engineering process: *Context Modeling*, *Scenario Modeling*, *Application Modeling*; specifically:

- in the *Context Modeling phase*, the stakeholders are identified along with their objectives as well as the dependencies among softgoals; moreover, the rules and regulations that govern the business context under analysis are identified and documented.
- in the *Scenario Modeling phase*, different business scenarios are derived from the Context model, in terms of roles that are played by the stakeholders involved in the modeled scenario, their specific goals and their dependencies, and the rules and regulations that govern each elicited business scenario. Furthermore specific analyses that show the strengths, weaknesses, opportunities and threats are also performed to guide and support strategic decisions at business level related to the future work.
- in the *Application Modeling phase*, one or more application scenarios are introduced in order to specify main functionalities which should be provided by a single business scenario resulting from the previous phase.

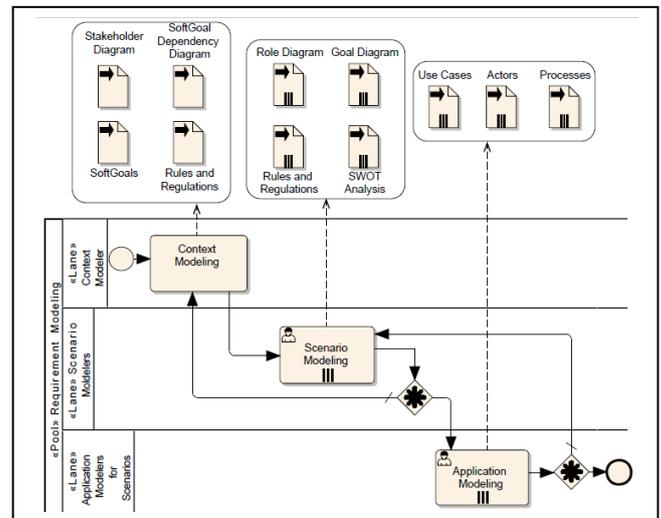


Fig. 1. The GOREM process

Multiple scenarios are concurrently set down. A sketch of the reference process for the GOREM method along with its main work-products is shown in Figure 1.

The lessons learned from the experience derived by exploiting the GOREM method on important research projects by cooperating with industrial partners such as ACI Informatica [1] and Poste Italiane [12], allowed to catch not only strengths but also weaknesses of the method, which have been considered to refine and improve GOREM. The most interesting and relevant “lessons learned” are reported in the following.

Lesson 1: human interactions and cooperation. It is probably the most difficult task due to different skills, backgrounds and knowledge which lead to big misunderstandings, lethal for establishing system requirements. It is likely to encounter mistakes when a new application domain is being explored because of: (i) misleading interpretation, due to the coexistence of different interpretations of stakeholder goals and requirements, that usually happens when people have different skills and the same concepts are interpreted differently according to the stakeholder’s background; (ii) conflicting specifications, when specific strategies, that could potentially create strong disadvantages in other application scenarios are adopted in order to reach a specific goals in a specific application context; (iii) late discovery of redundancy, when in advanced development project stages the same concept is described and represented differently several time or different terminologies is used for describing the same concepts (iv) fragmentation of efforts; (v) weak focus on objectives for achieving the desired goals and being competitive and effective; (vi) partner coordination, when there exist different partners having different objectives to reach; (vii) work-product integration, when there is a need to integrate, harmonize and handle deliverables, services and products coming from different tasks.

Lesson 2: cross-domain aspects. There are some recurrent features that might be identified once for all as well as common characteristics for each domain of interest that have to be considered and properly represented, which in turn arise questions that need to be answered, such as:

- *space:* Is the considered context model influenced by the location and the territorial extension (e.g. regional, national, international, members states)?
- *time:* Is the considered context model influenced by temporal aspects (e.g. a new law replaces partially or totally a previous one)?

Whereas there are some features that need to be identified and analyzed according to the specific scenario, such as:

- *subject:* who/what is the subject of the described context?
- *user profile:* are the user preferences/personal features represented in the context model? Does the system describe the user’s characteristics one by one or does it provide a role-based model of user classes?

- *context history:* does the current context state depend on a previous ones?

Lesson 3: legal aspects. The specific context model and the different business scenarios are handled by several Rules and Regulations that might be in conflict. As a consequence, it is important for modeling a context and any specific business scenario, to understand which laws are involved, which is a policy as a “standard” or a best practice as a “guideline” that can be adopted or not, depending on the stakeholders needs. In addition, there are stakeholders of specific customers that can have a set of internal policies which, in turn, should be considered and their eventual contrast with some laws or requested best practices should be discovered and resolved. Finally, as a desired service can be used in different Nations, the requirement model has to analyze and manage the legal usability of a service for a given customer. Furthermore, requirements engineering processes should manage legal aspects by continually monitoring their changes over the time, during the overall system lifecycle.

Lesson 4: tracing evolution. Business context, scenarios and applications can evolve because of their dynamic nature. It is important to have some tracing mechanism that allows knowing which application model version from which scenarios model version has been derived and this last one to which business context model version refers to. For big and continuously evolving system engineering process, this is of fundamental importance and especially for maintaining control and governing the system evolution along its life.

Lesson 5: inter-scenarios dependencies and reuse. Quite often, business scenarios evolve with a specific team of analyst/designer (sub)domain experts that have the objective to go ahead following their requirements engineering for specific final services. This can lead to duplication of work and, worse, to services which do the same thing (same requirements) but in a different way. This is often difficult to discover and create customer dissatisfactions. This happen, for example, when the same stakeholder has two different goals which belong to two different scenarios, but the two application models reaching the two goals, share many “what to do” but unawares.

In the light of the above reported lessons learned during the method exploitation, starting from *Lesson n.1*, an updated and refined version of the GOREM method in [4] is provided.

1) *The Context Modeling phase*

The Context Modeling phase aims at clearly representing the reference business domain for the project under consideration. The work-products of this phase are: a Stakeholder Diagram, which shows a (hierarchical) specification of all the involved stakeholders, each of which is in turn characterized by a set of Softgoals they intend to pursue; a Softgoal Dependency Diagram, which shows the relationships among Softgoals, (i.e., contribute, hinder, include, extend, generalize); a Rules and Regulations report shortly describing the rules and regulations governing the Context, distinguishing between Laws, which can be National or International, and known used Policies and Best practices.

Table I shows symbols already used in the first version of the methodology, while table II shows the identified and considered types of rules and regulations.

TABLE I. THE CONTEXT MODEL - MAIN CONCEPTS

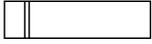
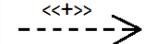
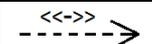
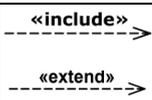
Concept	Graphical Notation	Description
Stakeholder		The UML Actor symbol extended through a yellow-filled head stereotype
Softgoal/Goal		The SysML[16] Requirement native construct
Contribute Dependency		A UML Dependency symbol extended with a "+" stereotype
Hinder Dependency		A UML Dependency symbol extended with a "-" stereotype
Include/Extend Dependencies		The UML native dependencies applied among softgoals or goals
Generalize Dependency		The UML Generalize Dependency native symbol

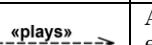
TABLE II. THE CONTEXT MODEL – RULES AND REGULATIONS

Type	Description
Best Practice	Best practice is considered a business buzzword, used to describe the process of developing and following a standard way of doing things that multiple organizations can use to maintain quality. It is not mandatory and can be based on self-assessment or benchmarking.
Policy	A Policy is a deliberate system of principles to guide decisions and achieve rational outcomes. It is a statement of intent, and it is implemented as a procedure or protocol.
National Laws	National laws are valid and affect the State or Country that has enacted them.
International Laws	International laws are enacted by specific Authorities and they govern the behavior of the Members States belonging to a specific community according to specific agreements.

2) The Scenario Modeling phase

The Scenario Modeling phase specializes the Context Model through the identification of evolutionary scenarios that have to be modelled within the context of interest. Such scenarios are identified through an analysis that takes into account the roles played by stakeholders in each scenario, by indicating the specific Goals related to some Softgoals in the context model and the Rules and Regulations that govern the scenario. Table III shows symbols used for roles and for the associations with the stakeholders.

TABLE III. THE SCENARIO MODEL – MAIN CONCEPTS

Concept	Graphical Notation	Description
Stakeholder's Role		The UML actor symbol extended through a pink-filled head stereotype
Plays Dependency		A UML Dependency symbol extended with a "plays" stereotype

The SWOT Analysis activity [11], represented in a matrix as showed in Table IV, provides an assessment of internal and external factors that may affect the scenario and may support decisions whereas to continue with the next phase, that is the *Application Modeling*. For Goals and dependencies diagram, symbols in Table I are used.

TABLE IV. THE SCENARIO MODEL – SWOT ANALYSIS

	HELPFUL	HARMFUL
Internal Origin	Strengths: what are the strengths (i.e. benefits controllable)	Weaknesses: what are the weak points (i.e. disadvantages controllable)
External Origin	Opportunities: possible opportunities (i.e. advantages not controllable)	Threats: potential threats (i.e. disadvantages not controllable);

Rules and Regulations selection activity considers which rules and regulations, identified in the Context Modeling phase, must be considered in the modelled scenario, by identifying them with a structured ID, describing them, specifying if they are laws, policies and best practices, indicating the adopters, and warning possible dependencies with other considered rules. In particular, GOREM uses the matrix formats, showed in table V. This is an improvement introduced and allows to better manage the issues discussed in lesson 3 related to legal aspects.

TABLE V. THE SCENARIO MODEL – RULES AND REGULATIONS

Identifier	Rule/ Regulation	Type	Location / Adopter	Warnings
Structured ID	Description	Policy/ Best Practices/ National Law/ International Law	Locations and/or names of known adopters	List of identifiers of other rules and regulations which can have influence on its application

3) The Application Modeling phase

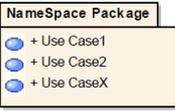
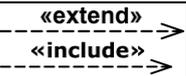
Starting from the scenarios defined during the previous phase, in the Application Modeling phase, a set of specific business scenarios might be identified. This phase defines application scenarios that are used to specify in detail the capabilities to be provided in the specific scenarios identified in the previous phase, along with main use cases description, actors and processes. In particular, each main use case may become a service to be developed as a research prototype and/or developed and engineered as part of a more complete industrial system.

In addition, some processes can be specified using UML or BPMN notations [13].

Table VI shows basic used symbols in modelling an application scenario. The Package is a Namespace of use cases, which are not in the scope of the application which is modelled, but are assumed that they exist in some different Application model, even in an Application model obtained

from a different Scenario Model, while in this Application Model they have to be identified and extended through the standard “extend” UML relationship.

TABLE VI. THE APPLICATION MODEL – MAIN CONCEPTS

Concept	Graphical Notation	Description
Application Scenario's Actor		The UML actor symbol extended through a blue-filled head stereotype
Use Case		The UML Use Case native symbol.
Package		The UML NameSpace for Use cases supposed already existent in another Application Model,
Extend /Include		The UML <<extend>>and <<include>>native dependencies among use cases

This is how GOREM is now responding to lesson n.2 cross-domain aspects and lesson n.5, Inter-scenarios dependencies and reuse. The corresponding work-products should be more precise and should indicate exactly to which use case of which scenario an extending use case refers to and the kind of needed extension.

Every UML based diagram can be enriched with the UML comment symbol which allows adding a description to all the GOREM diagrams. However, a textual description and complete information is located in the corresponding work-product.

Finally, concerning lesson n.4, tracing evolution, some shared existing policy of naming and versioning method/tool, for every model (context, scenario, application) and each of its work-products, must be used. In addition, some configuration management tool should be of help in maintaining the requirements evolution of the whole system [17]. This allows knowing exactly for each application model, which scenario model and context model refer to. In addition, whichever refinement for a model created in one of the three GOREM phases must produce a new model referring the model it wants to improve. Moreover, each application model, if implemented should refer to its development artefacts and releases in operation.

B. Combining RAMSoS and GOREM

RAMSoS [8] is an agent-based method that aims at supporting the dependability analysis of Systems of Systems (SoSs). It is conceived as an extension of RAMSAS [8], a model-based method for the reliability analysis of systems through simulation, based on UML/SysML for modeling the system structure and behavior, and on well-known simulation platforms, such as Mathworks Simulink and OpenModelica. The RAMSoS method defines three main phases, which in turn are divided into activities (see Table VII).

A full description of RAMSoS can be found in [8]; whereas Table VIII reports the main phases (Requirement Analysis, System Design, e System Risk Evaluation) that are identified by combing GOREM and RAMSoS for modeling the systemic risk aspects and supporting its analysis through agent-based simulation.

TABLE VII. PHASES, ACTIVITIES AND WORK-PRODUCTS OF RAMSoS

Phase	Activity	Work-product
SoS Structural Modeling	- Organizational Structure Modeling - Architectural Modeling	Organizational Model (MO) Architectural Model (AM)
SoS Behavioral Modeling	- Goal Modeling - Role Modeling	Goal Model (GM) Role Model (RM)
SoS Simulation Modeling	- Agent Modeling - Scenario Modeling	Multi-Agent Model (MAM) Scenario Model (SM)

In particular, some phases are complementary, some others use the output produced from a method as input for the other one. The resulting method will be exemplified through a case study in the next Section.

TABLE VIII. GOREM EXTENSIONS THROUGH THE RAMSoS METHOD

Phases	GOREM	RAMSoS	Description
Requirement Analysis	Context Modeling	-	Through GOREM it is possible to identify the involved entities: Stakeholders, Goals, Rules and Regulations, for the Systemic Risk Analysis.
System Design	-	SoS Structural Modeling	Starting from the entities identified in the previous phase, RAMSoS enable their formal structural and organizational representation as peer-to-peer or hierarchical entities.
	Scenario Modeling and Use Case Modeling	SoS Behavioral Modeling	GOREM is exploited for modeling the scenarios, roles and rules that characterize the scenario; the objectives to be achieved, weaknesses and strengths. By adopting RAMSoS, such Role Model can be exploited for identifying and defining tasks for achieving the identified objectives.
Systemic Risk Evaluation	-	SoS Simulation Modeling	Starting from the objectives defined in the Use Case Modeling phase of GOREM, the system is represented in terms of Simulation Agents that are used to simulate and evaluate the risk and its propagation among the involved entities.

IV. A CASE STUDY ON AN ONLINE PAYMENT SERVICE

The case study under consideration falls within the online payment services and in particular exemplifies the approach based on combination of GOREM and RAMSoS, adopted for systemic risk analysis applied to a service of *Electronic Payment Online (PEO)* of Poste Italiane. The main objectives of this study are: (i) The assessment of systemic risk, when there is a dysfunctional behavior in one of the service components, in terms of the propagation of a disservice among other components; (ii) impact of a service failure to the services.

A. Service Description, Risk Factors and Involved Actors

The PEO service is based on two services: *SMS Notifications* and *Payments and Transactions*, both designed to be used from smartphones and tablets. *SMS Notifications* allows to receive SMS messages on transactions made on a bank account or by “PostePay” card; whereas *Payments and Transactions* allows bank transfers, payment of bills, money transfer via MoneyGram, PostePay top up, or balance check and movements. In this context, the aim of this experience is the identification and the analysis of systemic risk factors linked to the PEO service. In particular, the risk of success or failure of the PEO service relies on two complementary services: *SMS Notifications* and *Payments and Transactions*, plus the IT Internal Infrastructure. A preliminary analysis shows that the *SMS Notification* service is linked to the *Mobile Service Provider* whose goal is to notify the user of the transaction (payment, charging, etc.). Whereas the *Payments and Transactions* is related both to the *Web Service Provider* that provides access to the Intranet / Internet and the *Energy Provider* that supports the entire infrastructure with the electrical service. An additional risk factor is related to the underlying IT infrastructure (hardware, servers, etc.).

In this context, the following risk factors: *IT Internal*, *Outsourcing and Contracts*, *Infrastructure Upstream*, are identified and described along with the related actors. In particular: (i) the *IT Internal* risk relies on the reliability of the *Internal IT infrastructure*; (ii) the *Outsourcing and Contracts* risk depends on the *WebServiceProvider* for supporting the monetary transactions; (iii) whereas *Infrastructure Upstream* risk is related to the availability of both the mobile notification service offers by the *MobileServiceProvider* and the electricity provided by the *ElectricityProvider*.

Furthermore, since the approach requires the input of information related to potential risk groups (e.g. contract type, involved partner), for each actor, the following risk groups have been identified:

- *IT-Internal-Infrastructure*: Good, Standard, Poor;
- *WebServiceProvider*: High, Medium, Low;
- *Energy Provider*: High, Standard;
- *MobileServiceProvider*: HighLevelOfService, StandardLevelOfService;
- *SMS Notification*: Good, Low;
- *Payments and Transactions*: LowRisk, HighRisk.

The output of this analysis is the risk level of the PEO service according to the different levels of risk of the other

services. It is estimated in terms of success and failure, where $Success = 1 - Failure$, therefore $Success + Failure = 1$. The higher the percentage / value of the Success, the lower the level of risk associated to it and as a consequence the lower the risk level of the PEO service. Vice versa the lower the percentage of the Failure variable, the lower the level of risk associated to it, and then the lower the risk level of the PEO service. In the following, the extended version of GOREM is employed for the modeling and evaluating the system above described.

B. Context Modeling

As described above, the context falls within the scope of online payment systems in which through a websites it is possible to make purchases, transfers of money etc. A particular important diagram of GOREM is the Dependency diagram (Fig. 2) that at the same time allow to represents the stakeholders, the goals that they are meant to achieve and dependencies (conflicts/extensions and so on among goals).

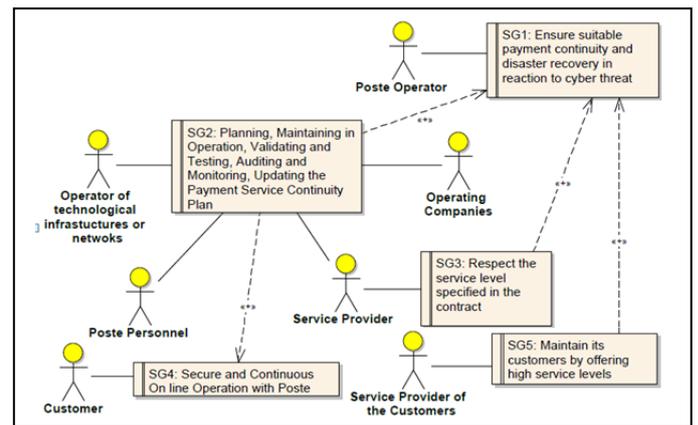


Fig. 2. Dependency diagram

C. Scenario Modeling

In this phase of the method, as it is shown in Figure 3, both the roles played by the stakeholders in each specific scenario are identified, and the goals related to each identified role are highlighted. Furthermore the dependencies among the Goals are shown in Table IX.

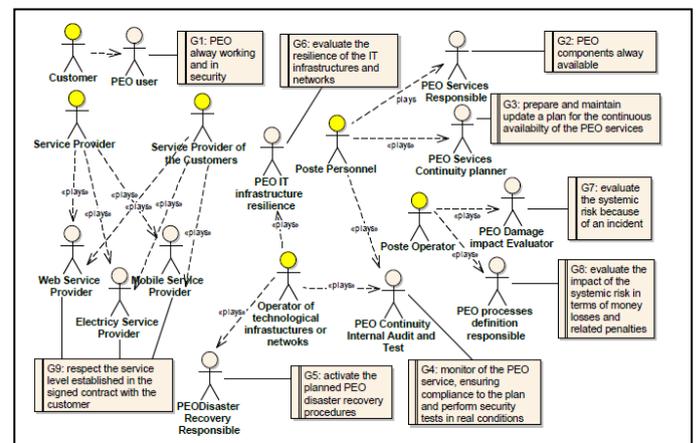


Fig. 3. Stakeholders, Roles and Goals

TABLE IX. STAKEHOLDERS, ROLES, GOALS AND DEPENDENCIES

Stakeholders	Roles	Goal	Dependencies
Customer	PEO User	G1	
Service Provider Service provider of the customer	Web Service Provider Electricity Service Provider Mobile Service Provider	G9	G9 contributes to G1
Poste Personnel	PEO Services Responsible PEO Services Continuity planner PEO Continuity Internal Audit and Test	G2 G3 G4	G2 and G4 contribute to G1
Operator of Technological infrastructures or networks	PEO Continuity Internal Audit and Test PEO IT Infrastructure resilience PEO Disaster Recovery Responsible	G4 G6 G5	G4 contributes to G1 G6 contributes to G3 G5 contributes to G1
Poste operator	PEO Damage Impact Evaluator PEO processes definition responsible	G7 G8	G7 includes G3 G8 includes G3

V. SIMULATION-BASED EVALUATION

Once the model and relationships among actors and their goals are well described and defined, it is possible to use simulation to provide an assessment about what can happen into an application scenario according to specific inputs to the system. In the following, first a statistic based tool is exploited for a static analysis and then a more dynamic is adopted.

A. A statistics-centered approach

GeNIe (Graphical Network Interface) is a development environment for the creation of decision models [9]. It is presented as a graphical user interface of SMILE, a platform-independent library that implements functions for the execution and analysis of probabilistic / decision models, such as Bayesian networks, used to make probabilistic reasoning in decision-making situations under uncertainty.

Starting from different contractual terms of the services described above, it is possible to obtain an assessment in terms of the level of success (and complementary to the failure level) of the PEO service, which in turn can be associated with a level of risk. From the experience of the domain experts of Poste Italiane, the following percentage range is used:

- $Success > 90\%$ then *LowRisk*
- $89\% \geq Success > 70$ then *MediumRisk*;
- $Success \leq 70$ then *HighRisk*;

A first example is shown in Figure 5. By considering a combination of services based on the percentages shown in each block the probability of success is 99%, which means a *LowRisk*. The diagram is also enriched with two additional blocks: *FinancialGain* and *InvestmentDecision*, lead the decision maker to make decisions about the quality of the services to be subscribed. In this case, as shown by the “InvestmentDecision” and “Financial income” blocks, it is convenience to invest (with a gain of € 9850) by subscribing services with such quality parameters indicated, compared to not invest (€ 6940).

D. Application Modeling

The application model allows describing, with more details, a particular instance of the scenario under consideration. Specifically, Figure 4 represents the case of failure of a service to third parties necessary for the provision of online payment services, and the impact on the other users who use the service, possible costs (impact) for the failure to provide the service.

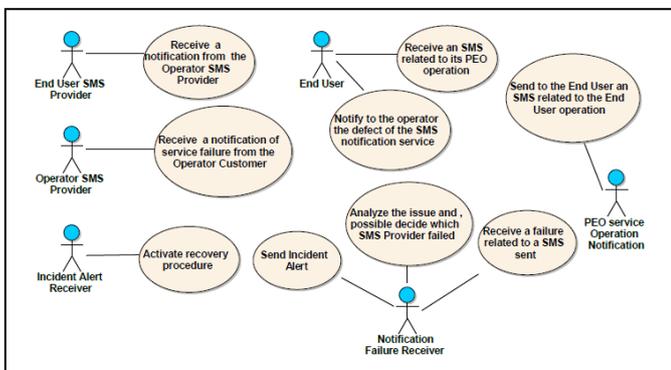


Fig. 4. Use Case diagram

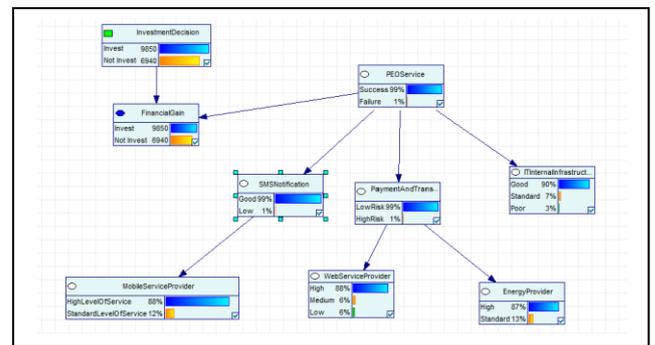


Fig. 5. Low Risk of the PEO service

Conversely, considering a low level quality of the *SMS Notification* service, and by also subscribing a low level quality of the *WebServiceProvider* service, the level of risk spreads systematically on the *Payments and Transactions* services by influencing drastically the PEO service. In fact, the success rate drops to 63%, which means “HighRisk” (Fig. 6).

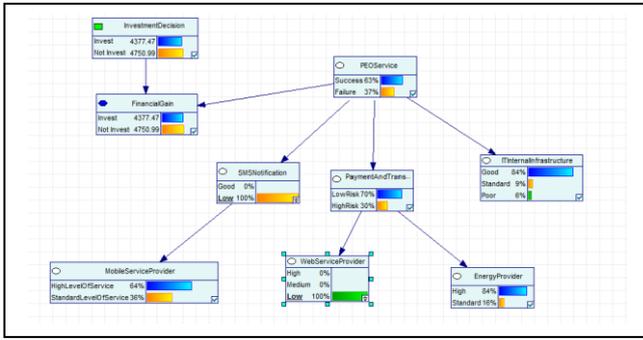


Fig. 6. High Risk of the PEO service

B. An Agent-based approach

This second approach is centered on a reference framework, called *ReActor*, an object oriented framework based on discrete-events simulation[3]. The reference model adopted for the definition and the development of the agent-based simulator for the analysis of the systemic risk is represented in Figure 7. In particular for each static blocks represented in Figure 6, a specific *ReActor* entity is defined. Then a behavior is associated to each of them, based on the follow four main actor models:

- *ServiceModel*: this model is employed for services belonging in the specific scenario to be analyzed; its aim is to provide the service associated to it;
- *AttackModel*: this model is adopted for modeling attack scenarios and related typologies of attacks respect to a specific *ServiceModel*;
- *RecoveryModel*: it aims to model policies and countermeasures in order to make more resilient a specific service when some anomalies occur;
- *ObserverModel*: it is employed for monitoring specific properties of interest which are strictly related to a specific service; it aims to collect information of specific properties, locally at service level or globally at scenario level.

Such models have been implemented by extending the above mentioned agent-based framework by mapping them as agents, that is, autonomous entities each of which has its own behavior. In particular, the *ServiceModel* is mapped as *ServiceAgent*; the *AttackModel* as an *AttackAgent*; the *RecoveryModel* is mapped as a *RecoveryAgent* and the *ObserverModel* as an *ObserverAgent*.

Such agents and their behaviors are achieved by implementing and extending the basic class *ActorBehavior* of the Reactor framework, which in turn, has been also defined as *Observable*. Consequently all agents that are introduced in the system, and that extends *ActorBehavior*, are potentially trackable. Whereas, the *ObserverModel* and as a consequence the *ObserverAgent*, has been marked as *Observer*, that is with the ability to monitor other agents. Finally, the behavior of each agent is characterized by different types of *Message*, that can respectively transmit, receive and handle in order to enable the communication with the other agents. As an example, the diagram in Figure 8 shows the behavior of the *ServiceAgent* defined as a state machine.

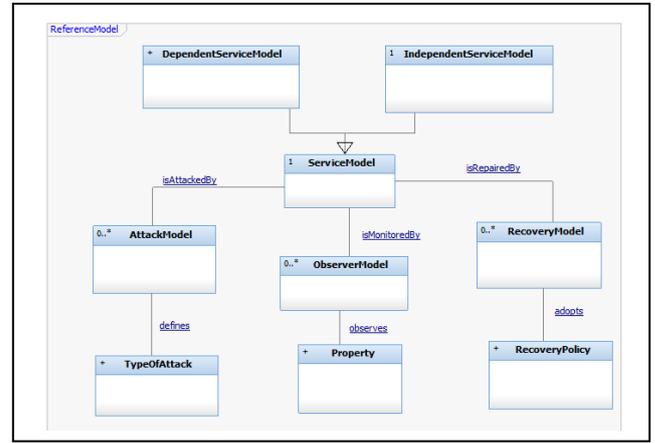


Fig. 7. Reference Model

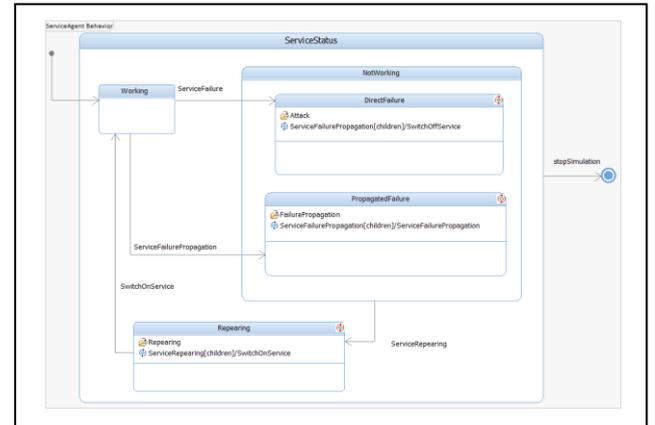


Fig. 8. ServiceAgent behavior

In particular, when the simulation starts, the status of *ServiceAgent* becomes *Working*. This means that the *ServiceAgent* is doing its job/delivering the service correctly. When an anomaly occurs, the state *Working* can get two types of events: *ServiceFailure* and *ServiceFailurePropagation*. Such events change the status of *ServiceAgent* into *NotWorking*, which, in turn, is defined in terms of two sub-states *DirectFailure* and *IndirectFailure*. In particular, when the *ServiceFailure* event occurs, the status *NotWorking* declines into the state of *DirectFailure*. This means that the failure of the service was due to internal factors of the service. This condition triggers the propagation of the failure by a *ServiceFailurePropagation* event to the services that depend from the *ServiceAgent*; this means that a service of the system, could receive a *ServiceFailurePropagation* event, which turns its status into *NotWorking* and specifically into the *IndirectFailure* status. This implies that its failure was due to a failure propagated by third parties on which it depends. Finally, from the *NotWorking* status, the *ServiceAgent* can receive a *ServiceRepeating* event that brings it into the *Repeating* status. This allows to recover/restore the *ServiceAgent* and propagate this information among the other services depending on it, so as to make them all *Working* again.

C. Discussion on the gathered results

From the analysis conducted on this case study, it is clear how the quality of services level and the involved system infrastructure (internal or third-party), strongly influence the success or the failure for the delivery of a service. In this case the use of a low quality Notification service is a critical. As a consequence, the choice of a good *MobileServiceProvider*, combined to a Medium/High quality of the *WebServiceProvider* is essential for making the system more resilient. Indeed, (i) in the first scenario, which involves the deployment of services with a high level of reliability, or in the second scenario, which combines medium-quality services, the system operates to keep resilient in presence of permanent failures, or temporary blackout, of some involved entities; (ii) instead, the second scenario highlights the high risk due to the strong dependence on entities that provide low robust / reliable services.

Whereas from the conducted study based software agents, other useful and more dynamic information are gathered from the simulation for each service involved (see Table X); for example: if a service is available (working) or unavailable (not working), the time when the failure of a service happened (timestamps), if the cause of the failure is due to external factors, the impact (e.g. in terms of money) per unit of time (e.g. per hours).

TABLE X. SIMULATION RESULTS RELATED TO THE PEO SERVICE

Service Name	Timestamp	Service status	External causes of failure	Impact (€) per Hour
WebService Provider	44	Not Working	no	3
Payment & Transaction	44	Not Working	yes	2
PEO	47	Not Working	yes	5
WebService Provider	56	Working	-	3
Payment & Transaction	58	Working	-	2
PEO	64	Working	-	5
...

VI. CONCLUSION

This paper presented a panorama on the concept of risk and, in particular, the systemic risk in the financial sector as well as in the cyber-security field. Furthermore, some recent research efforts about the modeling and assessment of systemic risk are also presented. In particular, an extended version of GOREM combined with the RAMSoS method has been employed.

A statistical analysis tool for the assessment of systemic risk based on a probabilistic approach, called GeNIe, has been adopted; whereas an actor-based and agent-oriented

framework for the development of a simulation platform for supporting the evolutionary assessment and dynamic behavior analysis of system has been exploited.

Finally, a first experimentation of such above mentioned conceptual and technical tools has been conducted on a case study concerning the assessment and the impact of failures on an online payment service.

ACKNOWLEDGMENT

This work has been partially supported by the “National Operational Programme for Research and Competitiveness” 2007-2013, Technological District on Cyber Security (PON03PE 00032 2 02), funded by the Italian Ministry of Education, University and Research, and the Italian Ministry of Economic Development.

REFERENCES

- [1] ACI Informatica – website <http://www.informatica.aci.it/>
- [2] M. Billio, M. Getmansky, A.W. Lo, and L. Pelizzon, “Econometric measures of connectedness and systemic risk in the finance and insurance sectors”, February 2012.
- [3] F. Cicirelli, A. Furfaro, L. Nigro, “A DEVS M&S framework based on Java and actors”, Proc. of 2nd European Modelling and Simulation Symposium, pp. 337-342, Barcelona (Spain), October 4-6, 2006.
- [4] S. Citrigno, A. Furfaro, T. Gallo, A. Garro, S. Graziano, and D. Saccà, “Mastering concept exploration in large industrial research projects,” Proceedings of the INCOSE Italian Conference on Systems Engineering (CIISE2014), Rome(Italy), November 24 – 25, 2014.
- [5] A. Furfaro, T. Gallo, and D. Saccà, “Modeling cyber systemic risk for the business continuity plan of a bank,” Proceedings of the International Cross Domain Conference and Workshop (CD-ARES’16), Salzburg (Austria), August 31-September 2, 2016.
- [6] A. Furfaro, T. Gallo, A. Garro, D. Saccà, and A. Tundis, “Requirements specification of a Cloud Service for Cyber Security Compliance Analysis”, Proceedings of the 2nd International Conference on Cloud Computing Technologies and Applications (CloudTech’16), IEEE, May 24-16, Marrakesh (Morocco), 2016.
- [7] A.Furfaro, T. Gallo, A. Garro, D. Saccà and A. Tundis, “ResDevOps: A Software Engineering Framework for Achieving Long-lasting Complex Systems”, Proceedings of the 24th IEEE International Requirements Engineering Conference, Beijing (China), September 12-16, 2016.
- [8] A. Garro, and A Tundis, “On the Reliability Analysis of Systems and SoS: the RAMSAS method and related extensions”, IEEE Systems Journal (IJS), vol. 9 (1), pp. 232-241, 2015.
- [9] GeNIe & SMILE – http://www.openclinical.org/dld_genieSmile.html.
- [10] National Security – <https://www.sicurezza nazionale.gov.it/sisr.nsf/lettere/prevenire-e-gestione-dei-rischi-globali.html>.
- [11] B.Phadermrod, R.M. Crowder, and G.B. Wills, “Developing SWOT Analysis from Customer Satisfaction Surveys”, Proc.of the 11th IEEE International Conference on e-Business Engineering (ICEBE), 2014.
- [12] Poste Italiane – website: <https://www.poste.it/>
- [13] Unified Modeling Language (UML) – <http://www.omg.org/spec/UML/>
- [14] World Economic Forum - Global risks 2014. Ninth Edition. 2014.
- [15] Zurich Insurance Company - Risk Nexus . Beyond data breaches: global interconnections of cyber risk. April 2014.
- [16] SysML V1.4 Specification Release <http://www.omg.sysml.org/>
- [17] Meyer B, “Agile! The Good, the Hype and the Ugly”, Springer International Publishing, 2014.