Réflexions sur les liens possibles entre Argumentation et V&V pour le Logiciel

Thomas Polacsek

ONERA, Département Traitement de l'Information et Modélisation
2, avenue Edouard Belin BP74025, 31055 TOULOUSE Cedex 4

RÉSUMÉ. *Le but de cet article est d'ouvrir une piste de réflexion concernant la possibilité d'utiliser les travaux existants dans le domaine de l'argumentation et, plus précisément, dans l'argumentation légale, pour les appliquer dans le cadre de l'acceptation d'un logiciel par une autorité, que cela soit une autorité de certification ou un client. A terme, l'objectif de cette approche est de définir un cadre pour la construction d'une argumentation, que l'on pourrait qualifier d'acceptable, cadre qui structurerait l'ensemble des documents composant le dossier de Vérification et Validation (V&V).*

MOTS-CLÉS : *argumentation, V&V, VV&A*

1. Introduction

Dans le cadre de l'ingénierie de la simulation est apparue depuis une dizaine d'année le terme de VV&A pour *Verification, Validation and Accreditation*[1]. Ici, aux opérations de vérification et de validation vient s'ajouter une activité dite d'accréditation qui consiste à ce qu'une autorité accepte l'usage d'une simulation ou d'un modèle pour une utilisation dans un contexte précis[2]. Nous pouvons faire un parallèle entre cette activité dans le domaine de la simulation et la tâche consistant, pour une autorité, telle qu'un client ou une autorité de certification, à accepter un logiciel, qui peut correspondre à sa validation ou sa certification. Pour réaliser cela, il faut disposer d'une documentation exhaustive à ce sujet, expliquant non seulement les résultats, mais aussi les données d'entrées, les hypothèses faites, les techniques appliquées, etc. La tâche d'accréditation consiste donc à collecter cette documentation, mais surtout à l'évaluer. Cette documentation n'étant pas formelle, ou du moins pas dans son intégralité, il paraît vain de chercher à en établir uniquement de façon formelle sa validité.

Face à de tels énoncés, nous devons substituer à la notion de validité celle d'acceptabilité. Il n'est plus question ici de la vérité d'un énoncé mais d'étudier si un énoncé est acceptable ou pas. Nous allons donc chercher à modéliser des énoncés (la documentation) que des logiciens jugent non scientifiques. D'ailleurs, (Carnap, 1962) qualifie de tels énoncés de préscientifiques, il les considère comme vagues et incorrects. En ce sens, nous pouvons opérer un rapprochement avec les travaux de (Hamblin, 1970) qui remet en question l'utilisation de la logique formelle face à l'étude de l'argumentation. Son but étant de comprendre ce qui

1. Définition donnée par le Département de la Défense des États-Unis, on trouve aussi dans la littérature le terme d'*acceptation* (*acceptance*).
2. DoD directive 5000.59 : *"the official certification that a model or simulation is acceptable for a specific purpose"*.

rend une argumentation acceptable, il donne un nouveau modèle de la validité d'un raisonnement où la validité ne dépend pas de critères logiques relatifs à la vérité des prémisses mais à des critères dialectiques. Pour lui, la relation entre les prémisses et la conclusion n'est plus de l'ordre de l'implication logique mais d'une dialectique qui autorise, ou interdit, des comportements discursifs. Notons que l'auteur y expose sa préférence pour le terme acceptabilité plutôt que celui de validité. Au même moment, les travaux de (Perelman et Olbrechts-Tyteca, 2008) définissent une nouvelle théorie de l'argumentation basée sur une approche dialectique. Pour eux, les logiciens n'admettent comme rationalité que la démonstration logique. Dès lors, il devient impossible d'établir des raisonnements autres que purement formels ce qui est *"une limitation indue et parfaitement injustifiée du domaine où intervient notre faculté de raisonner et de prouver"*.

L'étude de l'argumentation s'intéresse aux liens entre hypothèses et conclusions, à la structuration du raisonnement. La notion d'argumentation renvoie bien évidemment à la notion de preuve qui a largement évoluée dans l'histoire des sciences et qui ne revêt pas le même sens suivant que l'on se trouve dans les disciplines formelles et axiomatiques, les sciences expérimentales ou les sciences humaines et sociales. Aujourd'hui, l'étude de la validité d'une argumentation et des mécanismes sous-jacents est étudiée par un ensemble de disciplines telles que : l'informatique, la linguistique, l'épistémologie et les sciences légales.

Le but de cet article est donc d'ouvrir une piste de réflexion concernant la possibilité d'utiliser les travaux existants dans le domaine de l'argumentation pour les appliquer dans le cadre de l'acceptation d'un logiciel par une autorité. Notons que l'idée de structurer l'ensemble des éléments servant à établir un fait sous la forme d'un arbre d'argumentation semble aujourd'hui faire son chemin, notamment au travers de ce qui se nomme *Assurance Case*. De plus, un groupe au sein de l'Object Management Group[3] cherche à définir un métamodel de l'argumentation. Cependant, il nous semble primordiale que tous les travaux cherchant à modéliser une argumentation ne se bornent pas à définir des diagrammes de boites et de flèches, mais s'inscrivent plutôt dans la lignée des travaux menés en linguistique et en droit.

2. Schéma de Toulmin

C'est en 1958 que (Toulmin, 2003) présente son schéma de l'argumentation. Ce modèle, bien que quasiment inexistant dans la littérature scientifique française, est enseigné dans de nombreuses universités américaines afin d'expliquer les mécanismes de l'argumentation. Il sert, par exemple, dans l'enseignement et la compréhension de disciplines telles que l'argumentation légale ou encore, dans ce que les anglophones nomment *critical thinking*.

Dans le modèle de Toulmin, toute argumentation est composée d'une *conclusion*, notée (C) sur la Figure 1, et des *données* (D). Pour justifier le passage des données à la conclusion, on utilise des données supplémentaires appelées *garanties* (W). Distinguer données et garanties n'est pas toujours chose aisée, les garanties sont générales, elles attestent de la solidité de l'argumentation. Une conclusion n'étant pas toujours absolue, Toulmin ajoute la possibilité d'exprimer des réserves à l'aide de *qualificateurs modaux* (Q). Ces qualificateurs correspondent à des notions telles que possiblement, probablement. A cela se rajoute des *conditions de réfutation* (R) qui expriment les circonstances dans lesquelles la conclusion n'est pas vraie. Pour finir, Toulmin ajoute les *fondements* (B) qui sont les justifications des garanties.

A titre d'exemple considérons le cas suivant : nous avons (C) *"des données qui sont cryptées avec OpenSSL et une clé de 128 bits"* et nous savons, en nous appuyant sur l'Advanced Encryption Standard,

3. http ://www.omg.org/spec/ARM/

(D) les données sont cryptées avec OpenSSL et une clé de 128 bits	$\longrightarrow$	*Donc* (Q) nécessairement (C) les informations contenues dans le fichier ne peuvent pas être accédée par quelqu'un non autorisé à avoir la clé *Sauf si* (R) quelqu'un vole la clé
	Vu que (W) une clé de 128 bits suffit pour protéger les données d'après l'Advanced Encryption Standard. *En vertu de* (B) l'Advanced Encryption Standard publié en 2001 préconise une taille de clé de 128 bits (FIPS PUB 197)	

Figure 1. *Un exemple du schéma d'argumentation de Toulmin*

qu'une clé de 128 bits suffit pour protéger les données. Nous nous intéressons à l'argumentation, au pas de raisonnement, qui consiste à dire si (D) est vraie alors nous pouvons conclure que (C) *"les informations contenues dans le fichier ne peuvent pas être accédée par quelqu'un non autorisé à avoir la clé"*. Comme nous ne sommes pas dans un cadre formel, nous ne disposons pas d'une théorie axiomatique qui puisse nous donner la valeur de validité de la formule logique $D \rightarrow C$. Nous sommes ici dans un cadre purement rhétorique où nous devons essayer de définir si nous sommes face à une *"bonne"* argumentation ou pas. Dans cet exemple, la conclusion n'est pas toujours vraie. En effet, la confidentialité des informations contenues dans le fichier n'est pas établie dans l'absolue : un attaquant peut voler la clé. Nous devons donc ajouter une condition de réfutation, ce qui nous donne le schéma Figure 1.

3. Un schéma d'argumentation simplifié

Dans le cadre d'une argumentation visant à structurer les éléments permettant de statuer sur la validité d'une propriété logicielle, certaines subtilités développées par Toulmin ne nous semblent pas nécessaires. En effet, dans notre cas, nous cherchons à établir des propriétés qui sont toujours vraies, dès lors les *qualificateurs modaux* et les *conditions de réfutation* sont superflus. De plus, nous voulons simplifier le travail d'acceptation sans compliquer celui de vérification. Par conséquent, nous recherchons un schéma d'argumentation qui soit à la foi simple et utile. Nous proposons donc un schéma simplifié, structuré autour de deux notions clés : l'élément de preuve et la stratégie, ces deux éléments nous permettant d'établir une conclusion.

3.1. *Données ou éléments de preuve*

Toute argumentation, démonstration, se fonde sur des vérités préétablies. Par exemple, une démonstration dans un système formel postulera toujours qu'un ensemble d'axiomes sont vrais. Ces axiomes sont vrais par nature, il n'existe pas de démonstration qui les prouve, ils sont l'élément de base de tout raisonnement dans ce système. De façon analogue, toute argumentation repose sur un ensemble de postulats, acceptées par celui qui énonce la démonstration ainsi que son auditoire. Nous appellerons ces vérités préétablies : *éléments de preuve*[4].

Dans le cadre de l'argumentation légale, (Rodney A. Reynolds, 2002) définissent les éléments de preuve comme : "les données (faits et opinions) présentées comme des preuves pour une affirmation[5]". Les éléments de preuve ont la particularité de reposer sur l'autorité de celui qui les énonce. La validité, ou plutôt

4. traduction du mot anglais *"evidence"*.
5. *"data (facts or opinions) presented as proof for an assertion"*.

l'acceptation par l'auditoire, d'un fait ne repose plus sur le fait lui-même, mais sur la confiance que l'on accorde à celui qui l'énonce. Des exemples simples peuvent être : un résultat donné dans un article scientifique, une information donnée par un expert ou une pratique définie dans une norme. Ainsi, ce n'est pas la validité de l'information donnée qui est à démontrer, c'est la crédibilité de celui qui l'énonce, la source, qui est à prouver. Nous sommes typiquement ici dans un problème de confiance. Notons que la confiance n'est pas une valeur absolue, la confiance est relative à un domaine. On a confiance en quelqu'un dans un certain cadre.

3.2. *La garantie ou stratégie*

La *garantie* chez Toulmin est la pierre angulaire du raisonnement. C'est la garantie qui explicite clairement comment, à partir de données, il est possible d'inférer une conclusion. Remarquons que ce que Toulmin appelle *garantie* correspond exactement à ce que l'ISO 15026[6] appelle *Arguments* et que le Goal Structured Notation (GSN) (Kelly et Weaver, 2004) appelle *Strategy*. Afin d'homogénéiser les différents termes, nous avons décidé de ne plus utiliser le terme de garantie de Toulmin, mais le terme *stratégie*.

Parce que nous sommes dans un cadre où l'argumentation vise à convaincre une autorité (et doit être acceptée par elle), en plus de la stratégie, nous gardons le concept de Toulmin qui lui est directement associé : le *fondement*. Pour nous, les fondements sont les justifications sur le pourquoi une stratégie est acceptable. Prenons le cas de l'usage d'un outil d'analyse statique, l'utilisation de ce logiciel doit être motivée : il faut indiquer, par exemple, si le logiciel est acceptable dans le cadre de cette étude (cette acceptation pouvant être possiblement une certification) et, pourquoi pas, renvoyer à l'argumentation du logiciel qui elle-même explicitera son algorithme, son implémentation, etc.

4. Un exemple d'arbre d'argumentation

Dans (Pires *et al.*, 2013) les auteurs proposent une méthode pour vérifier automatiquement la conformité d'un code C, conforme à un certain patron de code, par rapport à sa spécification exprimée à l'aide d'une machine à états UML. Pour ce faire, ils utilisent des techniques d'analyse statique qui permettent la vérification de propriétés sur un programme sans avoir à l'exécuter. De façon pratique, ils génèrent automatiquement des annotations de preuve en langage ACSL[7] dans un code C à partir du modèle lui servant de spécification et vérifient ces annotations à l'aide d'un outil d'analyse statique, ici Frama-C. Si nous cherchons à établir l'argumentation, s'appuyant sur cette méthode, visant à établir la propriété qu'un code est conforme à sa spécification, nous devons éviter deux écueils.

Le premier consiste à assembler tous les documents (ici la spécification UML, les résultats donnés par Frama-C, etc.), considérer que ce sont autant d'éléments de preuve et que, suivant la stratégie qui consiste à suivre la méthode, la propriété est valide. Bien évidemment, ceci n'est pas une argumentation, cela ne structure pas le raisonnement sous-jacent à l'ensemble des documents. Dans la pratique, une telle approche a pour seul effet de noyer l'autorité en charge de l'acceptabilité sous un flot de documents, sans aucune indication sur l'articulation de l'ensemble.

---

6. La norme ISO/IEC 15026 est une norme applicable aussi bien à des systèmes qu'à des logiciels. Elle permet de définir des niveaux d'intégrités.
7. ANSI/ISO C Specification Language.

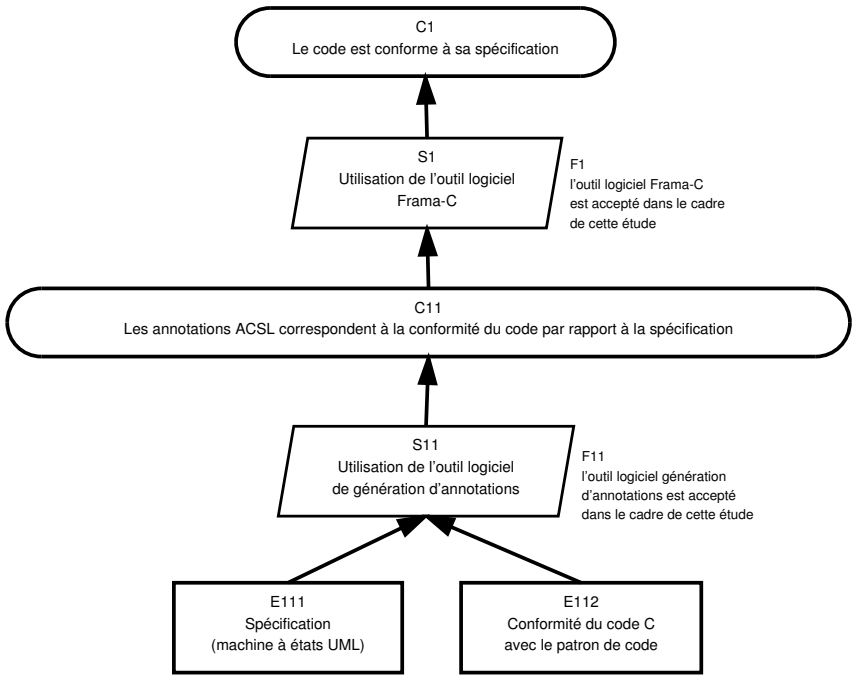


Figure 2. *Exemple d'arbre d'argumentation*

Le deuxième écueil consiste à confondre argumentation et description du processus. Le but d'une argumentation n'est pas d'expliquer les étapes qui ont permis de vérifier une propriété mais d'expliciter, sans ambiguïté, quels documents permettent d'établir la validité d'une propriété.

Finalement, une *bonne* argumentation, pour notre exemple, se compose de deux pas de raisonnement : Figure 2. Les stratégies employées dans cet arbre sont toujours de même nature, elles relèvent de résultats donnés par l'utilisation d'un outil logiciel s'appuyant sur des méthodes formelles. Attention, n'oublions pas que cette argumentation n'est qu'une représentation de l'articulation de l'ensemble des éléments qui permettent d'atteindre la conclusion, chaque élément doit renvoyer à un document (voir à un ensemble de documents) qui l'explicite.

Le premier pas de raisonnement dans notre arbre repose sur deux éléments de preuve qui renvoient (E111) à la spécification du système en UML et (E112) à un document qui établit la conformité du code C avec le patron de code de la méthode. De là, en appliquant la stratégie (S11) d'utilisation de l'outil logiciel de génération d'annotations, nous pouvons conclure que (C11) nous avons des annotations ACSL qui correspondent à la conformité du code par rapport à la spécification. Les fondements associés à cette stratégie sont que (F11) l'outil logiciel est accepté dans le cadre de cette étude (ce qui peut renvoyer, par exemple, au cahier des charges de l'étude). La conclusion (C11) du premier pas de raisonnement est l'élément de preuve du deuxième pas de notre argumentation qui, s'appuyant sur la stratégie (S1) d'utilisation de l'outil logiciel Frama-C, conclue que notre propriété de haut niveau est valide. Notons que des documents sont aussi associés aux stratégies, ainsi, est associé à (S1) les documents établissant que le solveur utilisé par Frama-C a bien prouvé les annotations.

5. Perspectives

Maintenant que nous disposons d'une argumentation qui motive l'acceptation d'une propriété, nous pourrions nous interroger sur le bien fondé de cette argumentation. Plus précisément, est-il pas possible de fournir à l'autorité en charge de la tâche d'acceptation du logiciel des éléments supplémentaires lui permettant de statuer sur le bien fondé de l'application d'une stratégie ? Pour cela, nous pouvons faire un parallèle entre les résultats donnés par de tels logiciels et la parole d'un expert. En effet, ces résultats relèvent d'un niveau d'expertise dont ne dispose pas forcément l'autorité en charge de l'acceptation, mais qui doit pour autant statuer sur l'acceptation, ou pas, des résultats donnés par un outil logiciel ou un expert. Pour l'aider dans cette tâche, nous proposons d'ajouter à notre schéma d'argumentation la notion de questions critiques. Depuis de nombreuses années, Douglas Walton mène des travaux où il cherche à définir un schéma représentant la parole d'un expert et à analyser comment une parole d'expert peut être réfutée ou affaiblie (Walton, 1996 ; Godden et Walton, 2006). Pour cela, il a défini ce qu'il appelle des *questions critiques*. Le but de cet ensemble de questions est d'évaluer et d'analyser de manière simple la parole d'un expert. Sur ce modèle, nous pourrions attacher à notre schéma d'argumentation dans le cadre de l'utilisation d'un outil logiciel des questions critiques. Le but de ces questions est, d'une part, d'obliger le créateur de l'argumentation à vérifier que son pas d'argumentation est bien construit, d'autres part, quand c'est nécessaire, d'ajouter à l'argumentation des documents répondant aux questions que peut légitimement se poser un relecteur.

Dans notre proposition, nous sommes restés à un stade très informel. Il pourrait être intéressant, dans l'avenir, de définir un cadre plus formel d'arbre d'argumentation, cadre qui nous permettrait de réaliser des opérations automatiques telles que : l'analyse des éléments de preuves, détecter les manques, les stratégies incomplètes, etc. Pour finir, une telle approche ne peut à terme se concevoir sans outil. Les arbres d'argumentations étant potentiellement immenses et gérant de l'hypertextualité, la question d'outil informatique de visualisation et de navigation reste cruciale pour leur utilisation sur des cas réels.

6. Bibliographie

Carnap R., *Logical Foundations of Probability*, University of Chicago Press, 1962.

Godden D. M., Walton D., « Argument from Expert Opinion as Legal Evidence : Critical Questions and Admissibility Criteria of Expert Testimony in the American Legal System », *Ratio Juris*, vol. 19, n° 3, 2006, p. 261–286, Blackwell Publishing.

Hamblin C., *Fallacies*, University paperback, Methuen, 1970.

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

Perelman C., Olbrechts-Tyteca L., *Traité de l'argumentation : La nouvelle rhétorique*, UBlire - Fondamentaux, Éditions de l' Université de Bruxelles, 2008.

Pires A. F., Polacsek T., Wiels V., Duprat S., « Behavioural Verification in Embedded Software, from Model to Source Code », Moreira A., Schätz B., Gray J., Vallecillo A., Clarke P. J., Eds., *MoDELS*, vol. 8107 de *Lecture Notes in Computer Science*, Springer, 2013, p. 320-335.

Rodney A. Reynolds J. L. R., « *Evidence* », p. 427-446, SAGE Publications, 2002.

Toulmin S. E., *The Uses of Argument*, Cambridge University Press, Cambridge, UK, 2003, Updated Edition, first published in 1958.

Walton D. N., « Practical Reasoning and the Structure of Fear Appeal Arguments », *Philosophy and Rhetoric*, vol. 29, n° 4, 1996, p. 301–313.