Une sémantique pour les patrons de justification

Clément Duffau, Thomas Polacsek, Mireille Blay-Fornarino

Une sémantique pour les patrons de justification

Clément Duffau, Thomas Polacsek, Mireille Blay-Fornarino

AXONIC, Sophia Antipolis, France
I3S, Université Côte d'Azur, CNRS, Sophia Antipolis, France
ONERA, 2 avenue Edouard Belin BP74025, 31055 TOULOUSE Cedex 4, France

RÉSUMÉ. *La création d'un produit, que cela soit un objet matériel ou un service, s'accompagne de la production de justifications qui peuvent être, suivant les cas, des éléments de conformité dans le cadre de la qualité, des documents de traçabilité, des rapports d'expérimentations, des rapports d'experts, etc. Dans des contextes critiques, comme le médical, le ferroviaire ou l'aéronautique, il est obligatoire de convaincre une autorité certificatrice, que le développement d'un produit a été correctement réalisé. Cette obligation entraîne une inflation des documents justificatifs, inflation qui rend la lecture et la compréhension de cet ensemble de justifications difficile. Pour structurer ces justifications, il peut être utile d'utiliser des diagrammes de justification. Cependant, ces diagrammes, bien qu'utiles, ne sont qu'une notation graphique informelle. Dans cet article, nous définissons une sémantique formelle du diagramme de justification et nous donnons les premières pistes de ce que pourrait être un logiciel d'aide à la conception de tels diagrammes.*

ABSTRACT. *The creation of a product, whether it is an object or a service, is accompanied by the production of justifications which may be, depending on the case, elements of conformity in the context of quality, traceability documents, experimental reports, expert reports, etc. In critical contexts, such as medical, railway or aeronautics, it is mandatory to convince a certifying authority that the development of a product has been carried out correctly. This obligation leads to an inflation of justification documents, which makes it difficult to read and to understand this set of justifications. To structure these justifications, it may be useful to use Justification Diagrams. However, these diagrams, while useful, are only an informal graphical notation. In this article, we define a formal semantics of the Justification Diagram and we give the first hints of what could be a software to support the design of such diagrams.*

MOTS-CLÉS : *Justification, Certification, Ingénierie des exigences, Exigences de qualité, Ingénierie dirigée par les modèles*

KEYWORDS: *Justification, Certification, Requirements engineering, Quality requirements, Model-driven engineering*

e soumission à *Inforsid 2018*, le April 18, 2018.

1. Introduction

Dans de nombreux domaines où il existe des risques pour l'homme, comme la médecine, le nucléaire ou l'avionique, il est nécessaire de passer par une phase de certification visant à garantir le bon fonctionnement d'un système ou d'un produit. Cette certification est délivrée par une autorité, qui est généralement une tierce partie, mais pas toujours. Parfois une entreprise peut avoir l'autorité d'effectuer elle-même les activités de certification. Il est important de comprendre que les activités de certification ne se concentrent pas seulement sur le produit final, mais concernent tous les aspects du processus de production. Dans la pratique, la certification se fait en fonction de documents normatifs qui expriment les exigences auxquelles le produit et le processus de développement doivent se conformer. Par exemple, la norme applicable aux instruments médicaux (ISO 13485) décrit le processus de haut niveau qui doit être suivi à chaque étape de développement. Dans le cadre de cette norme, un audit de certification consiste, pour l'industriel, à produire une documentation selon laquelle le processus suivi est conforme à la norme. Ces documents peuvent être des explications logicielles, des résultats d'essais ou des protocoles d'essais cliniques suivis pendant les expériences. Dans les systèmes logiciels avioniques, l'autorité de certification suit le processus de développement de bout en bout et elle doit être convaincue que le processus et le produit sont conformes à la directive DO 178 et à l'ARP4754. Ainsi, une grande partie des activités d'accréditation est liée à la production de documents justificatifs, à une argumentation pour convaincre une autorité.

Pour répondre à cette nécessité de justification, il peut-être tentant de considérer comme éléments de justification tous les documents produits, peu importe leur but premier. Dès lors, une partie du travail de certification consiste à appréhender cette masse de justifications, qui est non structurée, parfois sans liens clairs avec les exigences du produit final et sans liens entre les documents eux-mêmes. Pour faire face à ce tsunami documentaire, Polacsek(Polacsek, 2016) propose un nouveau type de diagramme, le diagramme de justification, qui vise à structurer une argumentation de certification. Le but de tels diagrammes est de donner à l'autorité de certification une vue de l'ensemble de la justification explicitant clairement les liens de raisonnement entre les différents éléments et les arguments avancés pour prouver la conformité du produit avec les exigences propres à la certification. De plus, le diagramme de justification peut permettre d'exprimer, dans une vue synthétique, la liste des éléments de preuves et des moyens de conformité que doivent fournir les équipes de développement pour être conformes à une norme de certification.

Dans la pratique, les diagrammes de justification sont utilisés à plusieurs niveaux. Ils servent certes à organiser les documents de justification, mais ils servent aussi à représenter des patrons d'argumentation (Duffau *et al.*, 2017a), (Polacsek, 2017). Ces patrons consistent en des diagrammes génériques qui listent, pour une solution donnée, les éléments de preuves et la conclusion attendue. Les patrons d'argumentation sont ainsi conçus par des experts du domaine, dans la même veine que les patrons de conception en ingénierie des modèles (Alexander *et al.*, 1977), (Gamma *et al.*, 1995). Ici, pour chaque patron de justification, les experts définissent, pour un ob-

jectif donné, quelles sont les preuves nécessaires et quelles sont les restrictions et les conditions d'utilisation dans la mise en œuvre d'un moyen de conformité.

De plus, un même élément de justification présent dans plusieurs patrons, peut être "*instancié*" avec plus ou moins de détail. Par exemple, dans le domaine médical, l'élément de justification que sont les documents d'analyse des risques servira à justifier la conformité aux normes de gestion des risques, mais servira aussi, dans une version beaucoup plus détaillée et spécifique, à justifier la gestion des risques dans toutes les parties techniques du projet et leurs normes associées. Dès lors, la distinction entre ces différents niveaux d'abstraction et les différents points de vue sur les justifications doit être soigneusement pris en compte.

Aujourd'hui, les diagrammes de justification ne posent pas clairement la distinction entre patron et réalisation de patron et le lien entre un élément générique et ses réalisations, instanciations, n'existe pas. Il apparaît donc comme nécessaire d'exprimer clairement, et sans ambiguïté, ces différents concepts. De plus, disposer d'une telle sémantique, permettrait de disposer d'outils de gestion de diagrammes de justification et de patrons qui s'appuient sur cette sémantique.

Le but de cet article est donc de définir une sémantique formelle pour les diagrammes de justification et plus précisément sur la relation de raffinement entre les patrons et les diagrammes finaux. Par ailleurs, nous présenterons une première implémentation d'un prototype de gestion de patrons de justification pour une aide à la certification, basée sur cette sémantique formelle. Dans cet article, en Section 2, nous présentons les concepts autour des diagrammes de justification ainsi que des travaux connexes autour des modèles de justification. En Section 3, nous introduisons la sémantique formelle pour les diagrammes de justification, sémantique qui nous permettra, en Section 4, de poser les pistes d'un outil pour l'aide à la création de diagrammes de justification. Pour finir, nous présentons les conclusions et perspectives de nos travaux en Section 5.

2. Justifier pour certifier

2.1. *Diagramme de Justification*

Les diagrammes de justification sont des diagrammes conceptuels permettant d'exprimer comment à partir d'un ensemble de faits on peut déduire une conclusion. Nous ne sommes pas ici dans le monde de la logique formelle, mais dans celui de l'explication. En effet, l'usage des diagrammes de justification est motivé par l'impossibilité d'être dans un monde formel. Ils capturent de bonnes pratiques, un pas de raisonnement accepté entre faits et conséquences, il relève d'une *"bonne rhétorique"* comme qualifiée par (Perelman et Olbrechts-Tyteca, 1958). Dérivée des travaux de l'argumentation légale de (Toulmin, 2003), leur représentation graphique s'inspire de la Goal Structured Notation (Kelly et Weaver, 2004), voir Figure 1. Les principaux concepts des diagrammes de justification sont :

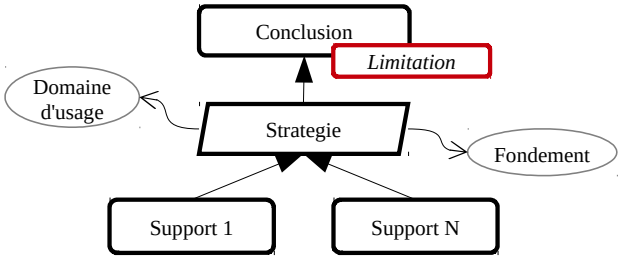

Figure 1. *Notation graphique d'un pas de justification inspirée de GSN*

– *Conclusion* explicite ce qui est soutenu pas la justification. C'est la conclusion du raisonnement, ce que nous cherchons à justifier.

– *Support* est un fait, une donnée, une hypothèse, etc. Ce sont les élément sur lesquels s'appuie la justification de la conclusion. Un support peut également être une *sous-conclusion* i.e. une conclusion d'une autre justification qui vient ici contribuer en tant que support.

– *Strategie* correspond à la méthode utilisée pour établir la connexion entre les supports et la conclusion.

– *Limitation* est une restriction à la conclusion. Elles sont séparées de la conclusion car elles n'ont vocation qu'à disparaître soit dans une justification de plus haut niveau, soit plus tard si les diagrammes s'inscrivent dans un processus temporel.

– *Fondement* est une explication de pourquoi une Stratégie est applicable. Si une stratégie consiste à suivre un processus défini par une norme, alors la Stratégie est l'application de la norme et le Fondement est l'explication de pourquoi la norme est applicable.

– *Domaine d'usage* donne les conditions précises d'usage et les limitations de la Stratégie. Si nous prenons l'exemple de l'application d'une norme, alors le domaine d'usage décrira dans quel contexte et pour quel besoin cette norme est applicable.

Même si, dans le cadre de la certification, l'usage de diagrammes de justification est prometteur, il faut tout de même souligner que, pour le moment, ce n'est qu'un système de notation. S'ils permettent de clarifier les différents concepts nécessaires à une justification, ils ne sont associés à aucun outil informatique et, par conséquent, toutes les opérations de vérification doivent être réalisées par un être humain. Avoir une définition formelle de propriétés souhaitables permettrait de disposer d'un outil automatique d'aide à la création et à la relecture de diagrammes de justification.

De plus, les différentes expérimentations d'utilisation, que cela soit dans le logiciel embarqué comme dans le médical, ont fait apparaître le besoin de patrons de justification. Cependant, les liens entre patron et réalisation de ces patrons, ainsi que les liens

possibles entre les patrons ne sont pas explicités et sont laissés à interprétation. Il est donc nécessaire de clarifier ces différents liens.

2.2. *Travaux connexes*

2.2.1. *Un besoin d'explication*

Le problème de fournir une explication pour convaincre qu'un résultat est bien fondé ne se pose pas uniquement dans le contexte de la certification. Au début des années quatre-vingt-dix, avec la montée des systèmes intelligents tels que les systèmes experts, les systèmes d'aide à la décision et les systèmes de prévision, de nombreux travaux ont insisté sur la nécessité de fournir une explication pour qu'un humain accepte un résultat automatique (Ass, 1992) (Int, 1993).

Dans ce domaine, des travaux comme ceux de (Chandrasekaran *et al.*, 1989) et (Southwick, 1991) soulignent le fait que les explications fournies par un système intelligent peuvent être classées en trois catégories. Tout d'abord, une trace des informations, les règles, ainsi que les étapes qui ont permis au système de produire le résultat correspondent à des explications, appelées *trace explications*. Deuxièmement, les *explications stratégiques* expliquent pourquoi une information est utilisée avant une autre, comment les différentes étapes du raisonnement sont choisies et comment elles contribuent à atteindre les objectifs principaux. Les explications stratégiques fournissent une explication du fonctionnement général du système. Troisièmement, les *explications profondes* justifient les fondements du système. Ces explications fournissent les théories à partir desquelles le résultat a été généré, les logiques sous-jacentes et les justifications de la base de connaissances.

Dans le cadre de la certification, nous sommes typiquement dans les explications profondes et, partiellement, dans les explications stratégiques. Ce qui est conforme avec (Ye et Johnson, 1995) qui explique que ce sont les explications profondes qui induisent la meilleure acceptation du système par des utilisateurs et la certification vise justement à convaincre une autorité. Notons ici qu'il existe des liens forts entre explication, et plus précisément explication profonde, et le schéma de Toulmin, comme le montre (Ye, 1995), schéma de Toulmin qui est le fondement des diagrammes de justification comme montré en Figure 1.

2.2.2. *Des modèles de justification*

Un des standards de modélisation pour la justification, appelé SACM (OMG, 2013), est un norme portée par l'OMG qui cherche à établir le méta-modèle de différents modèles d'ingénierie des exigences pour la qualité : les cas d'assurance structurés. Un cas d'assurance structuré est un argument structuré, supporté par un ensemble d'évidences, permettant de donner un cadre compréhensif et valide de sûreté et d'applicabilité d'un système dans un environnement donné (Weinstock et Goodenough, 2009). Différentes modélisations concrètes co-existent comme Goal-oriented Structured Notation (GSN) (Kelly et Weaver, 2004) ou Claim-Argument-Evidence

(CAE) (Emmet et Cleland, 2002). Ces deux modèles visent à expliciter la satisfaction de propriétés sur la sûreté d'un système. GSN et CAE bien que basés sur le modèle de Toulmin abordent celui-ci en rendant optionnel l'expression de la stratégie qui explicite le passage des supports de l'argumentation à la conclusion. Une partie du raisonnement est ainsi perdue. De plus, il est difficile d'utiliser ces modèles dans des cadres différents des propriétés de sûreté. C'est pour ces raisons qu'il est apparu crucial pour la communauté de pouvoir monter d'un niveau d'abstraction supplémentaire pour capturer un cadre générique pour les cas d'assurance structurés à travers SACM. Le méta-modèle proposé par SACM repose sur 3 aspects distincts : le modèle argumentaire, le modèle d'artefacts et le modèle de cas d'assurance. Le modèle argumentaire définit une logique d'assertions permettant de représenter des arguments liés entre eux par des liens typés permettant ainsi d'établir des inférences entre arguments comme l'ajout d'un contexte applicatif à un argument. Le modèle d'artefacts permet de formaliser des données factuelles représentées comme des artefacts (évènement, participant, technique, ressource, etc). Ces artefacts sont les éléments basiques sur lesquels des raisonnements peuvent être faits et ainsi mener à une argumentation. Pour finir le modèle de cas d'assurance permet de décorer, enrichir le modèle argumentaire en ajoutant des notes, des contraintes, la gestion de la langue. Ce canevas permet ainsi de définir toutes les couches pour un cadriciel d'exigences pour la qualité, des faits jusqu'au raisonnement menant à une revendication en passant par la couche de présentation de ce raisonnement. Ce meta-modèle a été utilisé dans diverses approches par la communauté pour en définir un langage utilisable à travers Eclipse pour editer et visualiser des diagrammes basés sur GSN (Matsuno, 2014) ou concevoir des exigences pour la certification (de la Vara et Panesar-Walawege, 2013).

Dans le domaine de l'ingénierie des exigences, i* est un langage qui propose de capturer les besoins très en amont depuis l'intérêt des parties prenantes jusqu'aux différentes variantes possibles du système à réaliser (Yu, 1997). Dans la nouvelle version i* (Dalpiaz *et al.*, 2016), un des changements notables est le remplacement des *"softgoal"*, qui permettaient de capturer des exigences non fonctionnelles, par le concept de *"qualité"*. Cette évolution ouvre la porte à une utilisation de i* dans le domaine des exigences qualité. Néanmoins, le langage reste très orienté processus, puisqu'il manipule principalement les notions d'acteurs, de ressources et de tâches. I* permet d'exprimer ce qui contribue ou empêche l'atteinte d'une qualité, mais ne permet pas de justifier cette contribution en s'appuyant sur du raisonnement et des artefacts de qualité. Cependant, pour palier cette limitation, (Van Zee *et al.*, 2016) proposent d'attacher de l'argumentation pour expliciter le raisonnement d'un diagramme i*. Plus précisément, il propose de rajouter une représentation des arguments et contre-arguments proposés par les différentes parties prenantes qui se sont prononcées pour, ou contre, un élément du modèle exprimé en i*.

Par rapport à ces travaux, les diagrammes de justification se situent a posteriori, il n'y a plus de pour et de contre, mais une explication rationnelle du choix final en vue de le certifier ou a minima de le justifier. Si nous prenons pour exemple la réservation d'un déplacement dans un cadre professionnel, dans un diagramme i*, nous trouverons des éléments relatifs aux tâches (acheter des billets, réserver une cham-

bre), des ressources (budget), des acteurs (agences de voyage), des objectifs (billets réservés) et des qualités (réservation rapide). Un diagramme de justification va lui se focaliser sur la preuve que des objectifs ont été atteints, par exemple, que les billets d'avion et la réservation de l'hôtel correspondent, ou que le déplacement est accepté (validé par le supérieur hiérarchique, conforme au budget, etc.).

3. Une sémantique formelle pour les diagrammes de justification

Dans cette section, nous allons présenter une sémantique formelle pour les diagrammes de justification. Nous nous sommes concentrés sur les éléments clés du schéma de justification et nous ne détaillons pas ici la *Limitation*, le *Domaine d'usage* et le *Fondement*. Concernant le *Domaine d'usage* et le *Fondement*, ils peuvent être vus comme un ajout à la *Stratégie* et donc être contenus, embarqués, dans la représentation que nous allons en donner. Concernant la *Limitation*, il est aussi possible de la voir comme une commodité d'écriture, du sucre syntaxique, et pas comme un élément constitutif du langage.

3.1. *Concepts de base*

Les diagrammes de justification manipulent des assertions, des allégations, du type "*le système résiste à une panne*" ou "*les tests sont tous positifs*". Ces assertions correspondent aussi bien à la conclusion, une sous-conclusion, qu'à un élément de preuve. Elles regroupent donc les supports présentés en Section 2. Elles peuvent être exprimées avec une simple phrase en langage naturel ou être exprimées en langage plus formel, comme la logique. Une assertion peut même être un élément plus complexe, comme l'intégralité d'un rapport ou la sortie numérique d'un logiciel. Nous noterons l'ensemble des assertions $\mathcal{A}$.

Afin de pouvoir comparer certaines assertions entres elles, nous définissons une relation, notée $\mathcal{R}$, que nous qualifierons de relation de conformité. L'idée sous-jacente à cette notion de conformité est celle d'un "*raffinement*". Ainsi, si deux assertions sont des formules logiques, $\mathcal{R}$ pourra être vu comme "*est un modèle de*", si ce sont des classes comme "*est une spécialisation de*" et si nous sommes dans l'utilisation du langage naturel, alors la signification de la relation de conformité $\mathcal{R}$ devra être donnée.

Définition 1 (Relation de conformité)
Soit $\mathcal{A}$ l'ensemble des assertions, $\mathcal{R}$ est une relation de conformité ssi $\forall a_1, a_2, a_3 \in \mathcal{A}$:

- $a_1 \mathcal{R} a_1$
- *if* $a_1 \mathcal{R} a_2$ *and* $a_2 \mathcal{R} a_1$ *then* $a_1 = a_2$
- *if* $a_1 \mathcal{R} a_2$ *and* $a_2 \mathcal{R} a_3$ *then* $a_1 \mathcal{R} a_3$

Notons qu'une relation de conformité définie un ordre partiel sur l'ensemble des assertions, elle est réflexive, transitive et antisymétrique. D'après cette définition,

plusieurs assertions peuvent être conformes à une même assertion. Prenons pour exemple deux rapports de tests différents, rt_1 et rt_2, qui sont conformes à une norme définissant les rapports de tests nt : $rt_1 \mathcal{R} nt$ et $rt_2 \mathcal{R} nt$. De façon duale, une assertion peut aussi être conforme à plusieurs assertions. Pour exemple, la parole d'un expert, notée a_e, peut être conforme à l'assertion de très haut niveau "*jugement d'expert*", a_{je}, et être aussi conforme à l'assertion "*hypothèse de travail*", a_h : $a_e \mathcal{R} a_{je}$ et $a_e \mathcal{R} a_h$.

Nous définissons un type d'assertions particulières, les assertions terminales. Nous dirons qu'une assertion est un terminal si aucune assertion n'est conforme avec elle. Généralement, un terminal est un fait concret comme un résultat bibliographique, une bonne pratique établie, un point de la spécification ou le résultat de test.

Définition 2 (Terminal)
$a \in \mathcal{A}$ *est un terminal ssi* $\forall a' \in \mathcal{A}, a' \mathcal{R} a \rightarrow (a' = a)$.

Comme chez Toulmin, la strategie est la pierre angulaire des diagrammes de justification. C'est elle qui explicite, comment, à partir d'éléments de preuve, il est possible de déduire une conclusion. Cette déduction s'appuie sur un domaine d'usage et des fondements, mais n'est pas de l'ordre d'une déduction formelle, sinon il serait inutile d'utiliser les diagrammes de justification. Dès lors, nous avons choisi d'encapsuler tous les éléments de cette déduction dans un concept *Stratégie*. Nous notons l'ensemble des stratégies $\mathcal{S}$.

Pour finir, nous allons définir un pas de justification (*pdj*), en d'autres termes donner une sémantique à la notation graphique présentée Figure 1. Nous noterons l'ensemble des *pdj* : $\mathcal{P}$.

Définition 3 (Pas de justification (*pdj*))
Un pdj *(pas de justification)* p *est un tuple* $\langle supports, strategie, conclusion \rangle$ *où :*

– $supports$ *est un ensemble d'assertions* $\subset \mathcal{A}$*,*
– $strategie \in \mathcal{S}$*;*
– $conclusion \in \mathcal{A}$

3.2. *Patron*

A l'aide de la relation de conformité $\mathcal{R}$, il est possible d'étendre le concept de conformité aux pas de justification. Nous donnons ci-après une définition de la conformité entre *pdj* .

Définition 4 (Conformité *pdj*)
Un pdj $s = \langle supp, strat, c \rangle$ *est dit conforme à un* pdj $s' = \langle supp', strat', c' \rangle$ *ssi :*

– $\forall a \in supp, \exists a' \in supp'$*,* $a \mathcal{R} a'$*,*
– $\forall a' \in supp', \exists a \in supp$*,* $a \mathcal{R} a'$*,*
– $\forall a, b \in supp, \forall a' \in supp'$*, si* $a \mathcal{R} a'$ *et* $b \mathcal{R} a'$ *alors* $a = b$*,*

– $strat = strat'$,

– $c\mathcal{R}c'$.

La définition que nous donnons ici privilégie la préservation des concepts encapsulés dans les supports d'un pas de justification plutôt que la cardinalité. En effet, si nous considérons les *pdj* $p_1 =< \{s\}, w, c >$ et $p_2 =< \{s_1, s_2\}, w, c >$, et $s\mathcal{R}s_1$ et $s\mathcal{R}s_2$ d'après notre définition p_1 est conforme à p_2.

Prenons pour exemple un pas de justification pdj_1 qui permet de faire une déduction sur une simulation numérique avec les deux supports suivants : "*la température est inférieure à 90 degrés*" et "*la température est supérieure à 0 degrés*". Considérons maintenant un autre pas de justification pdj_2 avec la même stratégie et la même conclusion, mais un seul support qui est un document validé attestant que la température est bien comprise entre 10 et 50 degrés. Si l'on considère ce support comme une spécialisation des deux autres, comme relié par R aux deux autres, alors ce *pdj* est bien conforme au premier. Le pas de justification pdj_3 qui prend en support un document d_1 qui atteste que la température est comprise entre 10 et 25 pourra alors être également conforme à pdj_2 et pdj_1.

Si nous voulons prendre en compte la cardinalité, c'est-à-dire ne considérer comme conforme entre eux que des *pdj* qui ont le même nombre de supports, nous devons considérer que la relation $\mathcal{R}$ est, pour l'ensemble des supports des deux *pdj* , une relation bijective, en d'autres termes, rajouter la condition suivante : $\forall a \in supp, \forall a', b' \in supp'$, si $a\mathcal{R}a'$ et $a\mathcal{R}b'$ alors $a' = b'$.

Notons que la relation de conformité entre les *pdj* , quelle que soit la définition choisie, est, elle aussi, une relation d'ordre partiel. La démonstration est triviale puisque cette nouvelle conformité est simplement basée sur la relation $\mathcal{R}$ entre les assertions. Par commodité nous noterons $s'\mathcal{R}s$, le fait que le *pdjs*$'$ est conforme au *pdjs*.

Propriété 1 *La relation de conformité entre pas de justification est réflexive, transitive et antisymétrique.*

Maintenant que nous disposons d'une relation entre les *pdj*, il nous est possible de définir formellement ce que sont un patron de justification, un raffinement de patron et une concrétisation de patron.

Définition 5

– $\forall s \in \mathcal{P}$, *s est un patron de justification ssi* $\exists s' \in \mathcal{P}, s'\mathcal{R}s$ *et* $s \neq s'$.

– $\forall s, s' \in \mathcal{P}$, *s raffine* s' *ssi* $s\mathcal{R}s'$.

– $\forall s \in \mathcal{P}$, *s est une concrétisation de patron de justification ssi* $\forall s' \in \mathcal{A}, s'\mathcal{R}s \rightarrow (s' = s)$.

Relativement à l'exemple donné précédemment : pd_1 et pd_2 sont des patrons. pd_2 raffine pd_1. pd_3 peut être appréhendé comme une concrétisation si aucun support ne peut raffiner le document d_1.

4. Vers un outil d'aide à la certification

4.1. *Utiliser des patrons d'experts*

Notre idée est de pouvoir disposer d'un ensemble de patrons de justification dédiés à un domaine. Cet ensemble de patrons forme une bibliothèque dans laquelle il est possible de venir piocher pour pouvoir guider et structurer la justification de conformité, dans le cadre de la réalisation d'un produit. Les patrons peuvent donc être vu comme des guides, listant les éléments nécessaires à la justification d'un objectif de certification.

La réalisation de cette bibliothèque ne peut être faite que sur la base d'experts qui vont définir les patrons de justification en fonction de leur connaissances, de bonnes pratiques établies, de normes, d'exigences de qualité, etc.

Dès lors, à la réalisation du produit, construire une justification pour un objectif fixé consiste à construire un diagramme de justification composé de *pdj* terminaux conformes aux patrons de la bibliothèque.

Notre but est donc de faciliter la construction des diagrammes de justifications. Cette construction peut être manuelle et/ou programmative en fonction du domaine ciblé, de l'environnement de développement et de la justification elle-même. Ainsi lorsque les justifications portent sur des tâches qui échappent au contrôle du système informatique, il appartient aux experts de construire les pas de justification. Inversement, dans un contexte de justification d'un processus pris en charge par le système informatique, il doit être possible de construire les pas de justification de manière automatique. En utilisant les patrons comme un guide, les outils doivent alors faciliter la construction incrémentale des pas de justification, permettre l'identification des assertions ou *pdj* manquants, forcer la vérification des conformités.

4.2. *La relation $\mathcal{R}$ dans la pratique*

La relation de conformité $\mathcal{R}$ est un élément clé de notre sémantique formelle. C'est pour pouvoir rester générique et accepter tout type d'assertion, que nous avons choisi de ne pas caractériser son comportement. Cependant, dans le cadre d'un logiciel, il nous faut faire des choix d'implémentation de $\mathcal{R}$. Comme nous l'avons précédemment évoqué, la relation $\mathcal{R}$ peut être vue comme un raffinement, une spécialisation, etc. Nous avons choisi de représenter ce lien au travers d'une procédure de décision qui permet de savoir si deux assertions sont reliées ensemble. De façon pratique, cela consiste en une méthode qui implémente la sémantique de $\mathcal{R}$. Cette sémantique est forcément dépendante du format choisi pour représenter les assertions. Ainsi, si les assertions sont représentées en logique, il est possible de considérer la conformité comme : "*est conséquence logique*", si les assertions sont dans un langage contrôlé (boilerplate), la conformité peut être une relation d'instanciation et dans le cadre d'assertion en langue naturelle, la conformité sera un algorithme réalisant du traitement automatique de la langue. C'est donc pour rester aussi générique que possible que nous avons décidé d'encapsuler la sémantique de $\mathcal{R}$ dans une méthode : $isConform$.

4.3. *Une aide automatique*

Le but de notre démarche est de proposer un outil d'assistance à la création de diagrammes de justification. Nous allons présenter ici un ensemble de fonctionnalités que devrait fournir un tel outil et, pour chaque fonctionnalité, exprimer ce qu'elle fait à l'aide de notre sémantique formelle.

4.3.1. *Comment justifier une conclusion ?*

Considérons une assertion, l'outil renvoie tous les patrons qui permettent de justifier cette assertion. D'un point de vue de la sémantique cela revient, pour une assertion $goal$ donnée, à retourner tous les $pdj \in \mathcal{P}$ avec $pdj =< S, w, c >$ tel que : $goal \mathcal{R} c$.

4.3.2. *Que peut-on justifier ?*

A partir d'un ensemble d'assertions, il s'agit de déterminer tous les *pdj* qui peuvent être appliqués. En d'autres termes, pour un ensemble d'assertions A, il faut trouver tous les $pdj \in \mathcal{P}$ avec $pdj =< S, w, c >$ tel que : $\forall s \in S, \exists a \in A, a \mathcal{R} s$.

4.3.3. *Le diagramme est-il constitué de concrétisations de patron ?*

Il peut être intéressant de considérer qu'un diagramme de justification a été construit sans assistance automatique. Dès lors, un outil automatique devrait pouvoir vérifier que (1) chaque pas de justification du diagramme est conforme à un patron et (2) que tous les pas de justifications du diagramme ne sont pas des patrons mais bien des concrétisations de patron. Sémantiquement, cela correspond à vérifier que pour tout pdj du diagramme:

(1) $\exists p \in \mathcal{P}, pdj \mathcal{R} p$ et $pdj \neq p$;

(2) pdj est une concrétisation de patron.

4.3.4. *Quels sont les pas de justification non conformes à des patrons?*

Inversement, il peut être intéressant de déterminer quels sont les *pdj* qui ne sont pas conformes à des patrons. Sémantiquement, cela correspond à déterminer quels sont les pdj du diagramme tels que $\nexists p \in \mathcal{P}, pdj \mathcal{R} p$.
L'ensemble des *pdj* non conformes à des patrons nécessite une étude approfondie pour confirmer la confiance qui leur est attribuée ou pour les éliminer comme non utiles à la justification. Les *pdj* utiles sont les *pdj* conformes à des patrons ou ceux qui conduisent à une sous-conclusion utilisée directement ou indirectement comme support dans un *pdj* conforme à un patron. Par expérience lors de deux études de cas industriels (Duffau *et al.*, 2018), ces *pdj* apparaissent lorsque les attentes des patrons nécessitent des supports qui ne peuvent pas être obtenus directement et qu'il convient donc de justifier.

Sémantiquement, déterminer les pas *utiles* revient à trouver tous les *pdj* dans le diagramme tels que

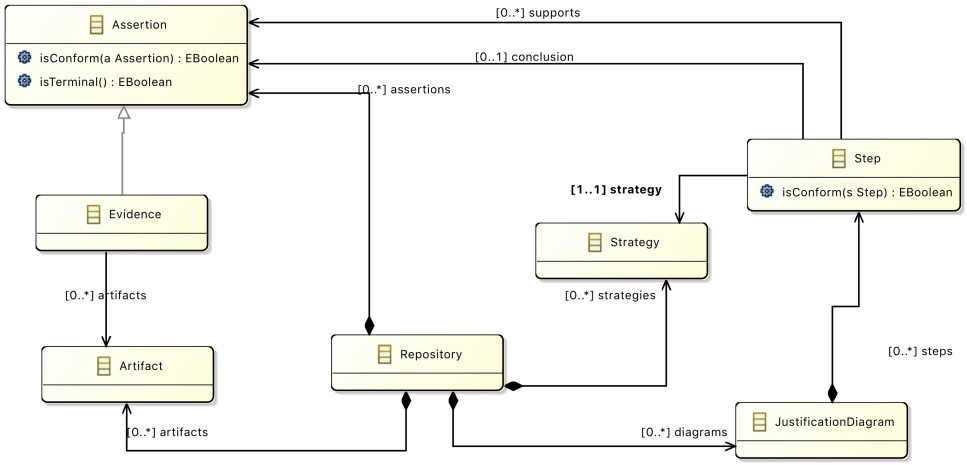

Figure 2. *Extrait du méta-modèle pour les diagrammes de justifications*

(1) $\exists p \in \mathcal{P}, pdj \mathcal{R} p, pdj \neq p$, ou
(2) $(pdj =< S, w, c >, \exists pdj\prime \in P, pdj\prime =< S\prime, w\prime, c\prime >, c \in S\prime)$ et pdj' est utile.

4.3.5. *Quels sont les supports réutilisables pour justifier d'une conclusion ?*

A partir d'un patron permettant d'atteindre une conclusion particulière, il peut être intéressant de savoir s'il existe des supports déjà présents dans un diagramme qui sont des raffinements des supports du patron. Ces supports sont soit des éléments de preuves, des faits, des données, etc. soit des conclusions de *pdj*, en d'autres termes ce sont des terminaux présents dans le diagramme de justification.

Dès lors, déterminer les supports réutilisables pour justifier d'une conclusion présente dans un patron, revient à rechercher toutes les assertions présentes dans le diagramme qui raffinent les supports du patron.

Un exemple d'une telle utilisation, pourrait être la justification de l'exécution des tests d'intégration qui a besoin que les tests unitaires aient bien été exécutés préalablement, point qui se trouve être déjà justifié par un pdj qui a été construit automatiquement comme l'illustre la Figure 3.

4.4. *Vers un outil*

Dans le méta-modèle pour les diagrammes de justification en Figure 2, nous retrouvons les concepts de base définis par le formalisme. Nous retrouvons la méthode $isConform$ qui matérialise la relation $\mathcal{R}$ entre *Assertion* ou entre *Step*. De manière identique, le concept d'assertion terminale a été implémenté par une méthode

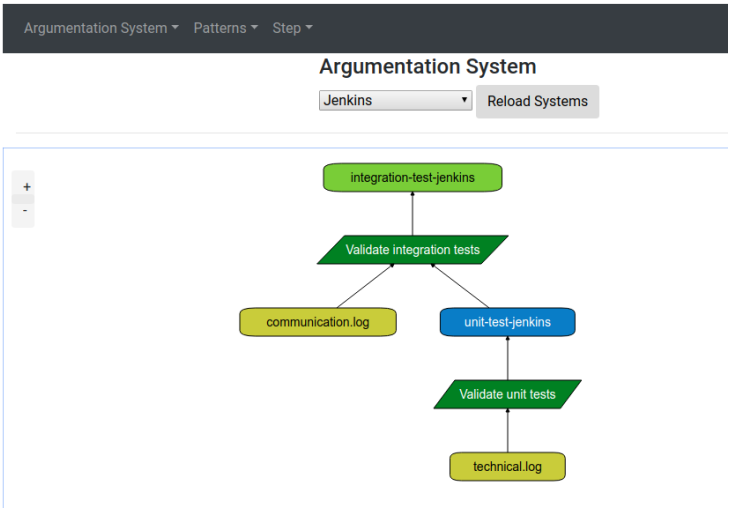

Figure 3. *Exemple d'un patron de diagrammes, qui justifie la bonne exécution de tests logiciels à travers Jenkins, conçu avec le logiciel ADEV*

$isTerminal$. Dans la pratique, les *Assertion*, *Strategy*, *Artifact*, *JustificationDiagram* sont conservés dans un *Repository* pour permettre de supporter le flot de construction, d'un pas dans un patron jusqu'à la concrétisation d'un diagramme de justification. Un *pdj* (dans la figure *Step*) est bien composé d'un tuple de *supports*, *strategy*, *conclusion*. Ce modèle permet bien de capturer la notion de *sous-conclusion* à travers l'utilisation d'assertions pouvant être utilisées comme une *conclusion* ou un *support*.

Ce modèle est au coeur du moteur d'argumentation intégré dans une usine nommée *Argumentation Factory* (Duffau *et al.*, 2017a). Cette usine expose trois services [1] permettant d'enregistrer des patrons, de construire des diagrammes et de vérifier des propriétés sur ces diagrammes. Afin de faciliter la création des patrons et pas de justification, un outil *What You See Is What You Get* a été développé, présenté en Figure 3. Cet outil interagit avec l'*Argumentation Factory* à travers les services qu'elle expose et est donc l'interface graphique proposée pour la visualisation et édition des diagrammes de justifications. Suite à ceci, grâce à la vision service mise en place, la réalisation d'un diagramme ainsi que la vérification de propriétés avec l'Argumentation Factory sont facilitées et peuvent être facilement intégrés à d'autres outils. Duffau (Duffau *et al.*, 2017b) propose, par exemple, son utilisation à travers un plugin pour Jenkins [2].

1. au niveau technologique, ces services sont des services web REST et le moteur sous-jacent est développé en Java
2. Jenkins est la plateforme d'intégration continue communément utilisée en industrie permettant de compiler, tester, déployer automatiquement des produits logiciels à partir des code sources.

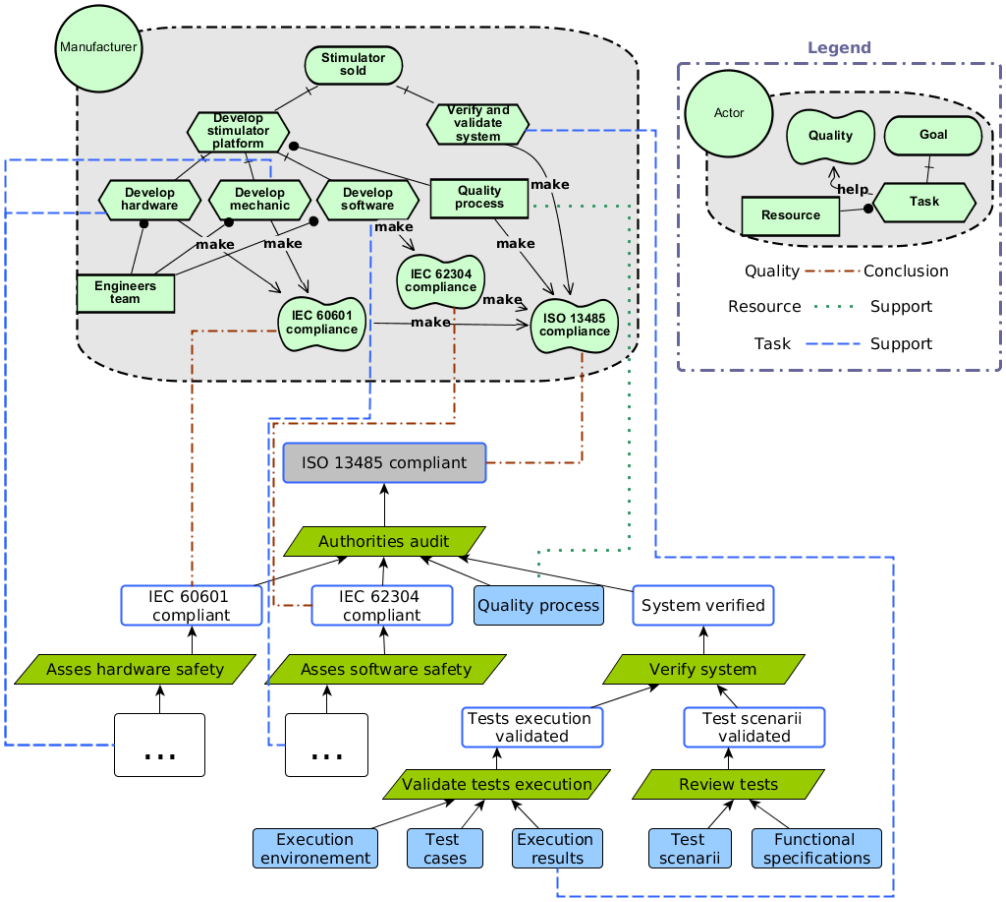

Figure 4. *Exemple d'interaction possible entre i* et les diagrammes de justifications*

5. Conclusion et perspectives

Le formalisme des diagrammes de justification a déjà été mis en oeuvre à travers un outil. Nous avons utilisé les diagrammes de justification, sur un cas industriel dans le domaine des technologies médicales. Dans cette étude, nous avons outillé un moteur d'intégration continue par appel aux services dédiés de l'*Argumentation Factory* permettant ainsi de justifier l'ensemble des processus de développement : d'une spécification à travers un ticket jusqu'à la validation de celle-ci par des tests exécutés par la plateforme d'intégration continue (pour plus de détails voir (Duffau *et al.*, 2017b)).

Dans la suite de nos travaux, nous comptons compléter les aspects théoriques des diagrammes de justification et dans le même temps étendre leurs usages sur d'autres

applications industrielles. Ainsi parmi les applications possibles, nous avons commencé à étudier l'utilisation des diagrammes de justification pour la certification de processeurs pluri-cœurs dans le cadre avionique (Bieber *et al.*, 2018). Là encore, nous cherchons à définir des patrons génériques qui permettent in fine de justifier des objectifs haut niveau, objectifs étant définis par les documents de certification.

Concernant la partie plus théorique, nous pouvons désormais axer la suite de nos travaux sur la collaboration avec d'autres modèles. Par exemple, le langage i* définit des types de contribution (crée, aide, nuit, casse) qui permet de définir des liens de contribution entre une qualité et une autre qualité, une tâche ou un objectif. Puisque le langage propose ce mécanisme, nous proposons de nous brancher sur i* pour permettre de justifier ces qualités notamment à travers ces types de contributions. Un exemple d'une telle interaction est donnée en Figure 4. Le diagramme i* présente les exigences permettant d'atteindre l'objectif de vendre des neurostimulateurs en prenant en compte des qualités comme la conformité avec la norme ISO 13485. L'atteinte de cette qualité est justifiée par le diagramme de justification du dessous. Les liens en pointillés entre les deux diagrammes matérialisent les points d'interaction possibles entre ces deux domaines. A travers cet exemple, nous illustrons la complémentarité de ces deux approches qui permet ainsi d'avoir une plus grande confiance dans l'atteinte des qualités souhaitées.

L'établissement de liens entre les diagrammes de justification et SACM nous permettent déjà d'envisager de vérifier les propriétés que nous avons présentées ici sur ces modèles. Modulo une difficulté liée à la multiplicité des conclusions dans SACM, nous pensons pouvoir définir une relation $\mathcal{R}$ propre aux modèles SACM, et ainsi introduire sémantiquement la notion de patron qui est aujourd'hui diffuse dans plusieurs méta-éléments. C'est une de nos perspectives à court terme.

6. Bibliographie

Alexander C., Ishikawa S., Silverstein M., i Ramió J. R., Jacobson M., Fiksdahl-King I., *A pattern language*, Gustavo Gili, 1977.

Association for the Advancement of Artificial Intelligence, *Proceedings of the the AAAI Spring Symposium on producing cooperative explanations*, 1992.

Bieber P., Boniol F., Bouchebaba Y., Brunel J., Pagetti C., Poitou O., Polacsek T., Santinelli L., Sensfelder N., « A model-based certification approach for multi/many-core embedded systems », *ERTS*, 2018.

Chandrasekaran B., Tanner M. C., Josephson J. R., « Explaining Control Strategies in Problem Solving », *IEEE Intelligent Systems*, vol. 4, 1989, p. 9-15, 19-24, IEEE Computer Society.

Dalpiaz F., Franch X., Horkoff J., « iStar 2.0 Language Guide », *CoRR*, vol. abs/1605.07767, 2016.

de la Vara J. L., Panesar-Walawege R. K., « Safetymet: A metamodel for safety standards », *International Conference on Model Driven Engineering Languages and Systems*, Springer, 2013, p. 69–86.

Duffau C., Camillieri C., Blay-Fornarino M., « Improving Confidence in Experimental Systems through Automated Construction of Argumentation Diagrams », *ICEIS 2017*, vol. 1,

2017, p. 495–500.

Duffau C., Grabiec B., Blay-Fornarino M., « Towards Embedded System Agile Development Challenging Verification, Validation and Accreditation : Application in a Healthcare Company », *ISSREW 2017*, 2017.

Duffau C., Polacsek T., Blay-Fornarino M., « Support of Justification Elicitation: Two Industrial Reports », *Advanced Information Systems Engineering - 30th International Conference, CAiSE 2018, Tallinn, Estonia, June 11-15, 2018. Proceedings*, Lecture Notes in Computer Science, Springer, 2018.

Emmet L., Cleland G., « Graphical notations, narratives and persuasion: a Pliant Systems approach to Hypertext Tool Design », Blustein J., Allen R. B., Anderson K. M., Moulthrop S., Eds., *HYPERTEXT 2002, Proceedings of the 13th ACM Conference on Hypertext and Hypermedia*, ACM, 2002, p. 55–64.

Gamma E., Helm R., Johnson R., Vlissides J., *Design Patterns: Elements of Reusable Object-oriented Software*, Addison-Wesley Longman Publishing Co., Inc., Boston, USA, 1995.

International Joint Conferences on Artificial Intelligence, *Proceedings of the IJCAI'93 Workshop on Explanation and Problem Solving*, 1993.

Kelly T., Weaver R., « The Goal Structuring Notation /- A Safety Argument Notation », *Proc. of Dependable Systems and Networks 2004 Workshop on Assurance Cases*, 2004.

Matsuno Y., « A design and implementation of an assurance case language », *Dependable Systems and Networks (DSN), 2014 44th Annual IEEE/IFIP International Conference on*, IEEE, 2014, p. 630–641.

OMG, « Structured Assurance Case Meta-model », 2013.

Perelman C., Olbrechts-Tyteca L., *Traité de l'argumentation : La nouvelle rhétorique*, Presses universitaires de France, 1958.

Polacsek T., « Validation, accreditation or certification: a new kind of diagram to provide confidence », *Research Challenges in Information Science*, IEEE, 2016.

Polacsek T., « Diagramme de justification. Un outil pour la validation, la certification et l'accréditation », *Ingénierie des Systèmes d'Information*, vol. 22, n° 2, 2017, p. 95–119.

Southwick R. W., « Explaining reasoning: an overview of explanation in knowledge-based systems », *Knowledge Eng. Review*, vol. 6, n° 1, 1991, p. 1–19.

Toulmin S. E., *The uses of argument*, Cambridge University Press, 2003.

Van Zee M., Marosin D., Bex F., Ghanavati S., « RationalGRL: A Framework for Rationalizing Goal Models Using Argument Diagrams », *Conceptual Modeling: 35th International Conference, ER 2016*, Springer, 2016, p. 553–560.

Weinstock C. B., Goodenough J., « Towards an assurance case practice for medical devices », rapport, 2009, Software Engineering Institute.

Ye L. R., Johnson P. E., « The Impact of Explanation Facilities in User Acceptance of Expert System Advice », *MIS Quarterly*, vol. 19, n° 2, 1995, p. 157–172.

Ye L. R., « The value of explanation in expert systems for auditing: An experimental investigation », *Expert Systems with Applications*, vol. 9, n° 4, 1995, p. 543–556, Elsevier.

Yu E. S., « Towards modelling and reasoning support for early-phase requirements engineering », *Requirements Engineering, 1997., Proceedings of the Third IEEE International Symposium on*, IEEE, 1997, p. 226–235.