



HAL
open science

A method for formalizing requirements interoperation in complex systems engineering

Anderson Luis Szejka, Alexis Aubry, Hervé Panetto, Osiris Canciglieri Júnior,
Eduardo Rocha Loures

► To cite this version:

Anderson Luis Szejka, Alexis Aubry, Hervé Panetto, Osiris Canciglieri Júnior, Eduardo Rocha Loures. A method for formalizing requirements interoperation in complex systems engineering. INSIGHT - International Council on Systems Engineering (INCOSE), 2015, 18 (4), pp.28-30. 10.1002/inst.12058 . hal-01238407

HAL Id: hal-01238407

<https://hal.science/hal-01238407>

Submitted on 4 Dec 2015

HAL is a multi-disciplinary open access archive for the deposit and dissemination of scientific research documents, whether they are published or not. The documents may come from teaching and research institutions in France or abroad, or from public or private research centers.

L'archive ouverte pluridisciplinaire **HAL**, est destinée au dépôt et à la diffusion de documents scientifiques de niveau recherche, publiés ou non, émanant des établissements d'enseignement et de recherche français ou étrangers, des laboratoires publics ou privés.

A method for formalizing requirements interoperation in complex systems engineering

Anderson Luis Szejka, (Anderson.Szejka@pucpr.br); Alexis Aubry (Alexis.Aubry@incose.org); Hervé Panetto (Herve.Panetto@incose.org); Osiris Canciglieri Júnior (Osiris.Canciglieri@pucpr.br); and Eduardo Rocha Loures (Eduardo.Loures@pucpr.br)

Systems Engineering (SE) is “*an interdisciplinary approach and means to enable the realization of successful systems*” that must meet and satisfy the needs of its stakeholders and it is based on concepts from systematic paradigms, methods and standardized processes (BKCASE, 2015, ISO/IEC 15288, 2015, ISO/IEC 29148, 2011). The continuing evolution of systems engineering comprises the consolidation, identification and formalization of new methods and modelling techniques, which can be applied in Product Development Engineering (PDE), Process Development (PD) and Software Development (SD). There is a significant amount of researchers working to formalize and standardize engineering design and to manage system life cycle when, in matter of fact, the attention should be drawn toward the information’s comprehension. Moreover, there is currently a debate regarding the semantic problems across different life cycle phases, which occur because of information’s misinterpretation and misunderstanding.

Each system engineering starts with a requirements elicitation and negotiation phases in which the main objective is to translate initial customers’ needs into functional and non-functional stakeholders requirements. Requirements are statements from the customers’ needs to identify and constraint a product, system or process (BKCASE, 2015). They must be unambiguous, clear, unique, consistent, stand-alone, measurable, verifiable and traceable, i.e., SMART (Specific, Measurable, Attainable, Realizable, Time bounded) requirements (Mannion and Keepence, 1995). Requirements are the entry point for the process production of specification analysis to a solution system with verified and validated solutions that meet initial needs. Functional and non-functional requirements are enriched with technical information, transforming into technical functional or non-functional requirements across different phases of system life cycle. However, this transformation involves, throughout its system life cycle, a set of heterogeneous knowledge, information and expertise. This results in a semantic gap between the initial requirements and the requirements that are taken into account during the life cycle, compromising the results of solution system. This semantic gap is caused by the non-interoperability of information within each requirement.

IEEE (1990) defines interoperability as “*the ability of two or more systems or components to exchange information and to use information that has been exchanged without special effort*”. In terms of categorization, interoperability has three levels (Panetto and Molina, 2008; EIF, 2004) as follows:

- (i) Technical Interoperability (TI) – it concerns technical properties, enabling machine-to-machine communication to take place. TI is usually associated with hardware and software components, systems and platforms (e.g. data and protocols format, physical characteristics, material resistance);
- (ii) Semantic Interoperability (SI) – it concerns the real meaning of content that is shared and understandable by any other application;
- (iii) Organizational Interoperability (OI) – it is the ability of organizations to effectively communicate and transfer (meaningful) data (information) even though they may use a variety of information systems over widely different infrastructures in or out of enterprises boundaries.

Semantic interoperability is feasible when the meaning associated to the captured information and knowledge can be effectively shared across different workgroups without any loss of meaning and knowledge (Chungoora et al., 2013). This can be done through the construction of formal domain ontologies (Gruber, 1995; Provine et al., 2004; Jovanovic and

Gasevic, 2005; Noy and Rubin, 2008), implemented to different fields such as engineering, medicine, business, etc. Applied to Systems Engineering, the research questions thus concern “*How heterogeneous information related to requirements can be formalised regarding to multiple knowledge domains to provide support during different phases of System Development Life Cycle?*” and “*What are the formalised relationships between system requirements related to multiples domains and impacting in different phases of Life Cycle?*”

This article presents an ongoing research project that aims to define a conceptual method to formally model the system requirements and their relationships in terms of transformation (translation, conversion and sharing) and traceability, based on an ontological approach. In this way, we are considering two hypothesis: (H1) *Multiple phases of System Life Cycle can be supported by system requirements in a semantically interoperable manner*; and (H2) *System requirements formalization can ensure the comprehensibility and verifiability, reducing inconsistencies between different domains across the phases of System Life Cycle*.

The approach needs to discover and identify the dependence relationship during any Systems Engineering. Thus, according to ISO/IEC 15288, 2008, ISO/IEC 29148, 2011 and discussion provided in Szejka et al. (2014), three perspectives are necessary to establish the system-project: (i) the domain of application; (ii) the system life cycle phase; and (iii) the requirements constraining the studied system. The first perspective concerns the set of heterogeneous knowledge and expertise involved during the Systems Engineering, i.e., mechanical expertise, electrical expertise, IT expertise, and so on. The second perspective refers to different phases of system life cycle, where each phase has its proper constraints and information. The last perspective considers the consistency of the relationships between requirements since a specific requirement is dependent of one or multiple domains and one or multiple phases of life cycle.

Based on this context, we can directly identify three interoperation issues. The first concerns the heterogeneity of information coming from multiple domains. It imposes some knowledge representation and analysis for managing requirements and their semantic relationships then, this is associated to hypothesis H2. The second interoperation issue concerns the system life cycle phases and the possible impacts between some requirements associated to different life cycle phases. The requirement definition can impact on other requirements and then, it is necessary to manage and to ensure the consistency of those requirements, so this is related to hypothesis H2. The last interoperation issue concerns the relationships between requirements and their properties of completeness, coherency, uniqueness, univocity and traceability that it is correlated to the hypothesis H1.

Currently, there are a significant number of researchers that are working in this area (Ratchev et al., 2003; Baudry et al., 2007; Canciglieri Jr. and Young, 2010; Chungoora et al., 2013). However, these teams focus their research on specific points to transform information between domains and/or single phases of system life cycle and/or requirements. In particular, requirements interoperation is the core concerns of industry when they are engineering a system (Micouin, 2008; Bernard, 2012). The main issue is to pragmatically formalize the requirements interoperation, based on ontological models, considering the tacit knowledge related to the processes involved in Systems Engineering. Thus specialists typically define requirements for project-systems using some informal document written in natural language (NL), because of the expression richness provided by it (Bryant, 2000). They are focused on establishing the references for the needs of systems engineering, in order to ensure the requirements completeness, consistency and coherency.

The main result of this research is to transform functional and non-functional requirements, written in natural language (informal requirements), into formal requirements (Figure 1). It is based on formal Common Logic (CL) (Pan and Liu, 2010; Jarrar, 2007) and ontology application for transforming requirements written in natural language into requirements written in formal language.

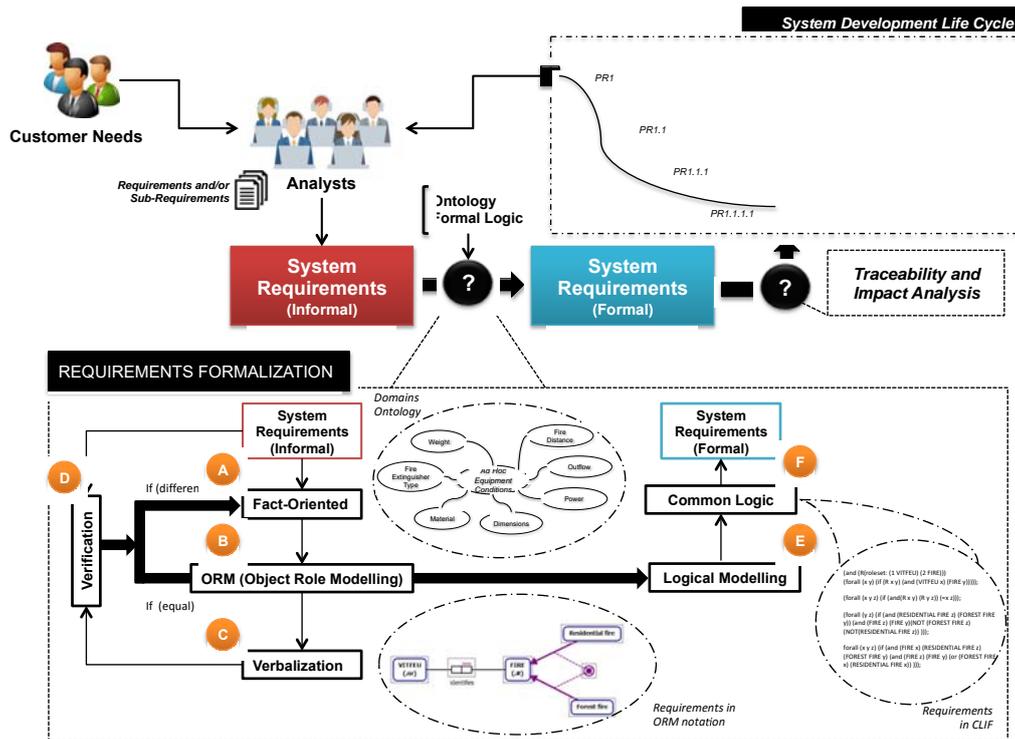


Figure 1. Requirements formalization approach

The proposed method is comprised of sub-methods and procedures that they perform in a semi-automatic manner. Firstly, requirements written in natural language are split in ‘subject + verb + complement’ for building a fact-oriented model (FOM) (Halpin, 2006). The requirements are analysed extracting the facts of interest or concepts (‘subject’ / ‘complement’) and relationships (‘verb’) between these facts (Detail A – Figure 1). These relationships can be unary, binary or ternary (Halpin, 2006). Sequentially, these facts must be modelled in a formal logic in order to structure these requirements. This fragmentation puts in a simpler manner the information within each requirement.

Different approaches can be used to structure this fact-oriented model such as: Object Role Modelling (ORM) (Detail B – Figure 1) and Cognition enhanced Natural language Information Analysis Method (CogNIAM) (Bollen, 2014). However, the method chosen must be able to model and verbalize the requirements, since it is important to compare the requirements model structured and the requirements originally defined (Detail C – Figure 1). So, it allows the identification of any discrepancy between the model and real desires (Detail D – Figure 1). Although these approaches rely on structured language, they are informal languages. Pan and Liu (2010) mapped some fact from Object Role Modelling into First-Order Logic (FOL). The latter has powerful expression to represent complex rules in a formal way. Some analysed research (Jarrar, 2007) provide methods to model those approaches in Description Logic or Common Logic that are based on First-Order Logic, and both methods have a inference machine. Inference machine is important to classify information and query answering. The presented method is based on Common Logic through mapping constraints enriched based on Common Logic Interchange Format (CLIF) (Detail E – Figure 1). This formal structure is linked with other requirements creating a knowledge model and possibilities of applying traceability methods and conflicts identification methods between requirements (Detail F – Figure 1).

As a conclusion, the presented approach demonstrates that there are significant lacks of interoperability between requirements in SE due to heterogeneity of information from multiple domains and different phases of systems life cycle. We identified three interoperation issues cross-domains, cross-systems life cycle and cross-requirements. These interoperation issues

have a direct impact in design success, because they can cause mistakes and misinterpretation with requirements. Moreover, it was proposed a method dealing with requirements interoperability across different phase of the life cycle and domains. This method aims to formally model requirements interoperation in terms of transformation, traceability and conflicts analysis. The goal is now to enrich this method and evaluate its performance with different case studies.

References

- Baudry, B., Nebut, C., Le Traon, Y. 2007. "Model-Driven Engineering for Requirements Analysis." In: Enterprise Distributed Object Computing Conference, 2007. EDOC 2007. 11th IEEE International: 459-467.
- Bernard, Y. 2012. Requirements management within a full model-based engineering approach. *Systems Engineering* 15 (2) 119-139.
- Bkcase, Editorial Board. 2015. "The Guide to the Systems Engineering Body of Knowledge (SEBoK)", Version 1.3.2, R.D. Adcock (EIC), Hoboken, The Trustees of the Stevens Institute of Technology.
- Bollen, P. 2014. "Fact-Based Semantic Modelling in the Information and Behavioural Perspectives." OTM 2014 Workshops LNCS 8842: 663-666.
- Bryant, B. R. 2000. "Object-Oriented Natural Language Requirements Specification." In Proceedings of Computer Science Conference. Canberra, ACT: 24-30.
- Canciglieri Jr., O. and Young R. I. M. 2010. "Information mapping across injection moulding design and manufacture domains." *International Journal of Production Research* 48 (15): 1565-1586.
- Chungoora, N., Young, R. I. M., Gunendram, G., Palmer, C., Usman, Z., Cutting-Decelle, A., Harding, J., Case, K. 2013. "A model-driven ontology approach for manufacturing system interoperability and knowledge sharing", *Computers & Industry* 64 (4): 392-401.
- EIF, European Interoperability Framework for pan-European eGovernment Services. 2004. "Interoperable Delivery of European eGovernment Services to public Administrations". *Business and Citizens (IDABC)*: 1-25.
- Gruber, T. R. 1995. "Toward Principles for the Design of Ontologies Used for Knowledge Sharing." *International Journal of Human-Computer Studies* 43 (5-6): 907-928.
- Halpin, T. 2006. "Object-Role Modelling (ORM/NIAM)." *International Handbooks on Information Systems* Springer: 81-103.
- IEEE (Institute of Electrical and Electronics Engineers). 1990. *IEEE Standard Computer Dictionary: A Compilation of IEEE Standard Computer Glossaries*.
- ISO (International Organization for Standardization) and IEC (International Electrotechnical Commission). 2011. *ISO/IEC 29148. "System and software engineering – Life cycle processes – Requirements Engineering"*. Geneva, CH: ISO/IEC.
- ISO (International Organization for Standardization) and IEC (International Electrotechnical Commission). 2015. *ISO/IEC 15288. "Systems and software engineering – System life cycle processes"*. Geneva, CH: ISO/IEC.
- Jarrar, M. 2007. "Towards automated reasoning on ORM schemes mapping ORM into the DLRidf description logic." In *Conceptual Modelling – ER2007* 4801: 181-197.
- Jovanovic, J. and Gas̄evic, D. 2005. "Achieving knowledge interoperability: an XML/XSLT approach". *Expert Systems with Applications* 29 (3): 535-553.
- Mannion M. and Keepence, B. 1995. "SMART Requirements," *Software Engineering Notes* 20 (2): 42-47.
- Micouin P. 2008. "Toward a property based requirements theory: System requirements structured as a semilattice," *Systems Engineering* 11 (3): 235-245.
- Noy, N.F. and Rubin, D.L. 2008. "Translating the foundational model of anatomy into OWL." *Web Semantics: Science, Services and Agents on the World Wide Web* 6 (2):133-136.
- Pan, W. and Liu, D. 2010. "Mapping Object Role Modelling into Common Logic Interchange Format." In *3rd International Conference on Advanced Computer Theory and Engineering (ICACTE)* 2: 104-109.
- Panetto, H. and Molina, A. 2008. "Enterprise Integration and Interoperability in Manufacturing Systems: trends and issues". *Computers in Industry* 59 (7): 641-646.
- Provine, R.; Schlenoff, C.; Balakirsky, S.; Smith, S.; Uschold, M. 2004. *Ontology-based methods for enhancing autonomous vehicle path planning. Robotics and Autonomous Systems* 49 (1/2): 123-133.
- Ratchev, S.; Urwin, E.; Muller, D.; Pawar, K. S.; Moulek, I. 2013. "Knowledge based requirement engineering for one-of-a-kind complex systems." *Knowledge-Based Systems* 16 (1): 1-5.
- Szejka, A. L.; Aubry, A.; Panetto, H.; Canciglieri Jr., O.; Loures, E. R. 2014. "Towards a conceptual framework for requirements interoperability in complex systems engineering." OTM 2014 Workshops LNCS 8842: 229-240.