Vers l'argumentation automatique d'expérimentations : application à un portfolio de workflows

Clément Duffau, Cécile Camillieri, Mireille Blay-Fornarino

Vers l'argumentation automatique d'expérimentations : application à un portfolio de workflows *

Clément DUFFAU[1,2], Cécile CAMILLIERI[1], and Mireille BLAY-FORNARINO[2]

[1] Université Nice Côte d'Azur
Laboratoire I3S prenom.nom@unice.fr
[2] AXONIC

Résumé

De nombreux systèmes sont construits aujourd'hui sur la base d'expérimentations à partir desquelles des connaissances sont apprises et construites. Ces connaissances évoluent en fonction des nouvelles expérimentations, de même que les systèmes qui les exploitent. Comment justifier la confiance dans ces systèmes non pas a posteriori mais au fur et à mesure de l'évolution de ces systèmes ? Au travers d'une étude de cas basée sur la mise au point d'un portfolio de ≪ Machine Learning ≫ nous présentons notre approche qui repose sur l'utilisation de graphes d'argumentation et leur construction automatique au fur et à mesure des évolutions de la base d'expérimentations et des connaissances acquises.

1 Introduction

Lorsqu'un système est construit sur la base des expérimentations, il est important d'avoir confiance dans le déroulement des expérimentations, dans les analyses qui en sont faîtes et dans leur interprétation. Pour cela, différents guides ont été définis qui visent à "certifier" et évaluer le niveau de confiance entre les données expérimentales et la précision prédictive issue de l'analyse et des expérimentations [1]. L'accréditation et plus largement la confiance que l'on met dans un système expérimental [1] reposent alors sur différents artefacts justifiant de la qualité des activités et du processus suivi e.g. humains, documents, logiciels. Tracer et documenter ces activités sont alors des tâches essentielles et lourdes qui peuvent générer un ensemble difficilement appréhendable de justifications. Il est difficilement appréhendable, dans le sens d'une masse énorme, peu structurée et ne faisant pas clairement le lien entre les résultats dont les documents de justification rendent compte et la fiabilité du produit final. Pour faire face à ce tsunami documentaire, Polacsek propose un nouveau type de diagramme qui vise à structurer une argumentation de certification [9]. L'argumentation correspond à un ≪ ensemble de techniques discursives destinées à provoquer ou à accroître l'adhésion de l'interlocuteur aux thèses qui lui sont présentées ≫ (Larousse). Malgré une correspondance très forte entre la notion d'accréditation et de confiance dans un système, la relation entre les deux n'a été établie dans un cadre logiciel que récemment par les travaux de Polascek. Ces travaux sont cantonnés à une description et une utilisation manuelles de ce qu'est un diagramme d'argumentation sans définir de cadre permettant leur utilisation dans le contexte de systèmes expérimentaux qui évoluent fréquemment. De manière complémentaire, des travaux autour de la construction automatique d'argumentation ont été menés dans une approche rétrospective [8]. A notre connaissance aucun des travaux liant argumentation et systèmes expérimentaux, peut-être parce que plus liés à des problématiques d'accréditation que de confiance, ne s'est intéressé à la construction automatique de l'argumentaire tout au long du cycle de vie du système expérimental. Or dans le contexte de systèmes expérimentaux qui doivent évoluer automatiquement en fonction des nouvelles expérimentations cette dimension est essentielle.

Nous proposons dans cet article d'expliciter la problématique liée à la construction de cette argumentation au travers d'une étude de cas portant sur une ligne de produits logiciels construite à partir d'expérimentations.

*Merci à Frédéric Precioso, co-fondateur du projet ROCKFlows et Thomas Polascek qui travaille avec nous à l'argumentation.

1. au sens *"A basic unit of experimental activity combining local, technical, instrumental, institutional, social, and epistemic aspects."* défini par Rheinberger [11]

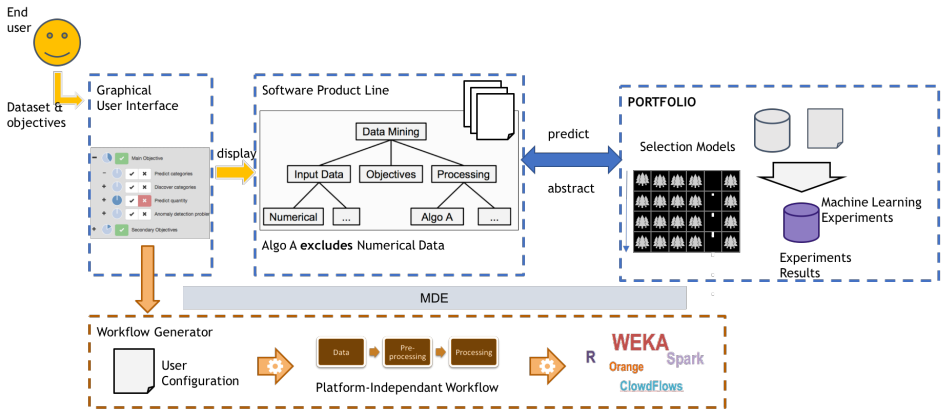

FIGURE 1 – Vision globale de Architecture de RockFlows

Cet article présente dans une première partie notre étude de cas et les objectifs associés, puis notre approche de l'argumentation dans ce cadre avant de conclure.

2 Une ligne de produits logiciels basée sur un portfolio d'algorithmes

D'un point de vue utilisateur béotien, la construction de workflows de Machine Learning (ML) est complexe par l'immense variabilité des algorithmes, des préprocessing et des stratégies de paramétrisation [6] mais également par la difficulté de faire les bons choix ; le meilleur algorithme n'est pas le même pour chaque problème [14], il dépend entre autre de la taille et de la nature des données et de ce que l'utilisateur veut apprendre de ces données. Pour répondre à cette problématique nous travaillons sur le projet ROCKFlows [2] qui, en fonction du jeu de données de l'utilisateur et de ses objectifs, vise à générer le workflow de ML le plus approprié [5]. La plateforme ROCKFlows repose sur la construction d'un portfolio [7] de workflows de ML, associé à une Ligne de Produits Logiciels (LPL). Le portfolio se présente comme un recueil d'algorithmes et de jeux de données à partir desquels nos expérimentations sont menées. Sur la base des résultats obtenus nous construisons des modèles prédictifs permettant de sélectionner en fonction d'un problème donné les workflows possibles et de prédire leurs performances. Cette approche rejoint, dans le contexte du ML, celle de ASLIB pour la sélection d'algorithmes SAT ou CSP[4]. Les informations se trouvant dans le portfolio sont remontées par abstraction au niveau de la ligne de produits dans des termes utilisateurs. Cette remontée est nécessaire pour réduire les choix de l'utilisateur et supporter le processus de configuration, tout en favorisant une évolution rapide de la ligne. L'étape de configuration d'un produit consiste alors à fournir le jeu de données ou les méta-informations qui lui sont associées, à sélectionner les principaux objectifs puis en fonction des prédictions proposées à choisir un workflow. Celui-ci est alors généré automatiquement. Il ne s'agit donc pas d'une approche d'assemblage de workflows mais bien d'une génération à partir d'objectifs de haut niveau. L'architecture de ROCKFlows est présentée en figure 1.

Au niveau de l'interface utilisateur n'apparaissent évidemment que les informations pertinentes en fonction du problème posé. Les prédictions pour chacun des workflows possibles, *e.g.*, la précision et le temps d'apprentissage, résultent d'un processus complexe d'apprentissage. Dans ce contexte, comment justifier à l'utilisateur qui le souhaite la pertinence des prédictions ? La justification repose sur les jeux de données sur lesquels nous avons appris, des protocoles d'expérimentations utilisés, de la manière dont le modèle de prédiction a été construit, etc.

2. rockflows.i3s.unice.fr

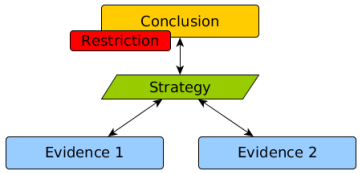

FIGURE 2 – Modèle d'un pas d'argumentation

Construction incrémentielle de l'argumentation Afin de répondre à ces questions, nous travaillons à produire automatiquement et de manière incrémentielle un graphe d'argumentation qui trace les activités permettant de construire la connaissance utilisée au niveau de la LPL. Cela suppose non seulement de prendre en compte les "ajouts" comme de nouveaux résultats d'expérimentation ou l'ajout d'un algorithme, mais également de filtrer les éléments non pertinents comme par exemple des jeux de données qui n'apportent pas de plus-value aux résultats et donc à l'argumentation.

Cohérence par construction de l'argumentation. Chaque activité (*e.g.*, expérimentation, analyse, abstraction) peut faire l'objet d'une argumentation. La manière dont celle-ci est construite doit être cohérente par rapport à l'activité elle-même mais également par rapport à l'ensemble de l'argumentation. Par exemples, l'ajout d'une nouvelle expérimentation doit vérifier que toutes les informations relatives à son contexte d'exécution sont bien données tandis que la construction des modèles de sélection ne doit se faire qu'à partir de résultats ayant suivi les mêmes protocoles d'expérimentations (*e.g.*, une évaluation par *cross-validation sur 10 folds*). Ces contraintes dans la construction de l'argumentation sont elles-même évolutives en fonction de nos apprentissages et des biais relevés dans la littérature, comme par exemple des comparaisons entre des algorithmes sur des données préparées différemment ou en n'utilisant pas exactement les mêmes méthodes d'évaluation [6].

Utilisabilité de l'argumentation Construire l'argumentation et assurer sa cohérence ne suffit pas. Cette argumentation doit pouvoir être présentée aux utilisateurs de la LPL ou aux experts pour validation et discussion. Ainsi, les experts métier doivent être en mesure de la lire facilement, en particulier en naviguant entre les arguments et les artefacts expérimentaux.

3 Des expérimentations à l'argumentation

Expérimentations La partie "Portfolio" se présente comme une infrastructure qui supporte la gestion des expérimentations sur les workflows. En cela elle se rapproche de ce que Basili définit comme une *Usine d'Experiences (UE)* [3][2]. Au niveau du processus expérimental nous nous intéressons en particulier aux activités : exécution, analyse/interprétation et présentation/packaging [13, Chapitre 2]. Dans notre contexte, un peu différent d'un développement logiciel, la phase d'exécution consiste à exécuter les différents workflows sur les différents jeux de données et à recueillir les résultats de ces expérimentations. La phase d'analyse consiste entre autre à construire les modèles de prédiction par régression qui permettront de prédire en fonction d'un jeux de données les différentes valeurs de métriques. La phase de "présentation" correspond à remonter les informations dans la ligne de produits. Dans la lignée de Ras *et all.* , nous envisageons d'outiller le portfolio par la mise en place de services [10] pour accéder aux artefacts d'expérimentation.

Diagramme d'argumentation Polacsek propose d'utiliser des diagrammes d'argumentation [9], dérivés du modèle d'argumentation mis en évidence par Toulmin [12]. Cette approche à l'aide de diagramme a déjà été utilisée dans le cadre de la certification avionique pour le logiciel embarqué [3]. La représentation d'un pas d'argumentation basé sur ces travaux est donnée en Figure 2 ; à partir de deux *Evidence*, selon une *Strategy*, une *Conclusion* a été établie relativement à une *Restriction*.

3. *Experience Factory* dans le texte originel

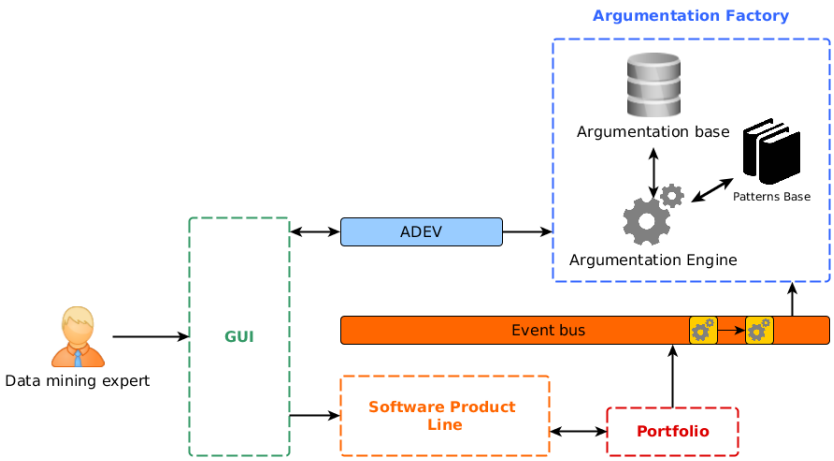

FIGURE 3 – Vision globale de l'architecture : Interaction entre le portfolio et l'usine d'argumentation via un bus d'évènements

Les pas d'argumentation peuvent être chaînés pour réutiliser une conclusion comme une évidence [4] d'une autre stratégie. Dans ROCKFlows, ces diagrammes peuvent devenir volumineux, le besoin d'automatiser leurs constructions est crucial pour le passage à l'échelle de notre solution [5].

Patrons d'argumentation Lors de la construction d'un système par expérimentation, on décide des règles à appliquer et respecter, *e.g.*, expérimentations, processus d'analyse. Au fur et à mesure que nos connaissances évoluent ces règles se modifient et s'enrichissent pour assurer la qualité des connaissances résultantes. Nous identifions ces règles comme des « patrons d'argumentation ». Un patron d'argumentation est un cadre générique qui définit précisément pour une stratégie donnée tous les éléments exigés et la conclusion attendue. L'enchaînement de ces patrons est également sujette à un contrôle (*e.g.*, on ne doit pas construire un modèle de prédiction à partir de résultats qui n'auraient pas été obtenus dans les mêmes "conditions"), nous considérons donc non pas des patrons isolés mais des bases de patrons pour lesquelles seuls des graphes cohérents peuvent être construits.

Intégration d'une usine d'argumentation Nous envisageons d'instrumenter le portfolio en construisant automatiquement des diagrammes d'argumentation à partir d'une base de patrons. Le but est alors de notifier une *Usine d'Argumentation (UA)*, chaque fois que les évolutions du portfolio sont potentiellement soumises à l'argumentation. L'UA a alors la charge de construire et de gérer la cohérence des diagrammes d'argumentation.

La Figure 3 donne une vision globale de l'architecture proposée pour lier une UA au portfolio. Lorsque des modifications du porfolio sont sujettes à argumentation, un évènement est levé qui est interprété par l'UA qui comprend un moteur d'argumentation. Ce dernier transforme les événements du bus d'événements en éléments d'argumentation. Les pas d'argumentation sont ajoutés ou modifiés selon l'évolution du portfolio et la base de patrons. Les patrons guident la construction des pas et assurent la cohérence du graphe d'argumentation. Dans la partie supérieure gauche, un outil de visualisation des diagrammes d'argumentation (ADEV, pour Argumentation Diagrams Editor et Viewer) prend en charge les interactions homme-machine avec les diagrammes produits.

4 Conclusion

Dans cet article, nous avons présenté une approche qui vise à argumenter de manière incrémentielle la construction de systèmes basés sur des expérimentations. Nous l'avons expliquée

4. Par abus de langage nous utiliserons évidence pour *Evidences* et stratégie pour *Strategy*
5. Un exemple de diagramme est disponible à l'adresse suivante : http://bit.ly/2p1Y0YJ

au travers du projet ROCKFlows qui intègre un portfolio de workflows de ML. Cette approche est également utilisée dans le cadre d'une collaboration avec la société AXONIC autour de l'outil AVEK qui porte sur le suivi d'expérimentations dans un contexte bio-médical. Sur la dimension argumentation, nous poursuivons actuellement nos recherches selon deux axes : *(i)* la formalisation de l'argumentation, en particulier pour définir les propriétés des bases de patrons et assurer leur cohérence, *e.g.*, atteignabilité des objectifs, connexité ou non des graphes, *(ii)* l'utilisabilité de l'argumentation en lien avec les résultats d'expérimentations et la navigation dans ces très grands graphes. Ainsi, dans le cadre de ROCKFlows, la connexion de l'interface utilisateur au niveau de LPL aux graphes d'argumentation reste un sujet que nous n'avons pas encore exploré. En argumentant sur les expériences menées, nous souhaitons non seulement favoriser la construction d'une communauté mais également renforcer les règles qui sous-tendent la construction des portfolios.

Références

[1] Osman Balci. Verification, validation, and certification of modeling and simulation applications. In *Proceedings of the 35th Conference on Winter Simulation : Driving Innovation*, WSC '03, pages 150–158. Winter Simulation Conference, 2003.

[2] Victor R Basili, Gianluigi Caldiera, and H Dieter Rombach. Experience factory. *Encyclopedia of software engineering*, 1994.

[3] Pierre Bieber, Frédéric Boniol, Guy Durrieu, Olivier Poitou, Thomas Polacsek, Virginie Wiels, and Ghilaine Martinez. MIMOSA : Towards a model driven certification process. In *Proc. 8th Int. Congress on Embedded Real Time Software and Systems (ERTS'16)*, 2016.

[4] Bernd Bischl, Pascal Kerschke, Lars Kotthoff, Marius Thomas Lindauer, Yuri Malitsky, Alexandre Fréchette, Holger H Hoos, Frank Hutter, Kevin Leyton-Brown, Kevin Tierney, and Joaquin Vanschoren. ASlib : {A} benchmark library for algorithm selection. *Artif. Intell.*, 237 :41–58, 2016.

[5] C Camillieri, L Parisi, M Blay-Fornarino, F Precioso, M Riveill, and J Cancela Vaz. Towards a Software Product Line for Machine Learning Workflows : Focus on Supporting Evolution. In *Proc. 10th Work. Model. Evol. co-located with ACM/IEEE 19th Int. Conf. Model Driven Eng. Lang. Syst. MODELS 2016*, pages 65–70, Saint-Malo, France, October 2016.

[6] Manuel Fernández-Delgado, Eva Cernadas, Senén Barro, and Dinani Amorim. Do we Need Hundreds of Classifiers to Solve Real World Classification Problems ? *Journal of Machine Learning Research*, 15 :3133–3181, 2014.

[7] Kevin Leyton-Brown, Eugene Nudelman, Galen Andrew, Jim McFadden, and Yoav Shoham. A portfolio approach to algorithm select. In *Proceedings of the 18th International Joint Conference on Artificial Intelligence*, IJCAI'03, pages 1542–1543, San Francisco, CA, USA, 2003. Morgan Kaufmann Publishers Inc.

[8] Andreas Peldszus and Manfred Stede. From argument diagrams to argumentation mining in texts : A survey. *International Journal of Cognitive Informatics and Natural Intelligence (IJCINI)*, 7(1) :1–31, 2013.

[9] Thomas Polacsek. Validation, accreditation or certification : a new kind of diagram to provide confidence. In *Research Challenges in Information Science (RCIS), 2016 IEEE Tenth International Conference on*, pages 1–8. IEEE, 2016.

[10] Eric Ras, Jörg Rech, and Sebastian Weber. Knowledge services for experience factories. In Knut Hinkelmann and Holger Wache, editors, *Fifth Conference Professional Knowledge Management : Experiences and Visions, March 25-27, 2009 in Solothurn, Switzerland*, volume 145 of *LNI*, pages 232–241. GI, 2009.

[11] Hans-Jorg Rheinberger. Toward a history of epistemic things : Synthesizing proteins in the test tube. 1997.

[12] Stephen E Toulmin. *The uses of argument*. Cambridge University Press, 2003.

[13] Claes Wohlin, Per Runeson, Martin Höst, Magnus C Ohlsson, Björn Regnell, and Anders Wesslén. *Experimentation in software engineering*. Springer Science & Business Media, 2012.

[14] David H Wolpert. The lack of a priori distinctions between learning algorithms. *Neural computation*, 8(7) :1341–1390, 1996.