Méthode de vérification des exigences pour l'ingénierie système à partir d'une représentation logique du système modélisé en RDF.

Albéric Cornière, Virginie Fortineau, Thomas Paviot, Samir Lamouri

Méthode de vérification des exigences pour l'ingénierie système à partir d'une représentation logique du système modélisé en RDF.

Albéric Cornière*†, Virinie Fortineau*, Thomas Paviot*, Samir Lamouri*
*ENSAM ParisTech LAMIH
151 blvd de l'hôpital
75013 PARIS, France
†alberic.corniere@ensam.eu

***Abstract*—L'ingénierie des exigences est souvent considérée comme le processus de départ du cycle de vie du produit, qui débuterait par l'expression des besoins des utilisateurs, et qui vise à en tirer un ensemble cohérent de spécifications.**

Les exigences portent par nature sur un système à concevoir, pour les définir il est donc nécessaire de disposer d'informations sur le système avant même sa conception. Il est également important de les capitaliser pour la gestion du cycle de vie étendu.

Cet article propose un cadre de modélisation d'une telle base de connaissance construite dans le paradigme RDF (Resource Description Framework)[1] autour des concepts qui définissent les exigences, et le système sur lesquelles elles portent.

I. Introduction

L'ingénierie des exigences est souvent considérée comme un processus amont du cycle de vie du produit qui a pour objectif la formalisation d'un ensemble cohérent de spécifications à partir de l'expression des besoins des utilisateurs. Le terme "ingénierie des exigences" a été introduit à la fin des années 70 par un rapport technique sur la sécurité informatique pour la défense des États-Unis [2]. Depuis les années 1990 elle est devenue une thématique de recherche, qui touche aux différentes activités de ce processus : formuler, modéliser, valider et vérifier l'ensemble des exigences s'appliquant sur le système, et gérer leur traçabilité [3]. En soi, l'ingénierie des exigences est donc intimement liée à la représentation sémantique du système lui-même ; et c'est au travers de la vérification des exigences que la vue "logique" de l'ingénierie système (IS) peut être confrontée à la vue "produit" conçue.

Dans une approche Product Lifecycle Management (PLM), l'ingénierie des exigences a cependant un périmètre plus large : il s'agit d'un processus itératif intervenant à chaque étape du cycle de vie, à chaque niveau de définition du système[4], et à chaque mise à jour du système, afin de spécifier, dimensionner, construire ou encore maintenir un système conforme aux besoins.

Xème Conférence Internationale : Conception et Production Intégrées, CPI 2015, 2-4 Décembre 2015, Tanger - Maroc.
Xth International Conference on Integrated Design and Production, CPI 2015, December 2-4, 2015, Tangier - Morocco.

Nous postulons dans ce travail que la formalisation et la vérification des exigences est ainsi au cœur de la démarche d'ingénierie système, et doit être intégrée à une approche PLM de gestion de l'information. De plus, dans le cas d'une application à des systèmes complexes et à longue durée de vie, comme le sont les centrales nucléaires, la vérification des exigences se doit d'être automatisée. Dès lors, leur formalisation doit reposer sur des éléments génériques dont la sémantique riche vient directement des objets métier de l'ingénierie système, c'est-à-dire à la représentation logique du système.

Une première proposition d'un modèle générique d'exigences en vue de leur vérification automatique a déjà été réalisée dans la contribution [5]. Les éléments de ce modèle générique eux-mêmes reposant sur des objets métier représentant la centrale [6].La problématique de ce travail est la modélisation des éléments de la vue logique du système et de leurs interactions sémantiques avec les exigences, en vue de la mise en œuvre effective de la conceptualisation proposée et de la vérification des exigences. L'enjeu est de permettre un atelier de modélisation métier des exigences pour l'ingénieur IS, qui permette leur vérification automatique par le système PLM.

Dans cette contribution, nous proposons une modélisation de la vue "logique" du système qui s'appuie sur une définition des concepts métier et des exigences sous la forme de triplets RDF (voir section III) : puisque "l'information est dans la relation" [7], le triplet (une relation entre deux concepts donnés) est l'élément fondamental du paradigme proposé. Le deuxième élément fondamental de la modélisation proposée est une formalisation en deux parties : d'une part, les objets sémantiques métier (ou "motifs") et d'autre part, les données réelles, appelées occurrences. La présentation d'un cas d'étude (section IV) permet d'illustrer l'imbrication entre les éléments d'exigence et les objets du système, et de montrer la démarche de vérification associée. Les avantages de la modélisation proposée sont finalement discutés en section V, tout comme les perspectives tant sémantiques que techniques.

II. État de l'art

A. La modélisation et la vérification des exigences dans une démarche d'IS

La majeure partie des contributions relatives à l'ingénierie des exigences (IE) relève du domaine logiciel ou des systèmes d'information, et/ou se concentre sur l'étape de formulation des exigences[8], [9]. Une revue complète des différentes méthodes d'élicitation des exigences a d'ailleurs été réalisé par [10]. Même dans le domaine des exigences produit ou des services, les travaux sur la vérification proposent des algorithmes spécifiques, pour des méthodes de vérification manuelles, formulées exigence par exigence, à partir d'énoncés généralement textuels.[11], [12]

Un autre pan de la littérature, en général regroupé sous le terme d'artifact-Based RE, s'intéresse à la classification des exigences en vue de leur traitement, proposant alors des modèle de gestion des exigences et de leur traçabilité. Cependant, ces travaux ne proposent pas une formalisation conceptuelle du contenu de l'exigence à partir d'objets métier, se contentant en général d'énoncés textuels comme c'est le cas dans la norme ISO15288 et dans le formalisme SysML.

[13] par exemple relève comme limite à la méthode RD-Mod l'absence de lien sémantique entre la liste des exigences (document) et l'architecture fonctionnelle du produit. Or, une formalisation rigoureuse et sémantique du contenu des exigences, liée aux arbres fonctionnels et organiques du produit et associée à un modèle générique, est une condition nécessaire à un traitement automatique de la conformité du produit vis à vis des exigences. Le présent travail ne discute donc pas de la formulation des exigences : elles sont les données d'entrée. Il ne contribue pas non plus aux processus de gestion des exigences en temps que modèles d'organisation (workflow).

Cette contribution présente une modélisation générique des exigences préalablement élicitées, qui permet grâce à un raisonnement sur la vue logique du produit de vérifier automatiquement la conformité du système à l'ensemble des exigences qui le contraint, et d'assurer la traçabilité des exigences tout au long du cycle de vie du produit. Pour y parvenir, la généricité du modèle d'exigence, la richesse sémantique de la représentation du système, et les liens (mappings) qui sont établis entre les deux sont des éléments cruciaux.

B. La représentation du système dans l'ingénierie système

L'expression des exigences repose sur une vue abstraite du système, représentant différents niveaux de granularité différents. Cette vue "logique" telle qu'elle est désignée dans l'ingénierie système complète alors la vue "produit" qui est une représentation fidèle du système réel. Par exemple, pour un ingénieur IS, une exigence peut porter sur l'ensemble des robinets du système d'alimentation, et une autre sur les robinets du barillet de ce système d'alimentation. Alors, les robinets du système d'alimentation et les robinets du barillet sont deux concepts différents dans la vue logique du système, bien que, *in fine*, le robinet physique Rob_{328} sera à la fois un robinet du barillet et un robinet du système d'alimentation : la vue produit n'aura qu'un objet, représenté par divers concepts de la vue logique. La modélisation des exigences doit donc reposer, avant toute chose, sur un modèle logique de la centrale.

Ce modèle logique contient dans une même vue une représentation multi-niveaux, abstraite et multi-vues d'un même système. En ce sens, c'est donc un modèle en réseau, interconnectant des taxonomies de systèmes, fonctions, matériels, etc. Il doit permettre également une richesse sémantique suffisante pour l'expression de l'ensemble des exigences, que les modèles produits existants, et particulièrement les modèles standards ne proposent pas [14], [15].

C. Contraintes liées à la complexité du système et à l'approche PLM

a) Traçabilité : La traçabilité des exigences est un point essentiel de la vérification ; particulièrement si la vérification intervient en vue d'une révision de la conception. Il s'agit là de la capacité à identifier les relations d'une exigence aux fonctions ou composants qu'elle contraint, et inversement, ainsi que l'identification des causes du résultat de la vérification, qu'il soit positif ou négatif.

Lors d'une vérification automatique, une traçabilité fiable dépend de de la disponibilité des informations pour le processus de vérification, autant que pour les ingénieurs systèmes. Une sémantique suffisamment riche permet d'identifier et analyser les sources d'une exigence, par la distinction des différents concepts et relations mis en œuvre dans sa définition.

b) Automatisation de la vérification : compte tenu de l'échelle et de la complexité des systèmes concernés, la vérification ne peut dépendre de la seule expertise humaine, ne serait-ce que par l'effort considérable qu'elle représente. Un processus de vérification automatique peut traiter une large part des exigences et permettre aux ingénieurs de concentrer leur effort sur les points sensibles.

c) Atomisation de la vérification : la complexité et l'échelle d'une centrale nucléaire peut induire un réseau logique fortement connexe et arbitrairement grand. Il faut donc réduire l'information pour la vérification au strict nécessaire pour ne pas ajouter à la complexité et pour analyser pour chaque vérification un ensemble réduit de données, afin de limiter la charge de calcul et obtenir des résultats dans un délai acceptable.

d) Fiabilité et justesse : la fiabilité du résultat et du processus est cruciale dans le contexte de l'ingénierie nucléaire. Certaines exigences portent sur la sécurité ou la sûreté, pour lesquelles la fiabilité du processus est essentielle.

e) Généricité du modèle : un modèle générique est choisi pour les exigences, de sorte que leur définition soit indépendante du projet concerné, et que l'algorithme de traitement soit le même pour toute exigence considéré, ce qui évite de recourir à des méthodes ou des algorithmes ad-hoc pour différentes applications.

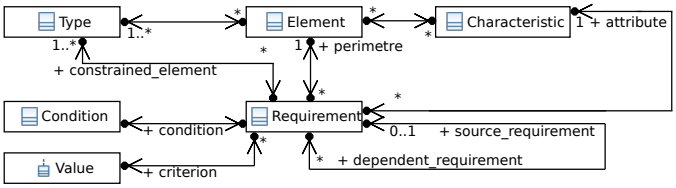

Fig. 1. classes liées aux exigences pour la vérification automatique

D. Représentation conceptuelle et générique d'une exigence pour sa vérification

Une conceptualisation générique des exigences en vue de leur vérification est présentée sur la figure 1. Elle a été proposée et discutée par [6]. Malgré la syntaxe UML utilisée, c'est uniquement une représentation des concepts mis en œuvre dans une exigence : ce n'est pas une modélisation implémentable des exigences. Cette conceptualisation repose sur 5 éléments génériques : d'abord les **conditions** circonstancielles, dans lesquelles l'exigence contraint l'**attribut** d'un **élément contraint**, qui peut être une fonction, un système ou encore un matériel. L'attribut doit, pour satisfaire à l'exigence, être conforme à un ensemble de valeurs admissibles, appelé **critère**. La vérification de l'exigence consiste alors à comparer les valeurs réelles de l'attribut de l'élément contraint au critère qui les restreint. Pour éviter toute ambiguïté dans la recherche de l'ensemble des éléments contraints d'une exigence, des informations contextuelles sont ajoutées : il s'agit du **périmètre** de l'exigence.

Par exemple, " *les vannes du systèmes de refroidissement doivent être manœuvrables localement*" est une exigence qui contraint l'**élément contraint** "*vannes*" à avoir l'**attribut** "*type de manoeuvre*" égal au **critère** "*local*". Cependant, il ne s'agit pas de toutes les vannes de la centrale, uniquement celles dans le **périmètre** du "*système de refroidissement*".

III. Modélisation des exigences à partir d'un réseau d'occurrences définies par des "motifs" sémantiques

f) Un réseau de donnée définissant des motifs : Les éléments de définition des exigences sont par nature applicables à des individus (ou occurrences) de la vue produit, en accord avec les concepts de la vue logique. Le jeu de données les définissant est une représentation multi-niveaux, et multi-vues. Elle consiste en un réseau dont les nœuds représentent les concepts, la taxonomie, les états du système, etc. ; les relations entre eux se rapportent aux vues métier du système. De la diversité des individus découle la richesse sémantique nécessaire pour mettre en correspondance les exigences avec les éléments du système qu'elles contraignent.

Les chaînes et sous-graphes relationnels de ce réseau sont considérés comme des "motifs" pour la définition des exigences : "les robinets du système de refroidissement" correspond par exemple au motif "*individu – de type – robinet*", où "*individu – est dans – système de refroidissement*"

Sur la figure 2, l'exigence 4 stipule que la fonction_1 doit avoir une *durée de mise en œuvre* inférieure à 1 heure. Le

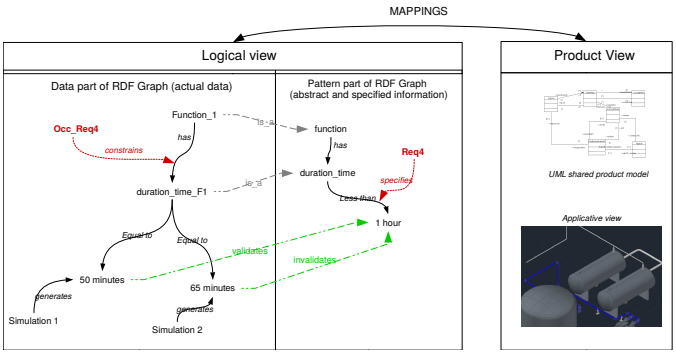

Fig. 2. Graphes jumeaux des données conceptuelles et factuelles relatives au système

schéma montre la vue logique RDF bipartite, avec un pan "données" et une partie pour l'information et les "motifs". Le graphe RDF est représenté en noir. Sur le côté gauche, la fonction_1 a sa propre durée de mise en œuvre, dans le pan de droite une fonction "peut avoir" une durée de mise en œuvre pour caractéristique. L'exigence_4 est représentée en rouge, et a deux parties : les données contraintes i.e. le triplet "*élément contraint – a – attribut*" et le motif de la spécification "*attribut – moins de – 1 heure*". Enfin, la vérification, en vert, est la comparaison de la valeur réelle du versant "données" avec la spécification. Dans cet exemple, deux valeurs distinctes issues de simulations différentes sont prises en compte, l'une satisfaisant à l'exigence, et l'autre non.

g) Syntaxe RDF : comme dit ci-avant, la la vue logique du système est représentée comme un réseau de relations entre individus. Le formalisme RDF[1] est tout-à-fait adapté pour déclarer des individus et leur relations en tant que triplets de la forme "$objet_1$ – $relation$ – $objet_2$", ce qui permet de construire un multi-graphe orienté aux relations identifiées.

Parallèlement, la possibilité offerte par RDF de séparer les données en deux *RDF triples store* autorise la séparation des données de la vue produit des motifs utilisés pour les traiter, tout en gardant un lien entre les *triples stores* –par des mises en correspondances elles aussi formalisées avec RDF.

h) Modélisation générique des exigences à partir d'un double réseau de triplets : Les exigences elles-mêmes peuvent être représentées en tant que triplets RDF : un premier triplet de la forme "*élément contraint* – a – *attribut*" et un second de la forme "*attribut* – est dans – *critère*" définissent le motif de la vérification proprement dite.

L'utilisation de RDF pour définir cette état attendu de la conception permet la distinction au sein du modèle de données entre la spécification et l'état réel du système. Leur cohabitation au sein d'un même système de données permet quand à lui de les traiter conjointement. En termes de niveau d'abstraction, les exigences lient en effet la spécification (une représentation conceptuelle) à la réalité de la solution constructive ou de l'implémentation (une occurrence spécifique du système).

i) Méthode de vérification automatique des exigences: des occurrences de vérification peuvent alors être dérivées

[1]Resource Description Framework

des définitions des exigences par triplets, en les faisant correspondre avec des vues partielles de la vue logique (i.e. des vues contenant les seules relations "a" d'un individu vers un attribut, et "est dans" d'un attribut vers un critère). Ces "vérificateurs" génériques consistent en une simple comparaison et des métadonnées de traçabilité. Un tel vérificateur est créé pour chaque occurrence de la vue produit qui corresponde au motif de définition de l'*élément contraint*. Leurs résultats individuels et agrégés déterminent la satisfaction de l'exigence (conceptuelle) dans son ensemble.

IV. Cas d'application

Le cas d'application est issu du domaine de l'ingénierie nucléaire qui se caractérise par des systèmes de grande complexité et de grande échelle. Les objets sémantiques nécessaires à la représentation d'une installation nucléaire se comptent en milliards, auxquels il faut encore ajouter les objets abstraits qui peuvent représenter des groupes de systèmes ou des phases de vie par exemple.

Ensuite, la complexité d'une installation nucléaire provient des interactions non triviales entre ses éléments, tant dans la réalité physique des processus mis en œuvre qu'au sein du modèle de données qui les représente. Par ces deux aspects l'ingénierie nucléaire est représentative des systèmes complexes de grande échelle.

Une des exigence de ce cas d'application (illustrée sur la figure 2) est : "Les fonctions de sûreté doivent être accomplies en moins d'une heure". Pour La modéliser, les éléments de définition présentés en II-D sont utilisés : elle contraint l'**attribut** "*temps de mise en œuvre*" de ses **éléments contraints** "*fonction de sûreté*" dans son **périmètre** : la "*centrale*" à être inférieure à "*une heure*", son **critère**.

Les triplets reflètent cette exigence dans le domaine conceptuel : Un premier triplet de concepts décrit l'élément contraint et l'attribut : "*fonction de sûreté* – a – *durée de mise en œuvre*" et un second la spécification : "*durée de mise en œuvre* – moins de – *une heure*". Dans le graphe des données, sitôt qu'un individu est référencé comme fonction de sûreté (au travers d'une relation '*est_un*' de cette occurrence vers le concept *fonction de sûreté*), une occurrence d'exigence est créé ainsi que le triplet "*Occ_ExigN – contraint – Fonction_1*".

Occ_ExigN est le point de départ de l'algorithme de vérification. Celui-ci renvoie une insatisfaction si *Fonction_1* n'a pas d'attribut *durée de mise en œuvre*, ou bien s'il est hors de la plage admissible. L'individu *ExigN* du graphe conceptuel recense les résultats des différents vérificateurs, et leurs occurrences associées, et considère l'exigence comme non satisfaite si l'un d'entre eux au moins reporte une insatisfaction. La vérification se fait individuellement sur les données de chaque individu contraint, assurant la traçabilité à ce niveau de granularité.

V. Discussion et perspectives

Les modèles et les méthodes proposées dans cette contribution admettent des limites. La première est qu'elles dépendent d'une représentation sémantique des données réelles pour la vérification. Cela peut poser le problème du transfert des données vers la vue logique, quand elles sont par exemple issues d'environnements de travail spécifiques comme certains outils de CAO. Ce problème d'interopérabilité a été présenté et discuté dans [16]. Le surmonter peut se faire par des mises en correspondance sémantiques entre la vue logique globale et les différentes vue produit. De telles mises en correspondance peuvent tirer parti de capacités de raisonnement telles que celles présentées dans les perspectives ci-après.

La taille et la richesse de la vue logique globale pose également une difficulté de passage à l'échelle : alors que les données concernées augmentent en nombre et en connexions, la complexité des mises en correspondance par motifs augmente exponentiellement, entraînant potentiellement un traitement arbitrairement long. Dans cette contribution, aucune supposition n'est faite quand à la structure du réseau logique ; qui pourrait avoir des propriétés utilisables pour optimiser le traitement du graphe d'occurrences lors de l'allocation des occurrences d'exigence. Le *périmètre* est une autre voie de mitigation du problème d'échelle. Quand à la vérification elle-même, elle peut se faire efficacement par lots, grâce à la généricité et à la faible empreinte du modèle de vérification [5].

Le modèle et les méthodes proposées avec lui sont basés sur un formalisme adapté à la conservation et à l'échange des données, ce qui permet d'envisager leur usage non seulement au cours du cycle de vie du produit, mais également d'un projet à l'autre au sein d'une faille de produits. L'enregistrement de modèles d'exigences dans une base de connaissances peut permettre de générer des exigences conceptuelles à partir des connaissances et des règles conceptuelles qu'elle contient – potentiellement par l'application de règles.

Le raisonnement direct sur les "motifs" et le graphe de la vue produit n'est pas trivial, non seulement à cause de sa taille, mais surtout à cause de sa complexité e de la variété des règles qui doivent s'y appliquer pour conserver sa cohérence. Par exemple la transitivité de la relation *système – a – sous-système* peut se faire par application de règles. En pratique, compléter le graphe d'une telle façon implique de nombreux ajouts au réseau. Les capacités de raisonnement apportées par des modèles ontologiques ainsi que l'utilisation de règles SWRL peuvent être utilisées pour expliciter cette part implicite de l'information, avant la génération de graphes RDF plus complets.

VI. Conclusion

Dans cette contribution, nous proposons une représentation sémantique de la connaissance associée à une conception de système et à ses données de définition. La première contribution de ce modèle est la représentation des données sous la forme d'un graphe orienté générique par construction, et construit avec des triplets RDF. La seconde contribution de ce modèle est al formalisation de ce graphe en deux volets conjoints qui reflètent la séparation de la spécification et des données système. Les modèles proposés peuvent fournir les données nécessaires à la vérification automatique des

exigences, et à la définition de modèles d'exigences et de motifs de typage pour les éléments du système.

REFERENCES

[1] W. W. W. Consortium, "The Resource Description Framework," W3C, http://www.w3.org/standards/techs/rdf. [Online]. Available: http://www.w3.org/standards/techs/rdf

[2] M. Alfor and J. Lawson, "Software requirements engineering methodology (development)," TRW Defense and Space Systems Group, Tech. Rep., 1979.

[3] P. Zave and M. Jackson, "Four dark corners of requirements engineering," *ACM Transactions on Software Engineering and Methodology (TOSEM)*, vol. 6, no. 1, pp. 1–30, 1997.

[4] S. Arnold, "Iso 15288 systems engineering—system life cycle processes," *International Standards Organisation*, 2002.

[5] A. Corniere, V. Fortineau, T. Paviot, and S. Lamouri, "Towards a framework for integration of requirements engineering in PLM," 2015.

[6] A. Cornière, V. Fortineau, T. Paviot, S. Lamouri, J.-L. Goblet, A. Platon, and C. Dutertre, "Modelling requirements in service to plm for long lived products in the nuclear field," in *Advances in Production Management Systems. Innovative and Knowledge-Based Production Management in a Global-Local World*. Springer, 2014, pp. 650–657.

[7] S. Tsuchiya, "Improving knowledge creation ability trough organizational learning," *Proceedings of International Symposium on the Management of Industrial and Corporate Knowledge, ISMICK, Compiègne, France*, 1993.

[8] M. Rahman, S. Ripon *et al.*, "Elicitation and modeling non-functional requirements-a pos case study," *arXiv preprint arXiv:1403.1936*, 2014. [Online]. Available: http://arxiv.org/pdf/1403.1936

[9] N. Ahmed and R. Matulevicius, "A method for eliciting security requirements from the business process models," in *CAiSE Forum and Doctoral Consortium*, 2014, pp. 57–64. [Online]. Available: http://ceur-ws.org/Vol-1164/PaperVision08.pdf

[10] S. Nisar, M. Nawaz, and M. Sirshar, "Review analysis on requirement elicitation and its issues," *International Journal of Computer and Communication System Engineering (IJCCSE)*, vol. Vol. 2, pp. 484–489, 2015.

[11] S. Ben-David, B. Sterin, J. M. Atlee, and S. Beidu, "Symbolic model checking of product-line requirements using sat-based methods," in *Software Engineering (ICSE), 2015 IEEE/ACM 37th IEEE International Conference on*, vol. 1. IEEE, 2015, pp. 189–199.

[12] W. Viriyasitavat and L. Da Xu, "Compliance checking for requirement-oriented service workflow interoperations," *Industrial Informatics, IEEE Transactions on*, vol. 10, no. 2, pp. 1469–1477, 2014.

[13] M. Berkovich, J. M. Leimeister, A. Hoffmann, and H. Krcmar, "A requirements data model for product service systems," *Requirements Engineering*, vol. 19, no. 2, pp. 161–186, 2014.

[14] ISO, "Iso 10303."

[15] OMG, "Sysml v 1.3," Jun. 2012, http://www.omg.org/spec/SysML/1.3.

[16] T. Paviot, "Méthodologie de résolution des problèmes d'interopérabilité dans le domaine du product lifecycle management," Ph.D. dissertation, Ecole Centrale Paris, 2010.