Improving Confidence in Experimental Systems through Automated Construction of Argumentation Diagrams

Clément Duffau[1,2], Cécile Camillieri[1] and Mireille Blay-Fornarino[1]
[1]Université Côte d'Azur, CNRS, I3S, France
[2]AXONIC, France
{

Keywords: Argumentation, Experiment System, Accreditation, Confidence, Software Engineering.

Abstract: Experimental and critical systems are two universes that are more and more tangling together in domains such as bio-technologies or aeronautics. Verification, Validation and Accreditation (VV&A) processes are an everyday issue for these domains and the large scale of experiments needed to work out a system leads to overhead issues. All internal V&V has to be documented and traced to ensure that confidence in the produced system is good enough for accreditation organism. This paper presents and proposes a practical approach to automate the construction of argumentation systems based on empirical studies in order to represent the reasoning and improve confidence in such systems. We illustrate our approach with two use cases, one in the biomedical field and the other one in machine learning workflow domain.

1 INTRODUCTION

When development of a system is based on results of experiments (Wohlin et al., 2012), the Verification and Validation (V&V) process focuses on measuring different variables, changing treatments and replaying experiments. During these investigations, quantitative and qualitative data is collected that is next analyze statistically to build new information. Knowledge is then deduced from computer-simulated cognitive process or from the transcripts of knowledge acquired by human actors (Chen et al., 2009).

An *experimental system* has been defined by Rheinberger (Rheinberger, 1997) as "a basic unit of experimental activity combining local, technical, instrumental, institutional, social, and epistemic aspects". The process of validating and argumenting on the confidence in an experimental system involves many reasoning elements, such as the way in which the statistical analyses were carried out or the people who took the decisions. All those elements, including their order, must be taken into account to improve confidence in the system and to produce summarized information if needed. This is even more important for critical industries which are costly involved in Verification, Validation, and Accreditation (VV&A) procedures (Balci, 2003). The accreditation process determines whether a system is reliable enough to be used. At each step of development of a system, the objective is to produce documents, archives, software, to ensure that the quality of the process is good enough for an external organization: the accreditation organization. In order for projects to go further, the accreditation process should begin as early as possible to provide all the information needed at each step to perform the accreditation assessment. Thus, for experimental systems in critical industries, traceability and documentation activities have to be done even more carefully during experiments than in common empirical studies.

For example, in the bio-medical domain, the international norm ISO 13485, applicable to medical devices, describes expected activities that must be followed at each level of the development process: initialization, input data, feasibility study, development, industrialization, validation, output data, marking request, commercialization. These stages have to be followed in this order and need to be completed in terms of the activities themselves but also accompanied with documents before being able to go further in the process. Documents may be software test results, hardware benchmark or the clinical protocols followed for the experiments.

Argumentation diagrams are a way to support the accreditation process (Polacsek, 2016). In critical systems, the need is not to construct such diagrams from the V&V results in a retrospective approach (Peldszus and Stede, 2013) but to build them throughout the development process, including during experiments.

Duffau, C., Camillieri, C. and Blay-Fornarino, M.
Improving Confidence in Experimental Systems through Automated Construction of Argumentation Diagrams.
DOI: 10.5220/0006358504950500
In *Proceedings of the 19th International Conference on Enterprise Information Systems (ICEIS 2017) - Volume 2*, pages 495-500
ISBN: 978-989-758-248-6
Copyright © 2017 by SCITEPRESS – Science and Technology Publications, Lda. All rights reserved

The remainder of this position paper is organized as follows. In the next section we discuss related work, introduce two case studies and determine objectives. Section 3 gives an overview of our approach and describes how it was applied to the case studies. Section 4 concludes the paper and briefly discusses future work.

2 ARGUMENTATION DIAGRAMS TO INCREASE CONFIDENCE IN EXPERIMENTAL SYSTEMS

In this section, we start by discussing related works, then we introduce two use cases and outline the research objectives being addressed by our approach.

2.1 Related Work

In (Basili et al., 1994), Basili describes how people in a project organization has to conduct experiences in an empirical way. An *Experience Factory (EF)* is then an infrastructure designed to support experience management in software organizations. While the initial definition refers to reuse of experiences regarding software products or IT projects, we generalize EF to any information system designed to support experience management. EF supports the collection (*Experimentation Software*), analysis *(Reasoning Software)* and packaging of experiences in an *Experience Base* as shown at the bottom of Figure 2 . While EF are interested in capitalizing on knowledge to find solutions to other problems later, we focus in this article on the certification of the approach followed in the analysis process.

In (Larrucea et al., 2016), the authors describe the tooling of an EF based on the process described in ISO/IEC29110[1] for certification purposes. The approach is based on verification of requirements. Our work is positioned differently, the objective is not the verification of requirements but the construction of the associated argumentation. Thus, for example, if we take a property ”Assets have their owner formally identified”, we do not try to verify wether it is satisfied or not, but rather how the property was obtained (*e.g.*, code analyzer).

In regards to critical domains, the EF is not sufficient for accreditation purposes. EFs support the collection of artifacts (*e.g.*, documents, statistics, experiment data, conclusions) used and resulting from experiments (Wohlin et al., 2012, Fig.6.5) but not the arguments supporting the link between them, which

[1]Standard for systems engineering, software engineering, and lifecycle processes for very small entities.

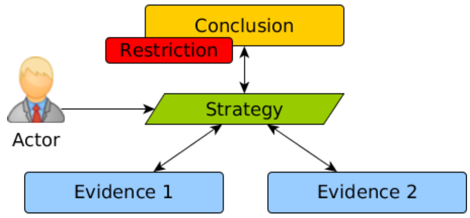

Figure 1: Model of an argumentation step.

are necessary in critical domains. In this context, there exists argumentation notation formalisms, however they can be applied only in an *a priori* approach. For example, a notation for structuring safety arguments has been identified in Kelly’s work (Kelly and Weaver, 2004). This notation allows to instrument the risk analysis to work out a system, but not experimental activities.

Argumentation is the process of convincing people of a conclusion based on proof elements. Formalizing argumentation can be a way to trace reasoning about experiments. Thus, the argumentation diagrams of Polascek’s approach (Polacsek, 2016) are derived from the argumentation model outlined by Toulmin (Toulmin, 2003) to take into account critical systems. The representation of an argumentation step in Figure 1 is based on these works. It shows that the *Actor* has followed a *Strategy* based on two *Evidences*, to determinate a *Conclusion* relatively to a *Restriction*. Argumentation steps can be chained thanks to the reuse of a conclusion as an evidence to another strategy. In (Rech and Ras, 2011), authors propose a similar representation but with a different purpose which is to formalize the knowledge in the EF. Thus we can make an analogy between strategies and actions, conclusions and benefits. The evidences are the same, while the concept of context in A2E includes the actor, evidences, and restrictions. The objective of argumentation and validation of this one explains, among other things, this difference.

Argumentation diagrams for accreditation are currently defined manually. But, as VV&A consists of multiple activities leading to complex argumentation diagrams, the construction has to be automated for empirical and critical domains so as to scale for an organization. Creating a link between argumentation building and system production process including experiments and V&V will increase confidence in the system and help to detect conception defects earlier, as suggested in (Rus et al., 2002).

The research question asked by this is how to automatically report argumentation elements during the experiments and their analysis and construct argumentation diagrams from these elements.

2.2 Use Cases and Motivation

We motivate the necessity to deal with automatic building of argumentation diagrams using two real-world case studies. The first relates to medical experiments while the second relates to the construction of a portfolio of Machine Learning workflows.

AXONIC: A Bio-Medical Use Case. The experimentation software (ASF) and the reasoning software (AVEK) are developed by the AXONIC company. This company is focused on neurostimulation to treat pathologies linked to the nervous system (*e.g.*, epilepsy, chronic pain, obesity). Thanks to a specific stimulation (an electric waveform) injected by a stimulator on a nerve, the pathology is addressed. The stimulator can be on table, portable or implantable. To assess that kind of medical devices, domain experts need to perform experiments following a protocol established by clinicians, according to medical authorities. At each step of the process, experts collect data, create or improve knowledge. In this context, activities are mainly focused on human expertise, even when dealing with statistical analysis. Thus, people and the confidence in their work is very important. The accreditation process is based on a large quantity of documents produced during all stages of experiments. External tools exist to summarize the results but their exploitation is still a complex task, while arguing on the followed process and experiments results are critical to the final product production. In this context, we aim to build our argumentation at each step of the system construction in an automatic way by integrating artifacts form experiments as well as support like articles from the state of the art and mathematical models used to reason. Such an argumentation tooling for each step of this process is more and more crucial and strategic for the AXONIC company.

ROCKFlows: Automated Construction of Machine Learning Workflows. In Machine Learning (ML), data scientists know that the best algorithm is not be the same for each problem (Wolpert, 1996). Finding a good algorithm depends at least on the size, quality, nature of the data and on what we want to do with the data. According to user dataset and objectives, ROCKFlows aims to generate the most suitable ML workflow, depending on the problem to be solved (Camillieri et al., 2016).

To tackle this problem, the approach of ROCKFlows is to build a portfolio (Leyton-Brown et al., 2003) of ML workflows, associated with a Software Product Line (SPL). It is built from the results of experiments and their analysis and helps to find the best algorithms for a task.

For the end user, the workflow that is proposed by the SPL to solve her problem is provided by a black box. However it is still essential to be able to explicit the approach, to validate it, to enable reasoning on it and to expose it in a simple way. In this context, the use of an argumentation solution in ROCKFlows brings benefits to different actors: 1) non expert users who want explanations on workflows ranking in the platform to trust its choices, 2) data-mining experts who want to analyze if the choice of a workflow properly takes in account of the specificities of their problem, 3) contributors who want to understand how the platform is built before extending it by adding datasets, algorithms or improving evaluation strategy, algorithms parameters.

Building ROCKFlows' portfolio is a heavy process that involves a high amount of experiments with different datasets, algorithms, parameters, etc. Thus, constructing a global argumentation diagram requires an automated process to handle scaling, ensure the quality of the diagram and its automated update. Indeed, new algorithms are regularly proposed by data scientists for dealing with more or less specific problems and improving performances and accuracy, leading to an ever-increasing and evolving knowledge. Moreover, the reasoning process itself is still subject to change.

2.3 Objectives

Thanks to these uses cases, we identify the three following objectives, we want to achieve.
(c1) Incremental, automatic and seamless construction of argumentation diagrams. When data collection and analysis are conducted incrementally (*e.g.*, adding new steps of experiments analysis in AVEK or adding new algorithms in ROCKFlows), confidence and consequently argumentation may be developed in small increments rather than one large piece. To perform an experiment, several steps have to be taken. According to the granularity of the data collection and analysis, a great number of increments should be taken into account. Consequently, automatic building of argumentation diagrams is needed to tackle the big number of steps. While EFs must be frequently updated and adapted, seamless integration between an EF and argumentation management is essential.
(c2) Controlling consistency of the experiment process at each step of argumentation. It comes to construct the argumentation on the basis of the process which had been defined (*e.g.*, in AVEK, we have to ensure that the process steps have been followed and the proper artifacts have been provided).
(c3) Usability of the argumentation diagram. Building

argumentation diagrams and ensuring their consistency is not enough. In critical contexts this argumentation has to be reviewed, discussed, validated and presented to an audit. Thus, business experts should be able to read them easily, in particular by navigating between the arguments and the experiment artifacts.Depending on the granularity chosen to construct the argumentation diagrams, they can be very large. The interactions with these diagrams must therefore allow different points of view.

Below, we present the approach we propose to achieve these objectives.

3 AUTOMATED CONSTRUCTION OF ARGUMENTATION DIAGRAMS

We propose to instrument the experimentation process by automatically building argumentation diagrams. For this, we consider the EF representation for experimental systems: *i.e.*, "a separate organization that supports product development by analyzing and synthesizing all kinds of experiences, acting as a repository for such experience" (Wohlin et al., 2012).

The purpose is to tool the EF to notify an argumentation engine each time that the information received or reasoning steps are subject to argumentation. This includes evidences, strategies and conclusions.

3.1 Argumentation Factory on Top of the Experience Factory

Figure 2 gives an overview of the proposed architecture to relate an argumentation diagram with an EF. At the top right part of figure 2, the *argumentation factory* manages argumentation diagrams. At the top left part, a GUI tool (ADEV, for Argumentation Diagrams Editor and Viewer) supports interaction with the argumentation diagrams. Experience and argumentation factories communicate through an event bus that allows lowly coupled interactions between these two components. The argumentation engine transforms events from the event bus in argumentation elements. These elements conform to an argumentation meta-model based on Polacsek's work (Polacsek, 2016). Figure 3 shows an excerpt of the core of this meta-model. Argumentation steps are added or modified incrementally according to the evolution of the experience base. Constraints on the argumentation diagram should verify the rules of consistency imposed by the experiment process. This important part of the system lies on the argumentation meta-model and argumentation engine. ADEV, the GUI tool for the argumentation diagram is based on the argumentation meta-model.

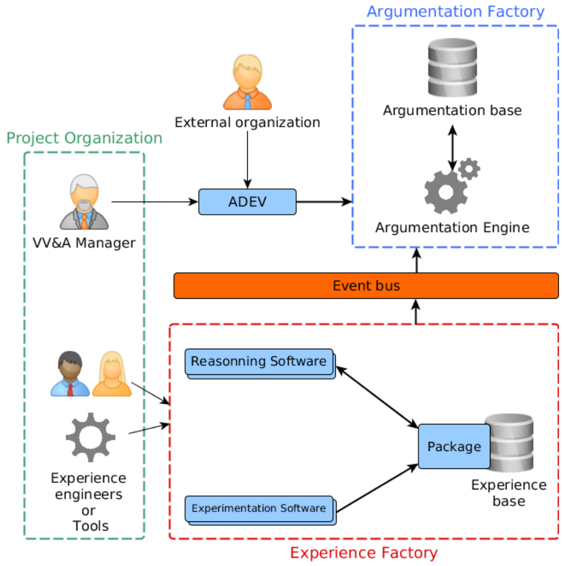

Figure 2: Architecture between argumentation factory and experience factory.

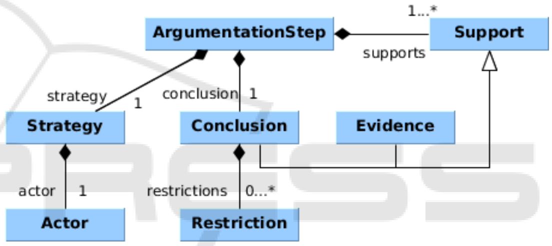

Figure 3: Meta-model excerpt of an argumentation step.

The bus and argumentation engine aim to support incremental and automatic building or modification of argumentation diagrams (c1). The argumentation engine has the responsibility to check the consistency of the experimentation studies (c2). The third challenge (c3) is achieved by ADEV by presenting to the user the argumentation diagrams built automatically. Thanks to ADEV, a quality manager can interact (viewing and cosmetic editing) with these argumentation diagrams in order to confront reality to the theoretical good practices and to present in accreditation audit. To deal with large argumentation diagrams, a common sub-graphing approach has been developed in ADEV to hide parts of argumentation diagram depending on the purpose of the user. Obviously, navigation between parent diagram and sub-diagram is possible.

The aim is then to notify the argumentation factory each time the experience base is modified. Two main steps are then needed to tool an experimental system.

1. Linking argumentation elements to the experiments and their analysis;

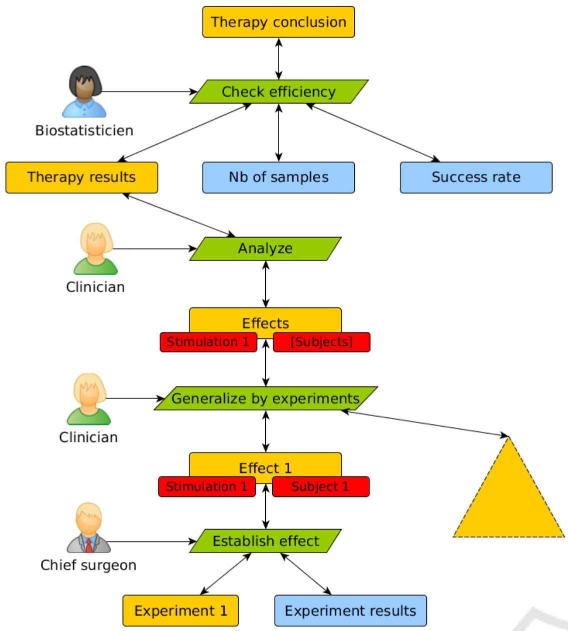

Figure 4: Excerpt of an argumentation diagram in the AXONIC use case.

2. Automatically constructing argumentation diagrams from experimental system notifications and checking consistency of the process.

We now describe how we handle these two aspects in our case studies.

3.2 Linking Argumentation Value with the Experiment Process

AVEK: An Instrumented Dedicated Tool to Notify Argumentation Factory. In AXONIC software ecosystem, the difficulty for us in this context is to deal with the evolutivity of ASF and AVEK. Thanks to the meta-data approach for argumentation elements, we just had to decorate the neurostimulation model of AXONIC to support argumentation and send notifications to the event bus.

As the analyses are achieved by humans, the argumentation diagram evolves, in particular when business experts add their analyses in the system.

ROCKFlows: From a Base of Experiments to an Evolving Argumentation Diagram. As the experience base already existed at the time we decided to associate it an argumentation diagram, we began by modeling an abstract argumentation diagram representing the relations between data set, algorithms and experimental results subject to argumentation. We then automated the construction of the diagram by an analysis of our experience base. A challenge in

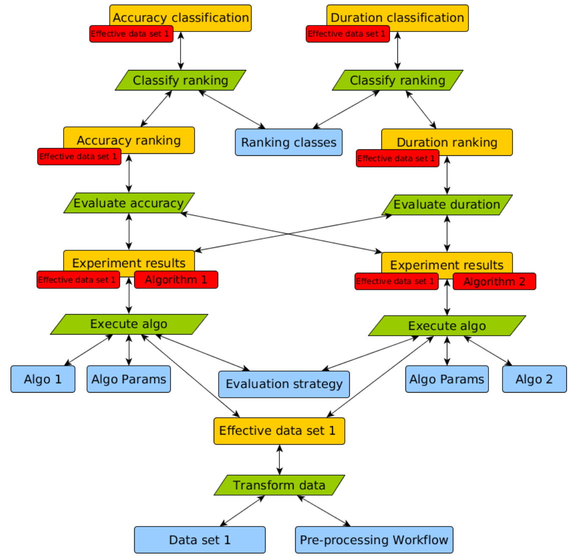

Figure 5: Excerpt of an argumentation diagram in ROCKFlows for one dataset.

ROCKFlows is to handle the evolution of this diagram, because knowledge such as the ranking of algorithms can be revised as new elements are added in the system. For this purpose, we had to instrument the services and daemons that support the evolution of the experience base. They now notify the bus of each change considered to be part of the argumentation, allowing to be fully transparent for the ROCKFlows ecosystem.

3.3 Argumentation Diagrams Construction from Experimental System Notifications

AVEK: Process Driven Argumentation Diagram Construction. Clinical studies have to follow a sequential process where the validation of the previous activity is the trigger of the next one. The validation is not just the notion of a step being done, it also states if the results are sufficient to go further in the process. By outfitting the experimental legacy system with argumentation tools, it is easier to determinate when the results are fitting the objectives and go to the next step. This incremental approach of argumentation on the system is not sustainable with argumentation diagrams constructed only by humans. The automatic building thanks to AVEK is a key for scaling in this purpose. A very basic example is shown in Figure 4. In this example, we deal with only one experiment on a single patient with one stimulation to achieve a goal. This is the minimal case for a study. Scaling issues are highlighted with this example. This aspect surfaces issues on the evolutivity of argumentation diagrams. How to deal with argumentation steps that are incrementally added to a diagram? What is the impact on the anterior argumentation?

ROCKFlows: Evolution Driven Argumentation Diagram Construction. The argumentation diagram for this use case is shown in Figure 5. This example shows an argumentation diagram for *ranking* on *accuracy* and *duration* metrics for two *algorithms* on one *dataset*. From notifications such as "a new experiment was added", we build a new argumentation step with the strategy *execute algo* whose inputs are the data used by the experiment and outputs are references to the results. As the conclusion of this steps needs to be used in already existing steps, the argumentation engine has to check the consistency of the diagram (*e.g.*, the *classify ranking* steps must take into account all the measurements resulting from a same *evaluation strategy*).

Today, in ROCKFlows, more than 100 datasets and 60 algorithms are analyzed together and ranked according to different metrics. Thus, the automatic construction of the argumentation diagram helps us to scale the massive argumentation of this use case.

4 CONCLUSION

In this position paper, we propose to automatically build argumentation according to the evolution of an Experience Factory. Through two applications we have shown how the approach, which involves an argumentation factory, can be applied. These two applications consist of surveys of experiments in a bio-medical context and in a portfolio of ML algorithms. In the short term we are going to improve the proposed approach along the following lines. At the level of the argumentation engine we work to take into account argumentation patterns and constraints between them. At the level of the event bus, we are interested in evolving the notion of events to better capture the information coming from experiments using formalisms such as the ExpML language (Vanschoren et al., 2012). At the level of the interaction with the argumentation diagrams, we will go through the evaluation phase of ADEV with end-users.

As future work, we plan to extend the approach to support large-scale lifecycle development, staging between each critical step (*e.g.* airplane simulations, prototyping, production). In the longer term, we intend to manage the history of the argument diagrams by the possibility of managing different versions.

REFERENCES

Balci, O. (2003). Verification, validation, and certification of modeling and simulation applications. In *Proc. of the 35th Conf. on Winter Simulation: Driving Innovation*.

Basili, V. R., Caldiera, G., and Rombach, H. D. (1994). Experience factory. *Encyclopedia of software engineering*.

Camillieri, C., Parisi, L., Blay-Fornarino, M., Precioso, F., Riveill, M., and Cancela Vaz, J. (2016). 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)*.

Chen, M., Ebert, D., Hagen, H., Laramee, R. S., van Liere, R., Ma, K.-L., Ribarsky, W., Scheuermann, G., and Silver, D. (2009). Data, information, and knowledge in visualization. *IEEE Comput. Graph. Appl.*, 29.

Kelly, T. and Weaver, R. (2004). The goal structuring notation–a safety argument notation. In *Proc. of the dependable systems and networks 2004 workshop on assurance cases*. Citeseer.

Larrucea, X., Santamaría, I., and Colomo-Palacios, R. (2016). Assessing iso/iec29110 by means of itmark: results from an experience factory. *Journal of Software: Evolution and Process*, 28(11):969–980.

Leyton-Brown, K., Nudelman, E., Andrew, G., McFadden, J., and Shoham, Y. (2003). A portfolio approach to algorithm select. In *Proc. of the 18th Int. Joint Conference on Artificial Intelligence*. Morgan Kaufmann Publishers Inc.

Peldszus, A. and Stede, M. (2013). From argument diagrams to argumentation mining in texts: A survey. *Int. Journal of Cognitive Informatics and Natural Intelligence*.

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

Rech, J. and Ras, E. (2011). Aggregation of experiences in experience factories into software patterns. *SIGSOFT*.

Rheinberger, H.-J. (1997). Toward a history of epistemic things: Synthesizing proteins in the test tube.

Rus, I., Biffl, S., and Halling, M. (2002). Systematically combining process simulation and empirical data in support of decision analysis in software development. In *Proc. of the 14th Int. Conf. on Software engineering and knowledge engineering*.

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

Vanschoren, J., Blockeel, H., Pfahringer, B., and Holmes, G. (2012). Experiment databases. *Machine Learning*.

Wohlin, C., Runeson, P., Höst, M., Ohlsson, M. C., Regnell, B., and Wesslén, A. (2012). *Experimentation in software engineering*. Springer Science & Business Media.

Wolpert, D. H. (1996). The lack of a priori distinctions between learning algorithms. *Neural computation*.