Techno-economic Analysis of SysML Models

Mara Nikolaidou and Christos Michalakelis
Department of Informatics and Telematics
Harokopio University of Athens
70 El. Venizelou St, Kallithea, 17671, Athens, GREECE
email: {mara,michalak}@hua.gr

***Abstract*—SysML is a prominent standard serving model-based system design for systems-of-systems (SoS). It facilitates the description of complex system design, as a hierarchy of interacting system components, while at the same time enables the independent description of each component behaviour. However, in order for the system designer to decide whether a SoS architecture is efficient, non-functional properties, such as performance or cost, should also be taken into consideration. There are numerous efforts integrating performance properties into SysML models and their verification using stochastic or analytical methods. In this paper, we focus on the financial assessment of SysML models, estimating economic attributes of SoS architectures, such as Total Cost of Ownership (TCO) and Return of Investment (ROI). Integrating different viewpoints into SysML SoS models, focusing besides structure and behaviour, also performance and economic attributes, enables the system designer to explore alternative solutions for system design, having in mind a wider perspective of the system under investigation. To this end, a profile extending SysML to serve techno-economic analysis of systems is introduced. As an example, the Total Cost of Ownership (TCO) for cloud infrastructures, modelled as SoS is explored. TCO properties are incorporated into existing SysML cloud models, enabling the designer to compute TCO for the whole system or parts of it.**

I. Introduction

System design, as an important phase of system engineering, regulates system architecture in order to satisfy specific requirements [1]. Model-based system design is supported by a number of methodologies [1], [2], [3] and is effectively accommodated by modelling languages and wide accepted standards as OMG. SysML [4], endorsed by OMG and INCOSE, facilitates the description of a broad range of systems and systems-of-systems in a hierarchical fashion. It enables the description of allocation policies and provides a discrete diagram for requirements specification. To describe specific system domains, SysML can be extended using the standard extension mechanism provided by MOF [5]. Stereotypes are a common mechanism to extend SysML functionally grouped in SysML profiles, an equivalent of UML profiles.

Most design decisions depend mainly on the conditions that a system should operate rather than the description of its structure. According to[1], a requirement denotes a capability or a condition that should be satisfied by the system under study and may be either *functional* (i.e., specifying a function that a system must perform) or *non-functional* (i.e., specifying a condition that a system must achieve). According to Wymore [1] there are six core categories of system design requirements, the prominent of which are performance and cost, that are non-functional requirements. Furthermore, system architectures should evaluated and properly adjusted until all design requirements, especially performance and cost, are verified in different levels of detail.

There are numerous efforts, where SysML requirement diagram and corresponding entities have been extended to effectively support the description of performance requirements [6]. Furthermore, their verification is performed using quantitative methods, such as simulation. There is a wide number of efforts to simulate SysML models using external simulators, such as the ones described in [7], [8], [9]. However, there are no similar efforts focusing cost requirements.

Techno-economics engineering combines process modeling and engineering design with economic evaluation, providing with both qualitatively and quantitative understanding of the impacts technology breakthroughs have on the financial viability of an ICT infrastructure [10]. In the context of SoS, such an analysis extends the corresponding components forming them by integrating into the corresponding SysML different types of constraints, beside technical also economic and social and technical, regarding the development and promotion of the SoS systems. [11]. COSysMO is a costing methodology, widely adopted with model-based system design, targeting SoS [12]. However, it is served by independent tools, not fully integrated within SysML models.

The vision of the proposed approach targets the development of a detailed model-based methodology based on SysML models, which will take into account all the appropriate economic properties, providing as output information for the assessment of a SoS investment[13], [14]. The economic view of SoS model shall facilitate the description of economic parameters such as initial capital expenditures (CapEx), operational expenditures (OpEx), costing schemes, amortization, forecasting details regarding the future demand of services and, even the market share of the provider to serve competitive market environments. The calculation of important economic indices, such as Net Present Value (NPV), Internal Rate of Return (IRR), Return On Investment (ROI), Economic Value Added (EVA) and Return On Assets (ROA), which have an important impact on the considered investment, should be incorporated within the SysML model.

Towards this direction, the work presented this paper can be seen as the first step in developing a model-based techno-economic methodology which should be able to provide reliable decision support to SoS designers. To this end, a SysML extension is proposed to incorporate cost-related properties in SysML models, so that techno-economic metrics like TCO may be computed and explored. This effort resulted in the construction of SysML Costing Profile, which is fully functional in Visual Paradigm modelling tool.

The rest of the paper is structured as follows. Related work is briefly presented in section II. The proposed SysML extensions and SysML Costing profile, emphazing TCO exploration, is presented in section III. In section IV, a case study where the proposed profile is applied in Cloud techno-economic engineering in conjunction with the CloudSim profile for cloud simulation. Conclusions and future work reside in section V.

II. Related Work

SysML facilitates the model-based design of systems or systems-of-systems (SoSs), providing different system views serving specific design activities [4]. Specifically, *Block definition diagrams* can be used to depict alternative system design views in multiple layers of detail (for example the software and hardware architecture of an information system). Requirements in SysML are described, as class stereotypes, in an abstract, qualitative manner, since they are specified by two properties, *id* and *text*, corresponding to a simple description without any qualitative or quantitative characteristic. Requirements can be grouped in packages based on common characteristics, as their category (for example functional or non-functional) or the activities they are related to (for example software or hardware requirements) forming a multi-level hierarchy.

There are numerous efforts to serve non-functional requirements (NFRs) by extending SysML. Defining NFR within SysML is used to compose components and verify their interoperability [15]. Interface automata method is employed to specify component interfaces and to verify interface compatibility for the domain of component based systems. In [16] SysML was employed for the verification of safety requirements which belong to NFR. To this end, a requirement stereotype was proposed to formalise requirements textual description in terms of logic predicates and/or numerical parameters.

There are numerous efforts to integrate performance properties within SysML models by extending SysML requirement concept([8], [17]). Many of them also provide for simulation code generation from SysML models [7] to verify such performance requirements. Briefly discussed, there are efforts using MARTE profile and related tools [18], TTool Toolkit, which uses AVATAR profile, SysML-to-Arena transformation tools [19], and SysML4Modelica Project [8]. One of the authors has also proposed a framework to simulate information system models expressed in terms of SysML using the DEVS simulation framework [9]. To this end, several tools and technologies are utilised, like QVT for the transformation of the system model to simulation model, while the verification process is triggered and finalised via the system modelling environment.

Following a similar approach, one could extend SysML utilising the requirement entity in order to embed cost-related properties into SysML models in order to enables the techno-economic analysis of SoS architectures.

Syndeia (formerly known as SLIM), [20] is a commercial collaborative model-based systems engineering workspace that uses SysML as the front-end for orchestrating system engineering activities from the early stages of system development. The SysML-based system model serves as a unified, conceptual abstraction of the system independent of the specific design and analysis tools that shall be used in the development process. It is designed to provide plugins to integrate the SysML system model to a variety of design and analysis tools. Until now, only the integration of SysML and other model repositories, such as product lifecycle management (PLM) tools is implemented. Integration with MATLAB/Simulink ,Mathematica and OpenModelica is offered in a variety of commercial tools, but these tools are used as math solvers and not as a verification method of a complete SyML model in a specific domain. We share the vision of Syndeia, while the proposed approach also integrates techno-economic analysis of SoSs.

Despite the fact that there are several proposed pricing schemes for the SoS, models emphasising techno-economical parameters has been mostly neglected. The most prominent effort in this area is CoSysMO, a methodology targeting to estimate the cost of a SoS architecture [12]. However, it is not integrated within SysML models. Also, Strebel and Stage [21] applied an economic decision model for business software application deployment in large-scale environments. In [13] a method and a software tool for cost calculation and analysis for large-scale and cloud was developed. This is the reason why integrating cost analysis in the SoS engineering, utilising SysML models is considered to be an innovative research area of interest, motivating the current study.

III. Extending SysML to serve techno-ecomonic analysis of SoS architectures

Enable techno-economic engineering upon existing SoS models, either general or based on domain-specific profiles, is imperative in SoS efficient design. Enable the integration of economic engineering with performance engineering, an important aspect when the system designer investigates question related to the fact: Having a specific budget, would the system architecture be sufficient in terms of performance? Is there a balance between performance and costing requirements or restrictions? What about Return-on-Investment (ROI) parameters? Could one predict the cost of system operation as system performance during system design? These questions become more important as systems and especially systems-of-systems constantly become more complex.

The estimation of the Total Cost of Ownership (TCO) in particular is a procedure that provides the means for determining the total economic value of an investment, including the initial capital expenditures (CapEx) and the operational expenditures (OpEx). In the context of SoS engineering, TCO corresponds to the estimation of the costs required to integrate and operate a SoS infrastructure. To enable the estimation and exploration of the TCO of alternative SoS architectures, corresponding SysML models should be properly extended to describe cost-related properties. The System-of-Systems (SoS) approach does not advocate particular tools, methods, or practices; instead, it promotes a new way of thinking for solving grand challenges of systems whose the interactions of technology, policy, and economics are the primary drivers [22]. Such model-based system approaches may also be applied in a techno-economic context, when designing or configuring a complex system architecture. The task of exploring economic properties is really interesting, since the desired performance and availability are constrained by and combined with ownership and operating costs and investment parameters [11], [23].

How could such properties be integrated in SysML system models? Cost-related attributed can be treated as non-functional requirements [1]. Thus, they may be represented using the $Requirement$ SysML entity and related with each other using standard SysML requirement relations, such as $satisfy$, $verify$, $derive$, etc [4]. However, as explained in the numerous efforts to extend SysML requirement concepts presented in Related Work, the $Requirement$ entity should be enriched with numerical properties, to adequately describe non-functional quantitative requirements, such as cost-related ones [24]. Furthermore, cost requirement computation should be facilitated, by enabling the system designer to define computation functions of capital expenditures (CapEx) and operational expenditures (OpEx), described as cost requirements associated to the corresponding SoS component they concern. SysML Requirement concepts extensions should be general enough to be applicable in any type of SoS.The corresponding meta-model is depicted in Fig. 1. Standard SysML entities are depicted with clear rectangles, while proposed extensions are depicted using light blue.

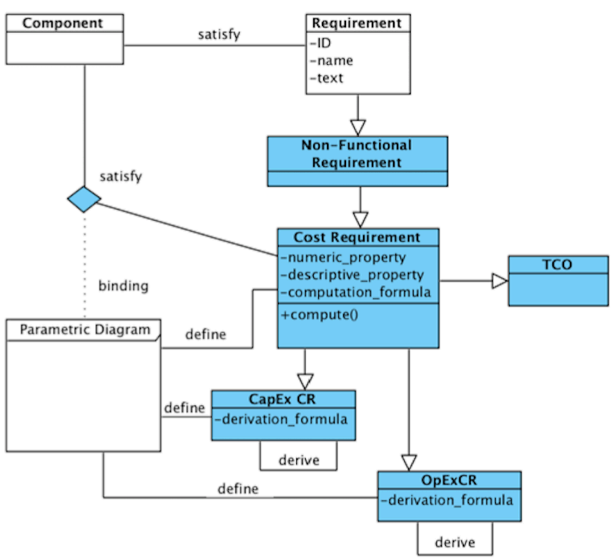

Fig. 1. SysML extensions to support Cost Requirements

Cost-related requirements are defined as a specialisation of Non-Functional requirement entity. In practice, they are defined as stereotypes of the $Requirement$ entity. Each $CostRequirement$ corresponds to a discrete cost category (for example acquisition, operation) is described by cost-related descriptive, but most importantly numeric properties. These properties are explicitly related to Cost categories and are independent of the description of SoS components, which must $satisfy$ the Cost requirements. Each cost requirement is also described by a $ComputationFormula$. This describes the equation or formula used to estimate the specific cost category based on $Cost$ properties. Properties of SoS components related to the Cost requirement may also be included in the equation. The $ComputationFormula$ is defined by the system designer using a SysML Parametric Diagram [4] in a graphical fashion, where both Component and Cost Requirement properties may participate. This enables the designer to describe cost computation functions in a standardised fashion using SysML.

Cost Requirement may be specialised in $CapEx$, related to acquisition costs, and $OpEX$, related to operation costs, requirements. CapEx and OpEx requirement categories are independent from each-other, but they may both be satisfied by the same SoS component. Since SoS components are integrated in a hierarchy, a CapEx requirement satisfied by a specific component may be derived by corresponding CapEx requirements of its constituting components. The corresponding $DerivationFormula$ of a derived requirement is defined using a SysML parametric diagram, the same way $ComputationFormulas$ are defined.

The Total Cost of Ownership (TCO) depicted as TCO requirement, satisfied by a SoS component, is estimated by combining CapEx and OpEx requirements of all its constituting components. It is also described by a $ComputationFormula$.

Proposed extensions are implemented in the $SysMLCosting$ profile. SysML requirement entity extensions are defined as stereotypes, while corresponding restriction are implemented as OCL constraints. The profile has been implemented in Visual Paradigm modelling tool. The $Compute$ method of all cost requirement, responsible for the automated execution of Computation formulas graphically define is SysML parametric diagrams, is implemented as a Java plug-in.

IV. An integrated SysML-based approach for the techno-ecomonic analysis of Cloud infrastructures

Cloud infrastructures are often explored using SysML [25], [26], [27]. Whether IaaS, PaaS or SaaS services are provided, cloud systems are indeed SoS. To this end, there are numerous efforts towards cloud engineering using SysML as the modelling language. Depended on the cloud aspect investigated, different SysML diagrams are utilised. In [28], SysML component diagrams are utilised to model and study multi-cloud systems (e.g. applications running on multiple cloud platforms). A domain-specific modelling language, called CloudML, is introduced to depict restrictions and specification of multi-cloud architectures, e.g. to manage the interaction restrictions between SysML components modelling corresponding cloud entities. In [29], where cloud software is modelled as a SysML component hierarchy, Architecture Description Language (ADL) is used to describe the interaction between cloud software components. Focusing on IaaS architecture, e.g. cloud infrastructures, in [26] a SysML profile focusing availability issues is introduced to explore cloud specification models expressed in SysML. A specific modelling framework, called Candy, was developed to serve this effort. However, there is a question raised for all these efforts. Are Cloud-related SysML profiles and corresponding tools easy to use or are they restricted by specific environments? In most cases, the suggested approaches are restricted to a specific modelling tool and can not be easily integrated, since the suggested profiles, may not be simultaneously applied.

CloudSim [30] is a simulation environment, enabling the simulation of SysML cloud infrastructure models in a automated fashion, provided that clouds are modelled in SysML utilising the corresponding CloudSim profile. The profile is formally defined and can be integrated in a standard UML modelling tool supporting SysML. SysML models can be consequently exported and simulated in a Java simulation environment. Using CloudSim SysML profile a system designer can describe the basic system components of the cloud, such as data centers, hosts, virtual machines and network topologies, along with their functionality, as a SoS hierarchy. There are several researchers and organisations using CloudSim to study cloud architectures and energy-efficient management of cloud data centers. CloudSim also provides for the estimation of operation cost or profit for each CloudSim entity during simulation.

In [27], we presented an attempt to extend CloudSim SysML profile to also include cost properties. This effort is restricted in the sense that the computation of cost parameters, such as TCO, is restricted in cloud specific architectures as defined by CloudSim meta-model. In the following, we shall demonstrate how the proposed SysML Costing Profile may be concurrently applied with CloudSim profile leading to the same result, e.g the computation of TCO for CloudSim infrastructures. Both profiles are implemented within the Visual Paradigm tool (https://www.visual-paradigm.com).

An example of a CloudSim SysML representation of a simple DataCenter consisting of 4 hosts is depicted in Fig. 2. The Datacenter entity depicts the ICT infrastructure that cloud providers offer. It encloses a set of 4 hosts representing a physical resource (computing or storage server) in a cloud. The allocation of memory, bandwidth and processing power to VMs is also represented. The VM component models a VM instance that runs inside a host, sharing the host with other VMs. There are 7 of them depicted in the figure. Each one of them is characterised by accessible memory, storage, processing power and configurations (software environment), which are allocated to it by the host or hosts serving it. CloudSim entities are represented as a hierarchy of SysML block entities (grey rectangles). Simulation-related properties are also assigned to them, to make the simulation of the infrastructure possible.

SysML Cost profile entities (e.g. cost-related requirements), depicted as clear rectangles in aFig. 2, are associated to CloudSim entities. More specifically, both CapEx and OpEx cost requirements are associated to DataCenter components, using corresponding SysML *satisfy* relation. According to [13], cloud TCO includes costs that are related with physical servers, software, facilities (wires, racks), labor, networking, power and cooling energy and real estate. As explained in [27], each one of these costs can be associated to a CloudSim entity, e.g. DataCenter, Host or VM. Adopting the concepts as in [27], the following CapEx cost requirements are defined:

- *ServerCost* describes the cost of physical servers of a cloud and this is why it is associated to Hosts. It is assumed that all servers are of the same type, CPU, memory disk and they share the same configuration.
- *SoftwareCost* represents the cost for the software licenses that run in the cloud. It is associated to VMs.
- *NetworkingCost* describes the cost related to the networking, meaning to switches, cables, Network Interface Cards (NICs) and is associated with the entity of NetworkTopology. It is associated to the DataCenter.
- FacilitiesCost class refers to the necessary supplies for the operation of ICT equipment, like cables and PDU. It is associated to the DataCenter.

Also the following OpeEx cost requirements are defined:

- *Power Cost* models the cost of the total power consumption. It is associated to Hosts.
- *CoolingCost* models the cost related with the amount of power consumed by cooling equipment. It is associated to the DataCenter.
- *SpaceCost* describes the real estate cost of a building that fulfils the specifications, in order to accommodate a cloud datacenter. It is associated to the DataCenter.
- *LaborCost* represents the cost of wages paid to employees of the data center. It is associated to the DataCenter.

Each cost requirement, either CapEx or OpEx, is described by its own attributes. Such properties may be either simple, defined by the designer, or can be derived by properties of the same or related entities. As such an example, the Power Cost OpEx requirement is examined.

Power consumption maintain a vital role in the management of the datacenter[31]. In CloudSim profile, the energy consumption of each host can be presented by HostPowerCost attribute of Host component. However, as it is a simple attribute, it should be entered by system designers, after calculating it themselves. The SysML Costing Profile entities may additionally facilitate him/her, to automatically compute HostPowerCost property, as explained in the following.

As indicated in [27], the attributes that are used to define energy consumption of hosts within a DataCenter are [13]:

- Es: The price per hour of 1kw of electricity
- Ls: Steady -state constant
- $Nrack$: The number of racks in working
- $Time$: Hours consumed
- Ap: Amortization period unit (year/month/hour)
- Arp: presents the cost amortizable rate parameter . It is derived from the operation *calculateArp()* that it is described by the relation:
  $Arp = (1 + 0.05) * time/(30 * 24 * Ap)$

$$HostPowerCost = Ls * Es * NRack * Arp(time) \quad (1)$$

How should this information be integrated within SysML Cloud model? The PowerCost OpEx requirement associated to Host component can be used to describe cost-related numeric parameters, except of $Nrack$ already defined as Host property. PowerCost value is also defined as a PowerCost OpEx requirement property, characterised as a derived one, thus a The relation defining HostPowerCost should be defined as the $DerivationFormula$ of $PowerCost$ OpexEx requirement.

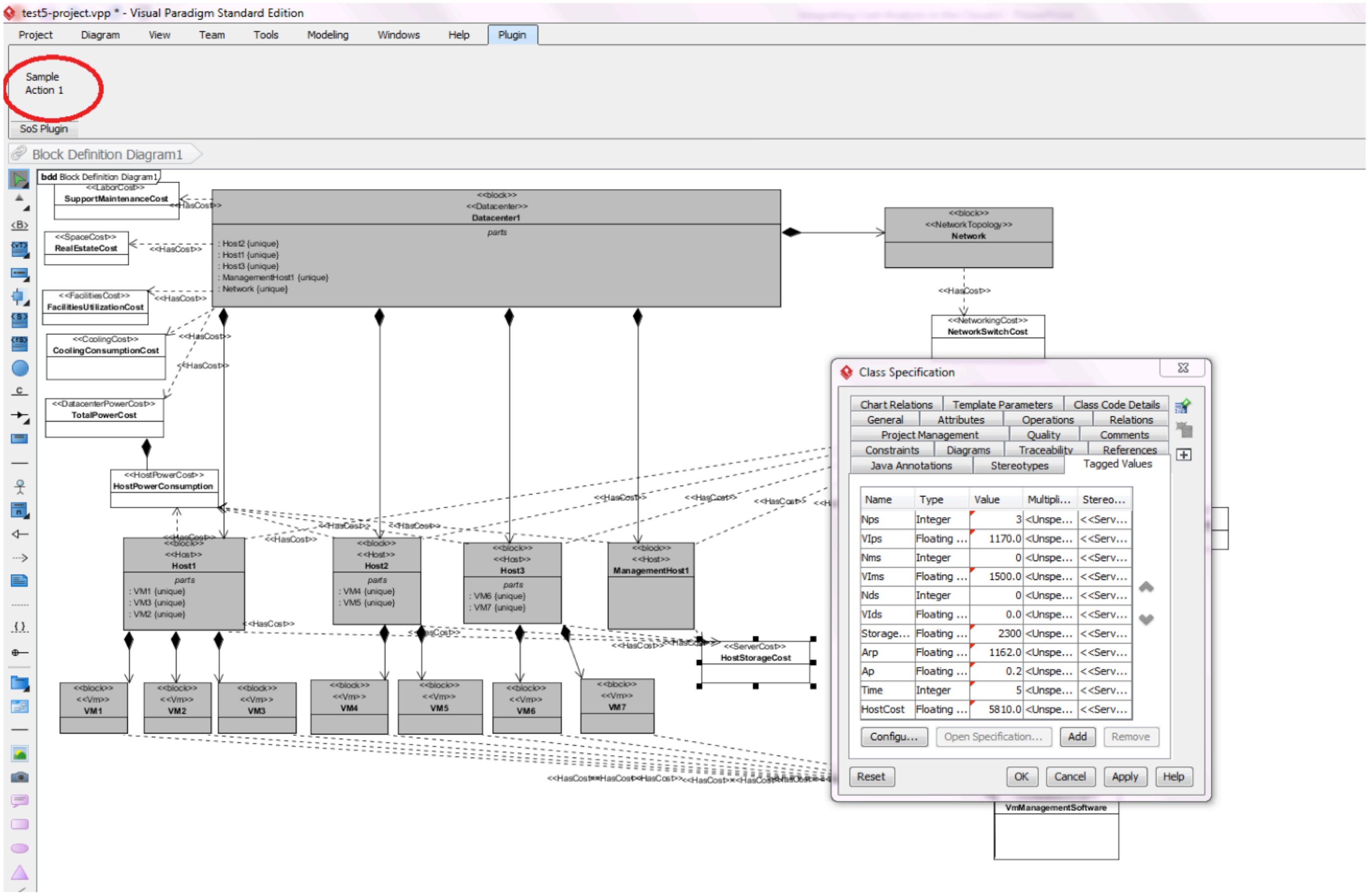

Fig. 2. SysML CloudSim representation of a Datacenter, enriched with SysML Costing Profile properties

It is defined using a SysML parametric diagram depicted in Fig. 3. Attributes of both $PowerCost$ OpexEx requirement and $Host$ block component participate in the formula. The system designer may graphically define the formula, which corresponds to PowerCost estimation.

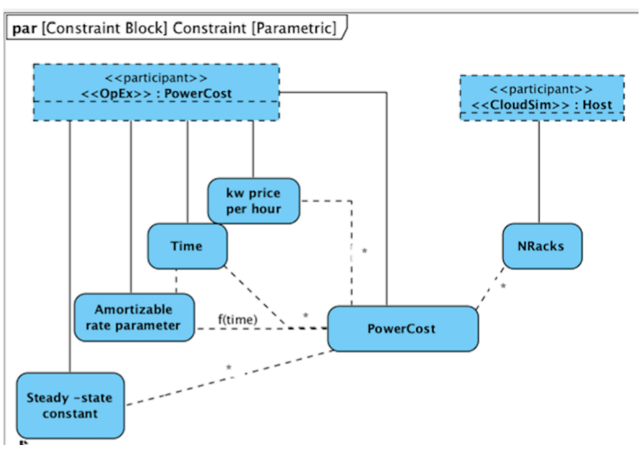

Fig. 3. A Parametric Diagram describing PowerCost Computation Formula

As indicated in the figure, $PowerCost$ attribute is a derived one (e.g. it is automatically computed using the computation formula defined in the corresponding parametric diagram), this functionality should be embedded within the corresponding modelling tool, in this case Visual Paradigm. This is accomplished by pushing the corresponding button (on the top left of Fig. 2), when selecting the corresponding requirement. In the bottom right part of the figure, one may see the attributes of the $PowerCost$ OpexEx requirement, after the execution of the computation formula.

One major difference from the approach presented by the authors in [27], lays in the fact that the code corresponding to the $compute()$ method is the same for all cost requirements and corresponds to the execution of the corresponding formula described in the parametric diagram. In [27], the computation formula is programmed for each cost property separately, thus no standardisation of the process is promoted. When applying the SysML Cost profile, the system designer is responsible for defining cost-related computation formulas using standard SysML notation in Parametric diagrams, while the profile provides the means for the automated execution. This contributes to the generality of the proposed approach. When integrating SysML Costing profile with CloudSim profile, the value of PowerCost attribute is also copied to HostPowerCost attribute of the Host component to ease the designer's work.

TCO requirement associated the DataCenter may by computed in the same fashion by adding all CapEx and OpEx requirements associated with all the entities belonging in its hierarchy. This feature is also independent of the structure of the SoS entity, in this case the DataCenter, and the way specific

CapEx and OpEx requirement attributes are computed. Thus it may be applied in any SoS hierarchy modelled in SysML.

V. Conclusions - Future Work

A SysML extension to integrate techno-economic analysis into SysML models is presented in the paper. The approach is based on extending SysML requirement entity and related concepts, and is general enough to be applied in any SoS system. Different cost categories are represented as requirements associated to SoS components that must satisfy them. As far as the case study is concerned, the combined application of both cost-related and performance-related profiles was very promising.

Taking into account the investments needed to operate a SoS architecture, the proposed approach and the corresponding SysML meta-model extension enable system designers to have a more integrated view of the system they are studying, combing economic, performance and structural data in order to evaluate a proposed solution. We consider this effort as a first step to develop a model-based techno-economic methodology for SoS engineering. Future research targets the identification and calculation of cost factors and indices that should be integrated to complete the techno-economic analysis and evaluation of SoS ecosystems.

References

[1] INCOSE, *Systems Engineering Handbook*, version 3.2.2 ed. San Diego, CA, USA: International Council on Systems Engineering, 2012.

[2] B. Nolan, B. Brown, L. Balmelli, T. Bohn, and U. Wahli, *Model Driven Systems Development with Rational Products*, IBM Red Book, 2008.

[3] M. Nikolaidou and D. Anagnostopoulos, "A systematic approach for configuring web-based information systems," *Distributed and Parallel Databases*, vol. 17, no. 3, pp. 267–290, 2005.

[4] OMG, *Systems Modeling Language (SYSML) Specification, Version 1.3*, Std., June 2012. [Online]. Available: http://www.omg.org/spec/SysML/1.3/PDF

[5] OMG, "Meta object facility (MOF) 2.0 Query/View/Transformation specification," *Transformation*, no. April, pp. 1–230, 2008. [Online]. Available: http://www.omg.org/spec/QVT/1.0/PDF/

[6] A. Tsadimas, M. Nikolaidou, and D. Anagnostopoulos, "Handling non-functional requirements in information system architecture design," in *Software Engineering Advances, 2009. ICSEA'09. Fourth International Conference on*. IEEE, 2009, pp. 59–64.

[7] M. Nikolaidou, G.-D. Kapos, A. Tsadimas, V. Dalakas, and D. Anagnostopoulos, "Simulating sysml models: Overview and challenges," in *System of Systems Engineering Conference (SoSE), 2015 10th*. IEEE, 2015, pp. 328–333.

[8] OMG, *SysML and Modelica Integration*, Dec. 2008. [Online]. Available: http://www.omgwiki.org/OMGSysML/doku.php?id=sysml-modelica:sysml_and_modelica_integration

[9] G.-D. Kapos, V. Dalakas, M. Nikolaidou, and D. Anagnostopoulos, "An integrated framework for automated simulation of sysml models using devs," *Simulation*, vol. 90, no. 6, pp. 717–744, 2014.

[10] R. S. Z.Fang Jr., X. Qi, *Production of Biofuels and Chemicals with Ultrasound*. Springer, 2014.

[11] S. Verbrugge, "Technoeconomics," 2009. [Online]. Available: https://www.ibcn.intec.ugent.be/content/techno-economics

[12] M. Dabkowski, J. Estrada, B. Reidy, and R. Valerdi, "Network science enabled cost estimation in support of mbse," *Procedia Computer Science*, vol. 16, pp. 89 – 97, 2013. [Online]. Available: http://www.sciencedirect.com/science/article/pii/S1877050913000112

[13] Y. L. X.Li T. Liu, J. Qiu, F. Wang, "The Method and Tool of Cost Analysis for Cloud Computing," 2009.

[14] A. Khajeh-Hosseini, I. Sommerville, and I. Sriram, "Research challenges for enterprise cloud computing," *arXiv preprint arXiv:1001.3257*, 2010.

[15] S. Chouali, A. Hammad, and H. Mountassir, "Assembling components using sysml with non-functional requirements," *Electronic Notes in Theoretical Computer Science*, vol. 295, no. 0, pp. 31 – 47, 2013, proceedings the 9th International Workshop on Formal Engineering approaches to Software Components and Architectures (FESCA). [Online]. Available: http://www.sciencedirect.com/science/article/pii/S157106611300025X

[16] J.-F. Pétin, D. Evrot, G. Morel, and P. Lamy, "Combining SysML and formal methods for safety requirements verification," in *22nd International Conference on Software & Systems Engineering and their Applications*, Paris, France, Dec. 2010. [Online]. Available: http://hal.archives-ouvertes.fr/hal-00533311

[17] A. Garro and A. Tundis, "Modeling of system properties: Research challenges and promising solutions," in *2015 IEEE International Symposium on Systems Engineering (ISSE)*, Sept 2015, pp. 324–331.

[18] M. R. S. Marques, E. Siegert, and L. Brisolara, "Integrating uml, marte and sysml to improve requirements specification and traceability in the embedded domain," in *2014 12th IEEE International Conference on Industrial Informatics (INDIN)*, July 2014, pp. 176–181.

[19] O. Batarseh and L. F. McGinnis, "System modeling in SysML and system analysis in ARENA," in *Proceedings of the Winter Simulation Conference*, ser. WSC '12. Winter Simulation Conference, 2012, pp. 258:1–258:12. [Online]. Available: http://dl.acm.org/citation.cfm?id=2429759.2430107

[20] M. Bajaj, D. Zwemer, R. Peak, A. Phung, A. Scott, and M. Wilson, "Slim: collaborative model-based systems engineering workspace for next-generation complex systems," in *Aerospace Conference, 2011 IEEE*, 2011, pp. 1–15.

[21] J. Strebel and A. Stage, "An economic decision model for business software application deployment on hybrid Cloud environments," *Multikonferenz Wirtschaftsinformatik 2010*, p. 47, 2010.

[22] "System of systems," Wikipedia.org. [Online]. Available: http://en.wikipedia.org/wiki/System_of_systems

[23] B. Martens, M. Walterbusch, and F. Teuteberg, "Costing of cloud computing services: A total cost of ownership approach," in *System Science (HICSS), 2012 45th Hawaii International Conference on*. IEEE, 2012, pp. 1563–1572.

[24] A. Tsadimas, M. Nikolaidou, and D. Anagnostopoulos, "Extending sysml to explore non-functional requirements: the case of information system design," in *Proceedings of the 27th Annual ACM Symposium on Applied Computing*. ACM, 2012, pp. 1057–1062.

[25] S. Hosono, H. Huang, T. Hara, Y. Shimomura, and T. Arai, "A lifetime supporting framework for cloud applications," in *2010 IEEE 3rd International Conference on Cloud Computing*, July 2010, pp. 362–369.

[26] F. Machida, E. Andrade, D. S. Kim, and K. S. Trivedi, "Candy: Component-based availability modeling framework for cloud service management using sysml," in *Reliable Distributed Systems (SRDS), 2011 30th IEEE Symposium on*. IEEE, 2011, pp. 209–218.

[27] E. Filiopoulou, P. Mitropoulou, A. Tsadimas, C. Michalakelis, M. Nikolaidou, and D. Anagnostopoulos, "Integrating cost analysis in the cloud: A sos approach," in *Innovations in Information Technology (IIT), 2015 11th International Conference on*. IEEE, 2015, pp. 278–283.

[28] N. Ferry, A. Rossini, F. Chauvel, B. Morin, and A. Solberg, "Towards model-driven provisioning, deployment, monitoring, and adaptation of multi-cloud systems," in *Proceedings of the IEEE Sixth International Conference on Cloud Computing, CLOUD*, vol. 13, 2013, pp. 887–894.

[29] A. N. Perez and B. Rumpe, "Modeling Cloud Architectures as Interactive Systems," *arXiv preprint arXiv:1408.5705*, 2014.

[30] R. N. Calheiros, R. Ranjan, A. Beloglazov, C. A. F. De Rose, and R. Buyya, "CloudSim: a toolkit for modeling and simulation of cloud computing environments and evaluation of resource provisioning algorithms," *Software: Practice and Experience*, vol. 41, no. 1, pp. 23–50, 2011.

[31] A. B. Rajkumar Buyya Jemal Abawajy, "Energy-Efficient Management of Data Center Resources for Cloud Computing: A Vision, Architectural Elements, and Open Challenges," Las Vegas, 2010.