Model of Early Specifications of Performance Requirements at Functional Levels

Khalid T. Al-Sarayreh

***Abstract*— This paper presents an integrated standards-based model that helps in early identification, specification and measurement for a single type of NFR, which is the performance requirement. The development stages of the standards-based framework have passed by two main steps: the first step is constituted in identifying and analyzing the system performance requirements and their allocated software performance requirements that are dispersed into the IEEE and ECSS international standards, the second step is modeling the identified system/software performance requirements using the Soft-goal Interdependency Graphs and clarifying the interdependency relations between these requirements.**

***Keywords*—** Performance Requirements, International Standards, Soft-goal Interdependency Graphs.

I. INTRODUCTION

The proper identification, specification and measurement of the system requirements at early development phases constitute the most significant factor to build a successful system that satisfies the stakeholder expectations and needs. In software engineering, the requirements are classified under two types: the functional requirements (FR) which are defined as the functionality that is required to be provided by the system (for instance: "The system shall be able to transfer data via internet"), and the non-functional requirements (NFR) are defined as the restrictions that should be applied on the required functions (for instance: "The system shall be able to transfer data via internet with low response time").

In the academic field, several researchers have referred in their reports to the difficulties and challenges that the developers are faced to handle with NFR, for instance: taking NFR as a quantitative input to be measured and involved in the project budget estimation alongside with the FR [1-2]. Several approaches and methods are proposed from different researcher's perspectives to facilitate dealing with these challenges; nevertheless, there is currently a lack of generic models for early addressing and measuring these requirements at the system level and their related functional requirement at the software/ hardware level [3-4].

In parallel with the academic field, international standards organizations (such as the ECSS and the IEEE) are interested in describing and categorizing the NFR types. Since the European Cooperation for Space Standardization (ECSS) and the Institute of Electrical and Electronics Engineer (IEEE) categorized the performance requirements as a single type of NFR and discussed them by various terminologies and views.

This paper will account a new model for early specifications of performance requirements at functional levels based on the finding of international standards in parallel with academic previous work of some of the respected models regarding non functional performance requirements as an self-sufficient model to identify the size of the software performance separately of the languages types, whereas keep away from the limitations viewed in the performance measures presently offered.

The paper scope is to classify independently the all functionality allocated to software performance as a part of set pieces of the system application in the requirements phase for any software applications, whether the application has been built or it has already to be delivered.

In addition, the main contribution of this paper is the proposed model of software performance requirements. The proposed nonspecific model is considered as type of a orientation model in the common sense of an etalon standard that is being used for the measurement of software performance.

This paper is organized as follows. Section 2 presents the related works. Section 3 presents Performance REQUIREMENTS as defined in International Standards. Section 4 presents The Foundations of the proposed model of performance Requirements. A conclusion is presented in section 5.

II. RELATED WORK

Many early efforts have been concerned with defining, specifying and modeling NFR. For instance: [5] this paper proposed a performance requirements model; it joins together a multiplicity of types of knowledge of information systems and performance. The proposed framework includes the following performance conceptions, software performance

This work was supported in part by the U.S. Department of Commerce under Grant BS123456 (sponsor and financial support acknowledgment goes here). Paper titles should be written in uppercase and lowercase letters, not all uppercase. Avoid writing long formulas with subscripts in the title; short formulas that identify the elements are fine (e.g., "Nd–Fe–B"). Do not write "(Invited)" in the title. Full names of authors are preferred in the author field, but are not required. Put a space between authors' initials.

F. A. Khalid T. Al-Sarayreh is now with the Hashemite University, Prince Hussein Bin Abdullah II for Information Technology, Department of Software Engineering, 13115 Zarqa, Jordan. (e-mail: khalidt@hu.edu.jo).

principles for construction the performance into systems, and development knowledge. The performance and development process build by using goal-oriented approach, the performance NFR framework, which suggests a developer-directed graphical handling for stating NFRs, analyzing and connecting them, and identifying the impact of judgments leading NFRs. This move toward to built a customized solution of the domain.

For instance, [6] proposed a model for performance requirements specifications and consequential a validation testing. The model can be incorporated into agile development approaches. The performance requirements can be specified incrementally, without analysis.

More instance, [7] present a new algorithm for passive testing approach in specifying the performance time communicated protocol properties to test the real execution traces and evaluated the proposed algorithm using experimental testing on the basis of the software performance requirements through a set of properties for real execution traces.

Moreover, [8] proposed an approach to elicit performance requirement from customers for software banking system using ontology. This model divided the performance requirements into three parts: system, subsystem and component levels; between these parts ontology's inference function is used to validate and complete the requirements.

Furthermore, [9] focused on the development of performance requirements for ionospheric effects in low-frequency SAR data set applications. The performance requirements were derived considering the data quality needs of a set of SAR applications. The proposed requirements can serve as a benchmark for a performance assessment of ionospheric correction methods to define the system suitability for the system.

In addition [10] proposed a process of safety requirements with random failure of a supply system to describe geo location petri-net for model of verification and performance analysis with the widely increasing number of location-based services. Typical consumer geolocation [11] technologies are analyzed based on performance aspect for use with location-based services. While [12] proposed a performance model of requirements with related text updates to correct inconsistencies and remove limitations introduced by IEEE Std C37.118.1(TM)-2011.

In [13] present a model of distribution transformers to realize a smarter grid. The analysis has been carried out on the performance requirements and evaluation of distribution transformers when they are integrated to grid level and [14] proposed a wide Area Measurement Systems to observe the static and dynamic performance of power systems.

Finally, [15] described a method to define the performance requirements for Airport Surface Surveillance. The key idea is making the performance specification dependent on the underlying sensor deployment and geometric definition of the scenario, which enables its extension to any operational deployment.

III. Performance Requirements As Defined in International Standards

This section presents and discusses the performance terms and views for identifying the system performance NFR and their related software performance FR that may be used for specifying and measuring the system performance requirements.

A. ECSS concepts and views for performance requirements

ECSS standards [16], [17], [18] and [19] have discussed the system/software performance FR in the context of early system development phases whereas the system performance NFR have been discussed in much later phases. In the domain of these standards, the performance requirements have been defined as the specification that the output of the system does not deviate by more than a given amount from the target output.

Moreover, [20] offered a general knowledge of the control systems engineering and its applications to space missions, such as satellite system, spacecraft system, a launcher rocket system or any other technical system involving control. Such standards are emphasized on the necessity to conduct the performance analysis during all the control system development phases to evaluate that the control system is consisted and cohesive with:

- The control objectives: which are generated by the requirement engineering process?
- The numerical requirements: which are identified by the requirement analysis?

Monitoring or evaluating the performance of the system is often assisted by improving the use of the software in the system. The performance monitor is considered a facility which is integrated into a specific processer to monitor the selected characteristics to assist in debugging and analyzing systems by determining a machine's state at a particular point in time. Often, the performance monitor [20] produces information which are related to the usage of the processor's instruction execution and storage control: for instance, the performance monitor can be used to produce information related to the period of time that has passed between events in a processing system. The information produced usually guides system architects toward ways of improving performance of a given system or developing improvements in the design of a new system.

The following terms are mentioned by ECSS standards to describe the performance requirements:

- Response to reference signals (e.g. response time, settling time, and tracking errors for command profiles).
- Accuracy and stability errors in the presence of disturbances.
- Frequency domain requirements (e.g. bandwidth).
- Measurement errors (e.g. attitude knowledge)
- Processing speed.
- Resource consumption.
- Throughput.

B. IEEE concepts and views for performance requirements

IEEE organization defines the performance requirements as a static and dynamic numerical requirement that is located on the software or on the human interaction with the entire software [21]. These two types of performance requirements should be stated in measurable form.

The following terms are mentioned by IEEE standards to describe the performance requirements:

- Static numerical requirements (e.g. capacity and concurrency).
- Dynamic numerical requirements (e.g. workload).

IV. THE FOUNDATIONS OF THE PROPOSED MODEL

The proposed framework has been developed based on using the following two foundations:

- The soft-goal interdependency graphs: which is used to model and describe the interdependency relations between the system performance NFR and their allocated software performance FR.
- The roles of ISO 19761[22] method: which are used to measure the data movement size for the allocated software performance FR.

The next sub-sections are described the referred foundation in more details way.

A. The view of Softgoal Interdependency Graphs (SIGs)

The Soft-goal Interdependency Graphs (SIGs) is introduced by [23] for describing and modeling the non-functional requirements and the interdependencies relation between them. SIGs represents the NFR as soft-goals, each soft-goal (parent soft-goal) is decomposed into one or more specific soft-goals (child soft-goals until reaching one or more solutions that satisfy the NFR (parent soft-goal).
SIGs introduce three types of soft-goal, which are:

- Soft-goal: displays the NFR to be satisfied by the system.
- Operationalizing soft-goals: represents possible solutions (operations, processes, data representations) or design alternatives that assist to satisfy the NFR.
- Claim soft-goals: shows the refinement between soft-goals or the rationale related to a soft-goal [23].

The child-soft-goals provide two contribution types to satisfy the parent soft-goals: positive contributions and negative contributions. Table 1 shows that the positive contributions are divided into make, help and some+ contributions while the negative contributions are divided into break, hurt and some- contributions (see Table 1 and Figure 1).

Table 1: SIGs contributions types

Contribution types	Contributions	Contributions description
Positive contributions	Make contribution	A strongly positive contribution, adequate enough to satisfy the soft-goal.
	Help contribution	A positive contribution can help to satisfy the soft-goal, it's inadequate to introduce full satisfaction.
	Some+ contribution	Either make or help positive contribution to satisfy the soft-goal.
Negative contributions	Break contribution	A strongly negative contribution, adequate enough to deny the soft-goal.
	Hurt contribution	A partial negative contribution, but it's inadequate by itself to introduce fully soft-goal deny.
	Some- contribution	Either break or hurt negative contribution to satisfy the soft-goal.

The parent soft-goals may be connected with the child goals by one of the following three links (see Figure 1):

- AND decomposition: the parent soft-goal is decomposed into more than one related goal and it's satisfied if all the related goals are satisfied.
- OR decomposition: the parent soft-goal is decomposed into more than one related goal and it's satisfied if at least one related goal is satisfied.
- Equal decomposition: the parent soft-goal is decomposed into one related goal and it's satisfied if the linked goal is satisfied.

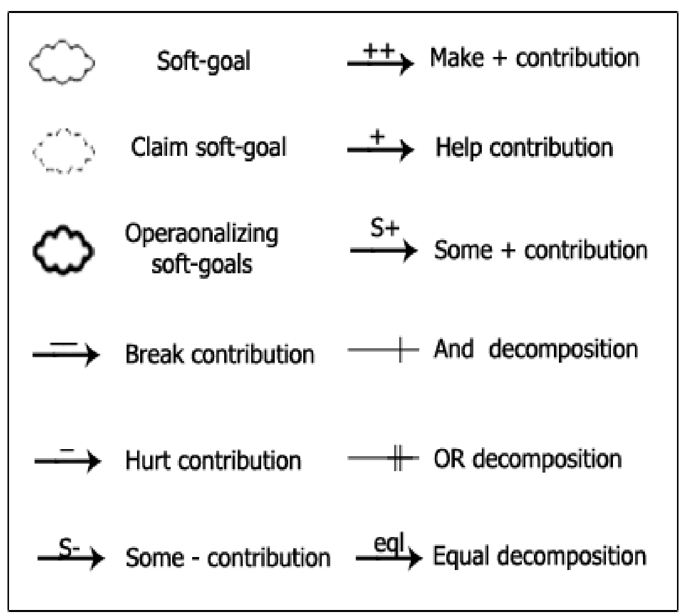

Figure 1: Soft-goal Interdependency Graphs [23]

B. Design standards-based framework for system performance requirements at functions level

At this level, the SIGs and the ISO 19761 are used for modeling and measuring the system performance requirements and their related software performance requirements. for simplicity, the proposed framework are divided into four sub-models: each sub-model clarifies one system performance requirements. Figure 2 shows a full view of the performance framework at functions level.

- The main memory time function may exchange data in a direct way with the storage device time function using COSMIC EXIT and ENTRY data movements or it may exchange data in an indirect way through the persistent storage using COSMIC READ and WRITE data movements.
- The storage device time function may exchange data in a direct way with the main memory time function and the processor instruction execution function using COSMIC EXIT and ENTRY data movements or it may exchange data in an indirect way through the persistent storage using COSMIC READ and WRITE data movements
- The processor instruction execution function may exchange data in a direct way with the storage device time function using COSMIC EXIT and ENTRY data movements or it may exchange data in an indirect way through the persistent storage using COSMIC READ and WRITE data movements
- The main memory time function, the storage device time function and the processor instruction execution function can require data from all the functions in the overall performance framework through the intermediary service using COSMIC EXIT and ENTRY data movements.
- The system scalability function may exchange data in a direct way with the concurrency function using COSMIC EXIT and ENTRY data movements or it may exchange data in an indirect way through the persistent storage using COSMIC READ and WRITE data movements.
- The concurrency function may exchange data in a direct way with the system scalability function using COSMIC EXIT and ENTRY data movements or it may exchange data in an indirect way through the persistent storage using COSMIC READ and WRITE data movements.
- The system scalability function and the concurrency function may require data from any function in the overall performance framework through the intermediary service using COSMIC EXIT and ENTRY data movements.
- The absolute performance error function may exchange data in a direct way with the performance stability error function using COSMIC EXIT and ENTRY data movements or it may exchange data in an indirect way through the persistent storage using COSMIC READ and WRITE data movements.
- The performance stability error function may exchange data in a direct way with the absolute performance error function using COSMIC EXIT and ENTRY data movements or it may exchange data in an indirect way through the persistent storage using COSMIC READ and WRITE data movements.
- The absolute performance error function and the performance stability error function may require data from any function in the overall performance framework through the intermediary service using COSMIC EXIT and ENTRY data movements.
- The absolute knowledge error may exchange data in a direct way with the relative knowledge error function using COSMIC EXIT and ENTRY data movements or it may exchange data in an indirect way through the persistent storage using COSMIC READ and WRITE data movements.
- The relative knowledge error function may exchange data in a direct way with the absolute performance error function using COSMIC EXIT and ENTRY data movements or it may exchange data in an indirect way through the persistent storage using COSMIC READ and WRITE data movements.
- The absolute knowledge error function and the relative knowledge error function may require data from any function in the overall performance framework through the intermediary service using COSMIC EXIT and ENTRY data movements.
- The response time function may exchange data in a direct way with the settling time function using COSMIC EXIT and ENTRY data movements or it may exchange data in an indirect way through the persistent storage using COSMIC READ and WRITE data movements.
- The settling time function may exchange data in a direct way with the response time function and the tracking error function using COSMIC EXIT and ENTRY data movements or it may exchange data in an indirect way through the persistent storage using COSMIC READ and WRITE data movements.
- The tracking error function may exchange data in a direct way with the settling time function using COSMIC EXIT and ENTRY data movements or exchange data in an indirect way through the persistent storage using COSMIC READ and WRITE data movements.
- The response time function, the settling time function and the tracking error function may require data from any function in the overall performance framework through the intermediary service using COSMIC EXIT and ENTRY data movements.
- The bandwidth function may exchange data in a direct way with the workload function using COSMIC EXIT and ENTRY data movements or it may exchange data in an indirect way through the persistent storage using COSMIC READ and WRITE data movements.
- The workload function may exchange data in a direct way with the bandwidth function using COSMIC EXIT and ENTRY data movements or it may exchange data in an indirect way through the persistent storage using COSMIC READ and WRITE data movements.
- The bandwidth function and the workload function may require data from any function in the overall performance framework through the intermediary service using COSMIC EXIT and ENTRY data movements.

Measurement observations

From Figure 2, the following points can be observed for measurement purposes:

- In the direct data exchange situation, each EXIT and ENTRY data movement will be assigned by 1CFP.

- In the indirect data exchange situation, each READ and WRITE data movement will be assigned by 1CFP.
- To require data through intermediary service that requires using 4 EXITS and 4 ENTRIES. Such process will be assigned by 4 CFP.

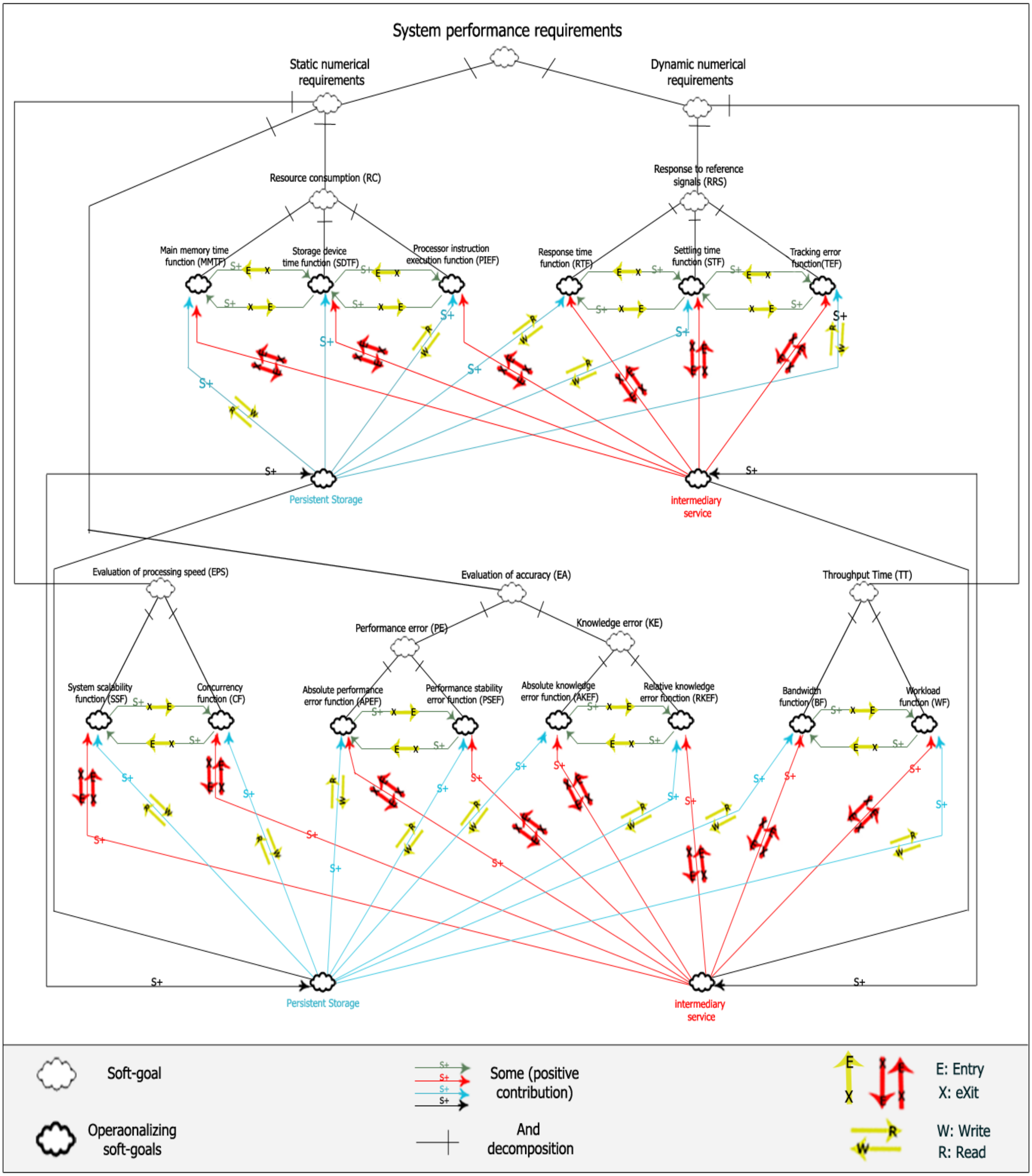

Figure 2: The full view of the system performance model at the functions level

V. Conclusion and Future Work

Developing software systems with quality attributes level is considered a significant factor to increase the value of the system. Several researchers have referred in their reports to the difficulties and challenges that faced the developers and limited their ability in identifying, addressing and measuring the NFR during early development phases, for instance: taking such requirements as a quantitative input. Recently, the researchers have introduced an extensive work for dealing with the NFR in different development phases; nevertheless, there is no based framework that can be used to facilitate early dealing with the NFR challenges and difficulties.

In this research work, we extends our previous work on the portability and maintainability NFR reported in [1] and [4] to cover the performance NFR; Where our main contribution from this work is achieved in proposing standards-based framework for identifying, specifying and measuring the system performance requirements. The proposed standards-based framework introduce to the developers three main contributions: 1) assisting in identifying and specifying the system performance requirements, 2) allocating the system performance requirements to the related software performance requirements, 3) measuring the allocated software performance requirements with an ISO-recognized measurement unit.

Our future work aims at extending this work to introduce integrated standards-based frameworks that cover all the NFR types. Also, we would like to present an experimental study to prove the effectiveness of using our standards-based frameworks during the systems development phases.

References

[1] Al-Sarayreh, K. T., A. Abran and and J. J. Cuadrado-Gallego," A Standards-based model of system maintainability requirements", Journal of Software: Evolution and Process, John Wiley & Sons, Ltd, 2013. http://dx.doi.org/10.1002/smr.1553

[2] Meridji, Kenza, Khalid T. Al-Sarayreh, and Ahmad Al-Khasawneh. "A generic model for the specification of software reliability requirements and measurement of their functional size." *International Journal of Information Quality* 3, no. 2 (2013): 139-163.

[3] Al-Sarayreh, Khalid T., Ibrahim Al-Oqily, and Kenza Meridji. "A standard-based reference framework for system operations requirements." *International Journal of Computer Applications in Technology* 47, no. 4 (2013): 351-363.

[4] Abran, A., K. T. Al-Sarayreh, and J. J. Cuadrado-Gallego, " A Standards-based Reference Framework for System Portability Requirements", Computer Standards and Interface, Elsevier, 2013. http://dx.doi.org/10.1016/j.csi.2012.11.003

[5] Nixon, Brian A., "Management of performance requirements for information systems," *Software Engineering, IEEE Transactions on* , vol.26, no.12, pp.1122,1146, Dec 2000

[6] Chih-Wei Ho; Johnson, M.J.; Williams, L.; Maximilien, E.M., "On agile performance requirements specification and testing," *Agile Conference, 2006* , vol., no., pp.6 pp.,52, 23-28 July 2006C. J. Kaufman, Rocky Mountain Research Lab., Boulder, CO, private communication, May 1995.

[7] Xiaoping Che; Maag, S., "Passive Testing on Performance Requirements of Network Protocols," *Advanced Information Networking and Applications Workshops (WAINA), 2013 27th International Conference on* , vol., no., pp.1439,1444, 25-28 March 2013

[8] Honghong Jiang; Xiaohu Yang, "Performance requirement elicitation for financial information system based on ontology," *TENCON 2009 - 2009 IEEE Region 10 Conference* , vol., no., pp.1,5, 23-26 Jan. 2009

[9] Meyer, F.J., "Performance Requirements for Ionospheric Correction of Low-Frequency SAR Data," *Geoscience and Remote Sensing, IEEE Transactions on* , vol.49, no.10, pp.3694,3702, Oct. 2011

[10] Guo Zhou; Huibing Zhao; Weiwei Guo, "Safety requirements analysis and performance verification of hot standby system using colored Petri-net," *Industrial Electronics and Applications (ICIEA), 2013 8th IEEE Conference on* , vol., no., pp.656,661, 19-21 June 2013

[11] Brachmann, F., "About performance requirements set against consumer-grade geolocation technologies," *System Science and Engineering (ICSSE), 2013 International Conference* , vol., no., pp.309,312, 2013

[12] IEEE Standard for Synchrophasor Measurements for Power Systems -- Amendment 1: Modification of Selected Performance Requirements," *IEEE Std C37.118.1a-2014 (Amendment to IEEE Std C37.118.1-2011)* , vol., no., pp.1,25, April 30 2014

[13] Rao, N.M.; Narayanan, R.; Vasudevamurthy, B.R.; Das, S.K., "Performance requirements of present-day distribution transformers for Smart Grid," *Innovative Smart Grid Technologies - Asia (ISGT Asia), 2013 IEEE* , vol., no., pp.1,6, 10-13 Nov. 2013

[14] Jun Li; Miao Yu; Yanyan Yang, "Cement Porter's Protective Clothing Part I: Performance Requirements and Tests for the Fabric," *Biomedical Engineering and Computer Science (ICBECS), 2010 International Conference on* , vol., no., pp.1,4, 23-25 April 2010

[15] Lira, R.; Mycock, C.; Wilson, D.; Kang, H., "PMU performance requirements and validation for closed loop applications," *Innovative Smart Grid Technologies (ISGT Europe), 2011 2nd IEEE PES International Conference and Exhibition on* , vol., no., pp.1,7, 2011

[16] ECSS-E-40-Part-1B, Space Engineering: Software — Part 1 Principle and Requirements, European Cooperation for Space Standardization, the Netherlands, 2003.

[17] ECSS-E-40-Part-2B, Space Engineeing: Software-part 2 Document Requirements Definitions, European Cooperation for Space Standardization, The Netherlands, 2005.

[18] ECSS-ESA, Tailoring of ECSS, Software Engineering Standards for Ground Segments, Part C: Document Templates, ESA Board of Standardization and Control (BSSC), 2005

[19] ECSS-E-ST-10C, Space engineering: system engineering general requirements, Requirements & Standards Division Noordwijk, The Netherlands, 2009.

[20] ECSS-Q-ST-80C, Space Product Assurance: Software Product Assurance, Requirements & Standards Division Noordwijk, The Netherlands, 2009.

[21] IEEE-Std-830, IEEE Recommended Practice for Software Requirements Specifications, 1998.

[22] ISO/IEC-19761, "Software Engineering - COSMIC v 3.1 - A Functional Size Measurement Method", International Organization for Standardization, Geneva (Switzerland), 2011

[23] Chung L, et al. Nonfunctional requirements in software engineering. Kluwer Academic Publishing, 2000.