



HAL
open science

Requirements interoperability method to support integrated product development

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

► **To cite this version:**

Anderson Luis Szejka, Osiris Canciglieri Júnior, Eduardo Rocha Loures, Hervé Panetto, Alexis Aubry. Requirements interoperability method to support integrated product development. The 45th International Conference on Computers & Industrial Engineering, CIE45, Oct 2015, Metz, France. hal-01224947

HAL Id: hal-01224947

<https://hal.science/hal-01224947>

Submitted on 5 Nov 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.



REQUIREMENTS INTEROPERABILITY METHOD TO SUPPORT INTEGRATED PRODUCT DEVELOPMENT

A.L. Szejka^{1,2,3*}, O. Canciglieri Júnior^{1**}, E.R. Loures¹, H. Panetto^{2,3}, A. Aubry^{2,3}

¹Graduated Program in Production and Systems Engineering (PPGEPS)

Pontifical Catholic University of Parana (PUCPR), Curitiba, Brazil

[{anderson.szejka, osiris.canciglieri, eduardo.loures}@pucpr.br](mailto:anderson.szejka,osiris.canciglieri,eduardo.loures@pucpr.br)

²Research Centre for Automatic Control of Nancy (CRAN)

University of Lorraine, Vandoeuvre-lès-Nancy Cedex, France

³CRAN UMR 7039, CNRS, France

[{herve.panetto, alexis.aubry}@univ-lorraine.fr](mailto:herve.panetto,alexis.aubry@univ-lorraine.fr)

ABSTRACT

Nowadays, more and more the complexity of product has increased. Moreover, product development must meet all costumers' needs. This complexity has required the establishments of different patterns with additional enterprises to complement their skills. However, the definition of the different patterns with heterogeneous knowledge can share information is important to avoid the risk of misinterpretation. So, product requirements must consider all constraints during the product realization. In this way, researches have identified that there are different issues related to requirement analysis and maintenance during the product realisation. In order to cope with these issues, a conceptual approach to formally model the requirements interoperability is proposed in terms of transformations and traceability. Initially, the authors have proposed a characterisation of product to identify the requirements interoperability issues and the results were: (i) domains; (ii) product life cycle; and (iii) product requirements. Based on these characterisation, the conceptual approach used a formal logic descriptions and ontology application to transform requirements written in natural language to requirements written in formal language (mathematical language). As a result, the research presented an overview of the existing gaps in one or more requirements interoperability to cope with the requirements inconsistency problem in multiples perspectives in product development.

Keywords: Product Requirements, Integrated Product Development Engineering, Interoperability, Formal Models, and Requirements Analysis.

1 INTRODUCTION

The complexity of Product Development Engineering (PDE) process required to meet customers' needs has increased over the years, together with the complexity of the products themselves. A product is an artefact that can be offered to a customer as something tangible (e.g., physical objects) or intangible (e.g., services and software) (Kotler et al. [1]; Magrab et al. [2]). Thus, in global scenario whose the time-to-market and quality are important to the enterprises thriving, a product must be designed and manufactured correctly throughout all phases of product life cycle. Therefore, it is necessary to have systems that can provide information to support all phases of product development.

* Ph.D. Candidate of Production and Systems Engineering Graduate Program (PPGEPS) at Pontifical Catholic University of Parana (PUCPR) and Ph.D Candidate of Automatic, Signal and Image Processing, Computer Engineering at University of Lorraine; Rua Imaculada Conceição, 1155, Prado Velho, Curitiba, CEP 80215-901, PR, Brazil; Phone: +55 (41) 32712425.

** Professor of Production and Systems Engineering Graduated Program (PPGEPS) at Pontifical Catholic University of Parana (PUCPR); Rua Imaculada Conceição, 1155, Prado Velho, Curitiba, CEP 80215-901, PR, Brazil; Phone: +55 (41) 3271-1304.



PDE begins with the definition of its concepts whose objective is to translate initial customer's needs into functional or non-functional requirements. This is the entry point for the process of producing a specification analysis of a product, process or system, with verified and validated solutions (technical information) that meet the initial customer's needs. Furthermore, requirements must be unambiguous, clear, unique, consistent, measurable and verifiable (Bkcase [3]). However, the definition of these products, throughout their life cycle, a set of heterogeneous knowledge and expertise (mechanical engineering, electrical engineering, computer sciences, business, etc.) and different enterprises, sharing information, expertise and resources to solve engineering problems.

This then has resulted in semantic gap, since the requirements originally identified are not the same as those that were taken into account during later of product life cycle. Misinterpretation has occurred during the life cycle, committing the final product. Thus, the tacit information embedded in each requirement must be extracted and explained into an understandable way. Moreover, the relationships among product requirements must be mapped and traced to identify the requirements impact across different phases of product life cycle. So that, this research proposes a conceptual method to formally model product requirements and their relationships in an integrated product development engineering (IPDE), providing information support to multiple perspectives. These perspectives include: (i) information heterogeneity from multiple domains; and (iii) settings of information to different phases of life cycle.

The following section addresses the problem statement regarding the need to keep the consistency and coherency in requirements sharing when multiple domains and different life cycle phases are involved. Section 3 presents a literature review concerning the IPDE and product requirements and knowledge formalization. Section 4 is devoted to propose an approach to formally model the requirements interoperations. Finally, section 5 concludes and presents perspectives for the research continuity.

2 GENERAL ASSESSMENT OF PRODUCT REQUIREMENTS ISSUES

In an IPDE process, multiples specialists with distinct skills work together, sharing a set of heterogeneous knowledge and information, during different phases of the product life cycle (ISO/IEC 15288 [4]; ISO/IEC 29148 [5]). This process must respect all product requirements that have all product constraints. However, due to this heterogeneity of information, misinterpretation and misunderstanding have been occurred between the real customers' desires and product's performed. This semantic gap is caused by the non-interoperability of information of product requirements throughout product life cycle.

To Chungoora [6], a semantic interoperability is feasible when the meaning associated to captured information and knowledge can be effectively shared across different workgroups without any loss of meaning and knowledge during the information exchange. Thus, semantic interoperability of information is necessary to maintain the coherency and consistency of requirements to fulfil all customers' needs. Based on this, all parameters involved in PDE process need to be analysed to identify the issues in product requirements.

Figure 1 shows three information perspectives exchanging that must be considering in an IPDE: (i) domain of application; (ii) product life cycle; and (iii) product requirements. The first perspective concerns the heterogeneity of domains information involved during the IPDE process, for example mechanical systems, electrical systems, computer systems and so on. For this particular case, each specialist in these domains defines specific requirements based on their individual skills. The second perspective refers to different phases of the product life cycle, where each phase has its proper constraints represented by specific requirements. And the last one considers the consistency and coherency of the relation between requirements associated to a single domain and single phase of life cycle.

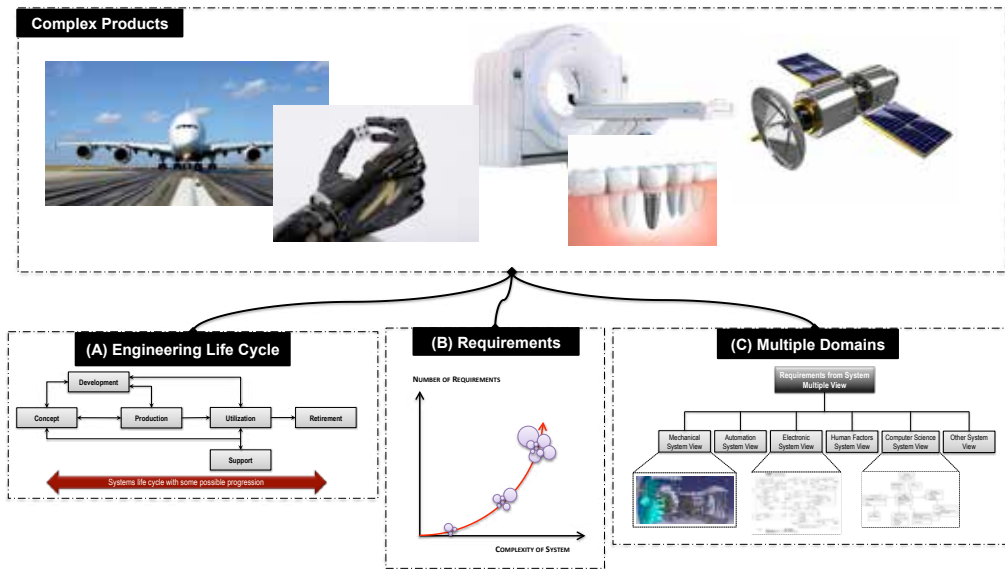


Figure 1: IPDE characterisation

According to this discussion, each requirement is dependent on, at least, a specific domain and phase of life cycle, as illustrate in Figure 2. Thus, based on the context discussed and IPDE characterisation, it is possible identified three interoperation issues. The first interoperability issue concerns the heterogeneity of information coming from multiple domains (Figure 2 - Detail A). It imposes some knowledge representation and analysis for managing product requirements and their semantic relationships. The second interoperability issue concerns the product life cycle (Figure 2 - Detail B). Each requirement is associated to a specific phase of product life cycle, but its constraints can influence in other requirements in different phases of life cycle, committing the consistency. The last interoperability issue concerns the relations between product requirements and its properties (completeness, coherency, uniqueness, univocity, verifiability and traceability associated to each of them - Figure 2 - Detail C).

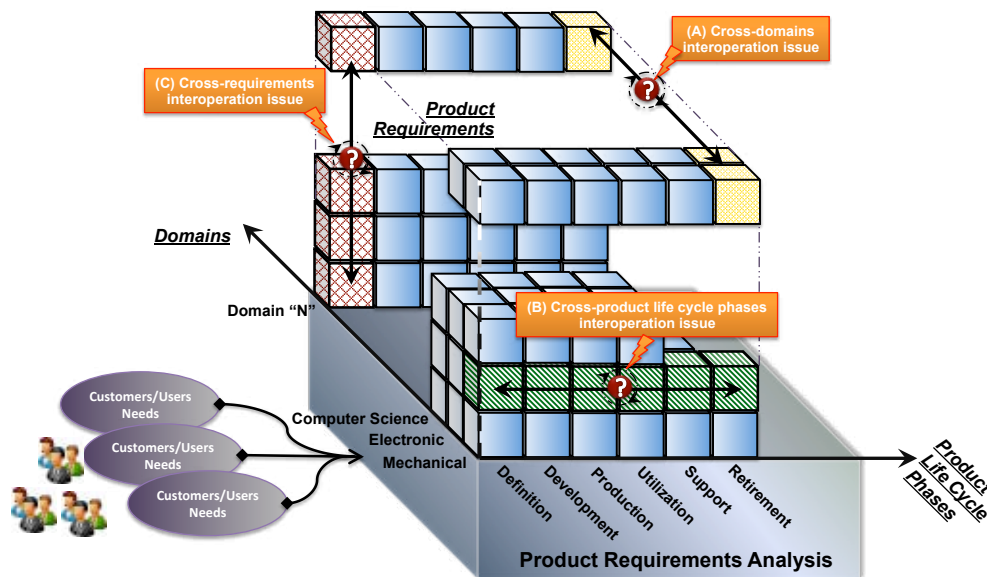


Figure 2: Product Requirements Interoperability Issues

In this context, Szejka et al. [7] has presented a literature review to identify the related works associated to cross-domains, cross-product life cycle and cross-requirements. The results achieved with this review were more than 100 scientific publications. Based on this review, a categorization has been made to identify the existing solutions and limitations on the semantic support in information sharing for these three perspectives, as shown on table 1.

Table 1. Related works classification according to each research issue Szejka et al. [7]

Authors and Publication Year	Cross-Domains issue			Cross-Product Life Cycle issue		Cross-Requirements issue		
	(D1)	(D2)	(D3)	(LC1)	(LC2)	(R1)	(R2)	(R3)
Adelson and Soloway, 1985	✓				✓			
Ramesh and lark, 2001					✓	✓		
Egyed and Grünbacher, 2002					✓	✓		
Cleland-Huang et al., 2002					✓	✓		
Canciglieri and Young, 2003	✓	✓		✓			✓	
Spanoudakis et al., 2004					✓	✓		
Ratchev et al., 2003				✓			✓	
Kececi, Garbajosa and Bourque, 2006					✓	✓	✓	
Schmidt, 2006	✓	✓			✓			
Holt and Pierry, 2008				✓		✓		
Stechert and Franke, 2008	✓	✓			✓	✓	✓	
Welsh and Sawyer, 2009					✓	✓		✓
Schamai et al., 2010				✓		✓	✓	
Moneva, Hamberg and Punter, 2011	✓	✓			✓			
Ahmad and Bruel, 2012					✓	✓	✓	✓
Bouffaron et al., 2012	✓			✓			✓	
Cmyrev et al., 2012				✓		✓	✓	
Strasunskas and Hakkarainen, 2012	✓	✓			✓	✓		
Oertel and Josko, 2012					✓	✓	✓	
Chandler and Matthews, 2013	✓	✓		✓				
Chungoora et al., 2013	✓	✓		✓			✓	
Liao, 2013	✓	✓		✓				
Haveman and Bonnema, 2013	✓	✓	✓	✓				
Barbieri et al., 2014	✓	✓		✓				

The adopted classification criteria were:

- **(D1) Particular cases** - Papers/articles concerning the requirements exchange limited to two specific domains;
- **(D2) Ability to be general** - Papers/articles concerning the requirements exchange among different domains and that can be adapted to other domains;
- **(D3) General approach** - Papers/articles concerning the requirements exchange among different domains whose approaches do not need any adaptation;
- **(LC1) Considering life cycle** - For papers/articles that concerns the requirement exchange among one or more phases of the product life cycle;
- **(LC2) Not considering life cycle** - For papers/articles that do not concern the requirement exchange among one or more phases of the product life cycle;
- **(R1) Requirements Traceability** - Papers/articles regarding the requirements traceability in one or more phases of product life cycle and different domains;
- **(R2) Requirements Interoperability** - Papers/articles regarding the exchange of requirements between one or more phase of product life cycle and different domains. This



interoperability issue does not consider any requirements changes during the product life cycle;

- **(R3) Requirements Inconsistency Impacts** - Papers/articles regarding the exchange of requirements between one or more phase of product life cycle and different domains. This interoperability issue considers the impacts caused by any requirements changes during the product life cycle.

According to Szejka et al. [7] and Table 1, there are some researched lacks in cross-domains (D3) and cross-requirements (R3). The first lack is concerning the General approach proposed, which makes evident the problem with the semantic gap in multiples domains as well as the risk of mistakes and misinterpretation. The second lack concerns the Requirements Inconsistency Impacts that will be the evaluation of the influence of a specific requirement in distinct domains and different life cycle phases. In order to ensure a complete requirements interoperation, it is necessary to consider: (i) an approach that allows information sharing between multiple domains; (ii) the requirements influence in different phases of product life cycle; and (iii) requirements traceability, requirements interoperability and requirements impacts. Based on the interoperation's issues as shown on Figure 2 and literature review that was categorized and illustrate in Table 1, two main questions have arisen:

- *How heterogeneous information related to requirements can be formalised regarding to multiple knowledge domains to provide support during different product life cycle phases?*
- *What are the formalised relationships between requirements related to multiples domains, and how they can ensure the consistency within different product life cycle phases?*

3 LITERATURE REVIEW

This section presents an overview of the relevant subjects to detail these questions. Section 3.1 discusses the subjects and relation between Product Development Engineering and Product Requirements to provide informational supporting to the proposed approach. Section 3.2 presents the components and methodologies to the information modelling and formalisation.

3.1 Product Development Engineering and Product Requirements

PDE process is a set of multidisciplinary activities structured to transform market opportunities, customers' needs and technological constraints in products design and manufacturing project (Smith [8]; Rozenfeld et al. [9]; Pereira [10]). PDE is a complex and interactive process because multiple domains and different enterprise's departments are concurrently involved during the product realisation.

This interactive process creates and uses information from different fields, specialists and partners' enterprise. Consequently, PDE systematization simplifies the number of interactions with action and activities necessary to achieve the project aim and structure the information necessary to each phase of PDE (Pereira [10] and Silva [11]). PDE requires a holistic view of the product, which involves multidisciplinary teams (marketing, design, engineering, business) with different viewpoints of a specific project (Rozenfeld et al. [9]). Furthermore, market competition, new technologies, product quality control from customers, manufacturer and suppliers affect concurrently the definitions, designs, evaluations, tests and manufacturing of a product (Fernandes [12]). To Rozenfeld et al. [9], the initial activities in PDE are the most costly (85% of product final cost) and offer higher risk of mistakes and uncertainties. Therefore, the improvement of information quality within product requirements (e.g. information unambiguous, clear, complete) can ensure the correct PDE during product life cycle, using formal standards and models to structure the information.

Requirement is a declaration from the needs of stakeholders and/or customers to identify a product, system or process characteristics or constraints, which is unambiguous, clear, unique, consistent, stand-alone and verifiable (Bkcase [3]). Initial stakeholders and/or customers' needs are not able to use as requirement, because they often have lacks of definition, analysis, coherency, consistency and feasibility. Thus, these needs must be refined, ratified and enriched with information to be a valid requirement. According to ISO/IEC 15288 [4], a requirement must be complete, coherent, unique, feasible, traceable and verifiable. Figure 3 illustrates requirements classification based on Bkcase [3], which provides support to structure product, system or process engineering. Each requirement matches a single part of the future product, system or process and is grouped in an appropriate combination of textual statement views.

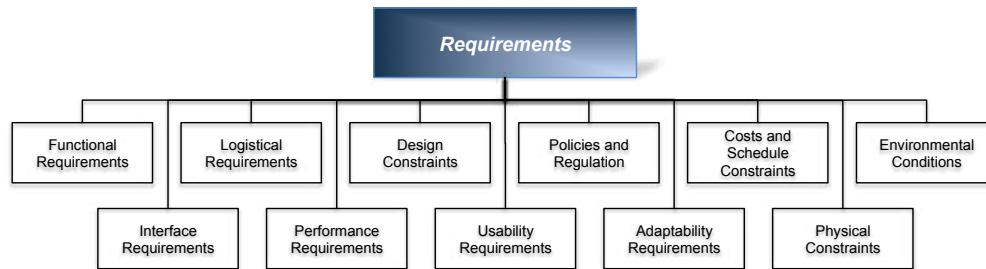


Figure 3: Requirements Classification Bkcase [3]

3.2 Knowledge Representations and Formalisation

Knowledge Representation (KR) is a subarea of Artificial Intelligence (AI) interested in understanding, designing, and implementing methods of information representing by computer science (Kamsu-Foguema et al. [13]). According to Shapiro [14], KR objectives are: (i) to transform implicit information into explicit information; (ii) to standardize heterogeneous information; and (iii) to formalize the knowledge. There are two classifications of works in KR according to schools of thoughts: logicist and anti-logicist. In Benacerraf and Putnam [15], the authors believe that logicist term comes from mathematics and uses the formal logic methods or concept methods for their representations, such as First Order Logic (Ben-Ari [16]), Description Logic (Baader et al. [17]), Second Order Logic and so on. Oppositely, according to Benacerraf and Putnam [15] anti-logicist do not formalise explicit plans to be executed by a system. They provide system ways of: (i) to interact with the environment; (ii) a set of skills; and (iii) a set of purposes.

In this way, according to Lezoche et al. [18] is necessary to understand the relationships between different concepts or models and unhide the tacit knowledge buried inside these concepts and models. To Lezoche et al. [18] developing conceptual models means specifying the essential objects, or components, of the system to be studied, the relationships of the objects that are recognised, the types of changes in the objects or their relationships, which affect the functioning of the system and the types of impact these changes, have on the system.

4 PROPOSED APPROACH TO FORMALLY MODEL REQUIREMENTS INTEROPERATION IN INTEGRATED PRODUCT DEVELOPMENT

Product Requirements must have all customers' needs to perform a product. These requirements have a set of heterogeneous information that are typically define by specialist using documents (data tables, presentations, texts) written in natural language (NL) to different phases of product life cycle (Eriksson, Börstler and Borg [19]). However, as earlier discussed semantic gaps are highlighted due to non-interoperability of information with product requirements in an IPDE. This occurs because NL tends to be unclear, incomplete and ambiguous (Bryant [20]). Thus, when other stakeholders exploit these requirements, they are always marked by subjective interpretation and

ambiguous, even if the verification process in requirements engineering tries to avoid these types of problem. According to Chen et al. [21], these issues are identified much later during the manufacturing phases.

Improvement information sharing and interoperability are being researched through the construction of formal domain ontology in different areas of knowledge, such as engineering, biomedicine, business (Ferrari and Madhavji [22]; Lin and Harding [23]; Nagahanumaiah and Ravi [24]). In this way, this paper is proposing to create a method to formally model the product requirements and their relationships across different phases of life cycle. Figure 4 is illustrating the structure of the method to transform informal requirements into formal requirements. This method is divided in two parts: (i) Requirements Conceptual Data Model; and (ii) Logical Model.

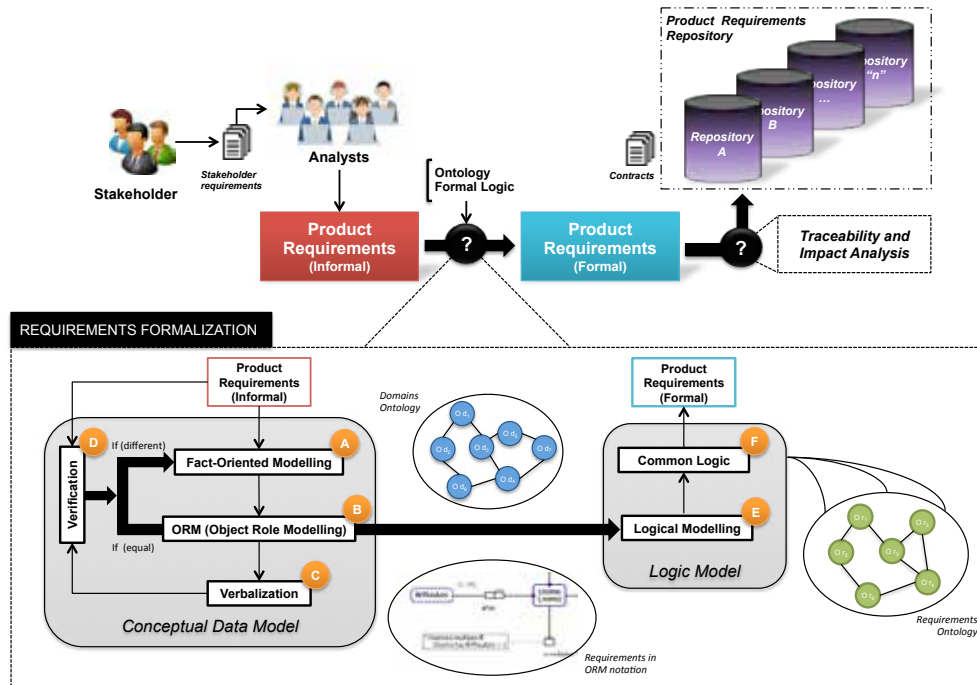


Figure 4: Product requirements formalization method

4.1 Conceptual Data Model of Product Requirements

In this part, the method consists in extract the information implicit in each requirement, identifying the main concepts of each sentence and dependence relations. Thus, requirements written in NL are split in *subject+verb+complement* for building a fact-oriented model (FOM). Each part of this requirement is analysed for extracting the facts of interest (subject and complement) and the relationships (verb), which they must be unary, binary and ternary, between these facts, according to Halpin [25]. These requirements fragmented are thus expressed in a simpler language than the initial ones. Furthermore, these facts can be linked with the domains ontology (Detail A - Figure 4).

To structure these facts, different approaches can be used such as: Object Role Modelling (ORM) (Detail B - Figure 4) and Cognition enhanced Natural language Information Analysis Method (CogNIAM) (Bollen [26]). From these approaches, the fact-oriented bases written in NL should be transformed to a conceptual level with objects (concepts) and roles (relationships among concepts). ORM approach has a property that it allows to transform the graphical model to NL (verbalization process - Detail C - Figure 4). Verbalization process is important to compare if the model created is equivalent to the initial requirements without modelling (Detail D - Figure 4).



4.2 Requirements Logic Model

This part of the method transforms (translate, convert and share) the graphical model requirements to logical models. Pan and Liu [27] mapped some graph concept into First-Order Logic (FOL). FOL has powerful expression to represent complex rules, but it does not have a reasoner. Some researches Jarrar [28] and Nguyen and Thanh [29] provide methods to model FOL in Description Logic (DL) or Common Logic (CL), and both methods have reasoner.

However, DL cannot express all constraints that are necessary for this formalization Pan and Liu [27]. Thus, these approaches may be formalised in CL through mapping constraints enriched based on Common Logic Interchange Format (CLIF) and ISO/IEC 24707 [30] (Detail E - Figure 4). In this context, ontology can be used to model the formal requirements in common logic (Detail F - Figure 4) and it will be used by different phases of system engineering to identify the inconsistencies between requirements.

5 CONCLUSIONS

This paper has presented a conceptual method proposal for requirements interoperation in IPDE. This approach aims to formally model the requirements interoperation from multiples domains and during different product life cycle phases in terms of transformation, traceability and impacts. Thus, it is important to ensure the information coherence and consistency in earlier steps of PDE, as it will avoid in later steps of engineering the risk of misinterpretation and mistakes between requirements.

In this context, we identified three interoperation issues: (i) cross-domains; (ii) cross-product life cycle; and (iii) cross-requirements. Furthermore, it is important to consider the simultaneous interrelationships between the last three issues. Based on these issues, a literature review about these issues has been presented, highlighting two main questions to be addressed. Therefore, we are proposing a conceptual methodology to structure the formalization of product requirements interoperations written in natural language (informal requirements) to formal logic requirements (formal requirements). This method is structured in two parts: (i) conceptual data model; and (ii) logical model. The first part is responsible for conceptual modelling requirements, extracting the main facts from each sentence and establishing the links between these facts. The second part transforms, converts or shares these requirements modelling to formal logical requirements.

6 ACKNOWLEDGMENTS

The authors would like to thank the Coordination of Improvement of Higher Education Personal - CAPES, under Science without Borders Program (Special Visiting Researcher) - Project 178/2012 and Pontifical Catholic University of Parana (PUCPR) for the financial support.

7 REFERENCES

- [1] Kotler, P.; Brown L.; Adam S.; Burton S.; Armstrong, G. 2011. Principles of Marketing. 14th Edition, Pearson Education Australia.
- [2] Magrab, E. B.; Gupta, S. K.; McCluskey, F. P.; Sandborn, P. 2010. Integrated product and process design and development: the product realization process. 2nd Edition. CRC Press.
- [3] 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.
- [4] ISO (International Organization for Standardization) / IEC (International Electrotechnical Commission). 2008. ISO/IEC 15288:2008: Systems and software engineering - System life cycle processes, Geneva, CH.
- [5] ISO (International Organization for Standardization) / IEC (International Electrotechnical Commission). 2011. ISO/IEC 29148:2011: System and software engineering - Life cycle processes - Requirements Engineering, Geneva, CH.



- [6] **Chungoora, N. 2010.** A framework to support semantic interoperability in product design and manufacture. 428 p. Thesis (Ph.D.) - University of Loughborough.
- [7] **Szejka, A.L.; Aubry, A.; Panetto, H.; Canciglieri Jr., O. and Loures, E.R. 2014.** Towards a conceptual framework for requirements interoperability in complex systems engineering, *OTM 2014 Workshops, LNCS*, 8842, pp 229-240.
- [8] **Smith, D.W. 2002.** Introducing EDG students to the design process. Proceedings of the 2002 Annual Midyear Meeting of the Engineering Design Graphics Division of the American Society for Engineering Education, pp 1-6.
- [9] **Rozenfeld, H. et al. 2006.** *Gestão de desenvolvimento de produtos: uma referência para a melhoria do processo*. 1st Edition, Saraiva.
- [10] **Pereira, J. A. 2014.** Modelo de desenvolvimento integrado de produto orientado para projetos de P&D do setor elétrico brasileiro. Thesis (Ph.D.) - Pontifical Catholic University of Parana.
- [11] **Silva, R. R. B. 2014.** Proposta de estruturação do processo de desenvolvimento de produtos para empresas prestadoras de serviço de telecomunicações. Dissertation (M.Sc.) - Federal University of Parana.
- [12] **Fernandes, P. T. 2013.** Método de desenvolvimento integrado de produto orientado para a sustentabilidade. Dissertation (M.Sc.) Pontifical Catholic University of Parana.
- [13] **Kamsu-Foguema, B., Coudert, T., Béler, C., Geneste, L. 2008.** Knowledge Formalization in Experience Feedback Processes: An Ontology-Based Approach, *Computers & Industry*, 5(7), pp 694-710.
- [14] **Shapiro S.C. 2003.** Knowledge Representation. *Encyclopedia of Cognitive Science*, 2, pp 671-680.
- [15] **Benacerraf, P. and Putnam, H. 1984.** Philosophy of Mathematics Selected Readings. Cambridge University Press.
- [16] **Ben-Ari, M. 2012.** *Mathematical Logic for Computer Science*, 3rd Edition. Springer London Heidelberg.
- [17] **Baader, F.; McGuinness, D.L.; Nardi, D. and Patel-Schneider, P.F. 2003.** The description logic handbook: Theory, implementation, and applications. 3rd Edition, Cambridge University Press.
- [18] **Lezoche, M.; Yahia, E.; Aubry, A.; Panetto H. and M. Zdravković. 2012.** Conceptualising and structuring semantics in cooperative enterprise information systems models, *Computers & Industry*, 63(8), pp 775-787.
- [19] **Eriksson, M.; Börstler, J. and Borg, K. 2009.** Managing requirements specifications for product lines - An approach and industry case study, *Journal of Systems and Software*, 82(3), pp 435-447.
- [20] **Bryant, B.R. 2000.** Object-Oriented Natural Language Requirements Specification, Proceedings of Computer Science Conference, Canberra, pp 24-30.
- [21] **Chen, Z. Y.; Lin, S; Yao, J. Q.; Zeng, Y. and Eberlein, A. 2007.** Formalisation of product requirements: from natural language descriptions to formal specifications, *International Journal of Manufacturing Research*, 2(3), pp 362-387.
- [22] **Ferrari, R.N. and Madhavji, N.H. 2008.** Architecting-problems rooted in requirements, *Information and Software Technology*, 50(1/2), pp 53-66.
- [23] **Lin, H.K. and Harding, J.A. 2007.** A manufacturing system engineering ontology model on the semantic web for inter-enterprise collaboration, *Computers & Industry*, 58(5), pp 428-437.
- [24] **Nagahanumaiyah, K.S. and Ravi, B., 2008.** Computer aided rapid tooling process selection and manufacturability evaluation for injection mold development, *Computers & Industry*, 59(2/3), pp 262-276.
- [25] **Halpin, T. 2006.** Object-Role Modeling (ORM/NIAM), *International Handbooks on Information Systems Springer*, pp 81-103.
- [26] **Bollen, P. 2014.** Fact-Based Semantic Modeling in the Information and Behavioural Perspectives, *OTM 2014 Workshops, LNCS*, 8842, pp. 663-666.
- [27] **Pan, W. and Liu, D. 2010.** Mapping Object Role Modeling into Common Logic Interchange Format. 3rd International Conference on Advanced Computer Theory and Engineering (ICACTE), 2, pp 104-109.
- [28] **Jarrar, M. 2007.** Towards automated reasoning on ORM schemes mapping ORM into the DLridf description logic. Conceptual Modeling - ER2007 4801, pp 181-197.
- [29] **Nguyen, T. and Thanh, N. 2007.** Modeling ORM Schemas in Description Logics. *Complex Systems Concurrent Engineering*, pp 547-555.
- [30] **ISO (International Organization for Standardization) / IEC (International Electrotechnical Commission). 2007.** ISO/IEC 24707:2007. Information technology - Common Logic (CL): a framework for a family of logic-based languages, Geneva, CH.