Yes, Configuring is Good, But Have You Ever Tried Justifying?

Sébastien Mosser[1], Corinne Pulgar[2], Mireille Blay-Fornarino[3],
Deesha Patel[1], Aaron Loh[1], Jean-Michel Bruel[4]

(1) McMaster University, McSCert Lab, Hamilton, Canada
(2) École de Technologie Supérieure (ETS), Montréal, Canada
(3) Université Côte d'Azur, I3S Lab (CNRS), Sophia Antipolis, France
(4) Université de Toulouse, IRIT Lab (CNRS), Toulouse, France

Abstract: *(590/600 words)*

Considering any complex system, reaching a correct configuration is challenging, and configuration errors might have dramatic consequences. In October 2021, *Meta* encountered a DNS issue, leading to a 6-hour outage that impacted *Facebook*, *Instagram*, and *WhatsApp* [2]. A few months earlier, *Orange* upgraded all their call servers simultaneously. It triggered a massive outage of the 112 service (European equivalent of 911) for 8 hours, playing a role in the death of six persons [3]. More recently, *Rogers* encountered a 15-hour outage that paralyzed Canada: 25% of the country was disconnected from the internet, the *Interac* banking system was down, and 911 was disrupted [4].

In all these situations, a configuration error was the problem. Interestingly, these configurations error were "accidents" that could have been prevented if the engineers had the right prior knowledge of the legacy system. Focusing on the *Rogers* outage, an apparently useless traffic filtering rule was deleted on the routers. The lack of filtering for incoming traffic triggered a domino effect on all the routers, flooding *Rogers'* core network.

So, the real cause was the "**change**" in configuration, which was not compliant with "**why**" the system was in its current state of configuration.

When documenting their configurations, it is up to the engineer in charge to describe not only the choice ("*Here is a traffic filtering rule*") but also its justification ("*avoiding network flooding by regulating core traffic*"). One can, for example, use an *Architecture Decision Record* (ADR) templates approach to force the expression of assumptions, constraints, arguments, and implications of the configuration choice [5]. More formal approaches from the requirements engineering community can be used to link configuration choices with semantic justifications, such as *Justification Diagrams* (JDs [6]) or goal models [7]. JDs were successfully used to justify the configuration of building systems for safety-critical medical devices [8], and, more recently, to justify DevOps pipelines at a larger scale [9].

A JD relies on Toulmin's argumentation model [10] to reach a conclusion by assembling strategies that consume factual evidence or intermediate sub-conclusions to produce new conclusions. As such, when used to document a given configuration, it forces the definition of an argumentation that justifies why the system is in its current state. Unfortunately, so far, this kind of artifact is only

used as documentation and is an additional burden for the engineer (co-evolving with the configuration).

To tackle this challenge, we propose an extension to the JD model, implemented as an operational domain-specific language. In addition to the evidence, strategy, and conclusion elements, we enriched the syntax by providing probes, operations and expectations associated with the justified elements. As such, one can attach to the strategy describing the router configuration deployment an expectation stating that the filtering rule must be present in the configuration and, if not, reject the deployment as being not compliant with its justification. As such, the justification becomes an operational artifact that supports the configuration.

We are currently developing a textual version of the language in the context of an industrial partnership with a major telco operator and an SME specialized in software-defined networks. We are currently focusing our efforts on how the justification can be leveraged by analyzing existing configurations of build systems and exploring how such configuration has evolved over the years. To respect NDA/IP concerns, we also apply our approach to open-source case studies, such as how Docker-compose continuous integration/continuous deployment (CI/CD) configuration has evolved. This talk will describe the language, its current (preliminary) implementation, and how it can be leveraged to support the configuration of such CI/CD systems.

References:

[1] Acher, Mathieu, et al. "Feature subset selection for learning huge configuration spaces: the case of Linux kernel size." *Proceedings of the 26th ACM International Systems and Software Product Line Conference-Volume A*. 2022.

[2] https://en.wikipedia.org/wiki/2021_Facebook_outage (access: 07/05/23)

[3] (***French***) *ÉVALUATION DE LA GESTION PAR L'OPÉRATEUR ORANGE DE LA PANNE DU 2 JUIN ET DE SES CONSÉQUENCES SUR L'ACCÈS AUX SERVICES D'URGENCE, 07/19/2021 (https://www.economie.gouv.fr/files/2021-07/Rapport-Orange-SNU.PDF)*

[4] https://en.wikipedia.org/wiki/2022_Rogers_Communications_outage (access: 07/05/23)

[5] Van Heesch, Uwe, et al. "A documentation framework for architecture decisions." *Journal of Systems and Software* 85.4 (2012): 795-820.

[6] Polacsek, Thomas, et al. "The need of diagrams based on Toulmin schema application: an aeronautical case study." *EURO Journal on Decision Processes* 6.3-4 (2018): 257-282

[7] Amyot, Daniel, et al. "Evaluating goal models within the goal-oriented requirement language." *International Journal of Intelligent Systems* 25.8 (2010): 841-877.

[8] Duffau, Clément, et al. "Support of justification elicitation: Two industrial reports." *Advanced Information Systems Engineering: 30th International Conference, CAiSE, Tallinn, Estonia, Proceedings 30*. Springer International Publishing, 2018.

[9] Pulgar, Corinne. "Eat your own DevOps: a model driven approach to justify continuous integration pipelines." *Proceedings of the 25th International Conference on Model Driven Engineering Languages and Systems: Companion Proceedings*. 2022.

[10] Toulmin, Stephen E. *The uses of argument*. Cambridge university press, 2003.