



HAL
open science

Le projet Pulse : vers la supervision des échanges dans un système IoT

Jannik Laval, Boubou Thiam Niang, Imene Ghzaiel, Kenza Riahi, Baudouin Dafflon, Giacomo Kahn, Yacine Ouzrout

► **To cite this version:**

Jannik Laval, Boubou Thiam Niang, Imene Ghzaiel, Kenza Riahi, Baudouin Dafflon, et al.. Le projet Pulse : vers la supervision des échanges dans un système IoT. CONGRES INFORSID -Atelier 2 : Évolution des SI : vers des SI pervasifs ? (INFORSID 2021), Jun 2021, Dijon, France. <hal-03250112>

HAL Id: hal-03250112

<https://hal.science/hal-03250112v1>

Submitted on 4 Jun 2021

HAL is a multi-disciplinary open access archive for the deposit and dissemination of scientific research documents, whether they are published or not. The documents may come from teaching and research institutions in France or abroad, or from public or private research centers.

L'archive ouverte pluridisciplinaire **HAL**, est destinée au dépôt et à la diffusion de documents scientifiques de niveau recherche, publiés ou non, émanant des établissements d'enseignement et de recherche français ou étrangers, des laboratoires publics ou privés.



HAL Authorization

Le projet Pulse : vers la supervision des échanges dans un système IoT

Jannik Laval¹, Boubou Thiam Niang¹, Imene Ghzaïel¹,
Kenza Riahi¹, Baudouin Dafflon², Giacomo Kahn¹,
Yacine Ouzrout¹

Univ Lyon, Univ Lumière Lyon 2, INSA Lyon, Université Claude Bernard Lyon 1,
DISP, EA4570, 69676 Bron, France
prenom.nom@univ-lyon2.fr

ABSTRACT. This article describes our work on monitoring data exchanges in an IoT information system. We address the collection, modelling and processing of data in order to analyse the evolution of the system architecture and identify the problems that may arise.

RÉSUMÉ. Cet article décrit nos travaux dans le domaine de la supervision des échanges de données dans un système d'information IoT. Nous abordons la collecte, la modélisation et le traitement des données afin d'analyser l'évolution de l'architecture du système et d'identifier les problèmes qui peuvent survenir.

KEYWORDS: Data Monitoring, IoT System, Dynamic analysis of IoT, Interoperability Assessment.

MOTS-CLÉS : Surveillance des données, système IoT, analyse dynamique de l'IoT, évaluation de l'interopérabilité.

© Lavoisier

1. Introduction

Les architectures orientées événements (EDA) (Sriraman, Radhakrishnan, 2005) permettent un couplage lâche, mais limitent la capacité de compréhension du comportement global de l'architecture. Cette limitation est due à la multiplicité des échanges de données. Par conséquent, il est nécessaire de mettre en place des systèmes de surveillance et d'analyse pour mieux comprendre le fonctionnement et l'évolution en temps réel du système d'échanges (Brand, Giese, 2019). Il existe, à ce jour, des plateformes proposant des consoles de supervision. Cependant, la plupart de ces solutions traitent principalement des informations de bas niveau, telles que la fréquence des messages échangés ou l'utilisation de la mémoire. Le but du travail de recherche présenté dans cet article est de proposer une solution permettant de collecter les informations et les messages provenant des systèmes IoT afin de les analyser. La supervision des systèmes IoT implique d'intégrer l'aspect dynamique de leur architecture

et la gestion de l'imprévu, tels que les problèmes de connexion ou de sécurité. La collecte d'informations provenant du système d'une part et des messages d'autre part est une première étape pour analyser un système IoT en fonctionnement.

2. Le framework Pulse

Un système de surveillance devrait fournir des moyens permettant de maintenir un niveau d'interopérabilité conforme aux exigences d'interopérabilité (Mallek *et al.*, 2010). Dans ce sens, nous proposons une approche permettant d'effectuer une supervision avancée en combinant les informations de l'architecture et les métadonnées des messages échangés. Ainsi, cela permet de produire les indicateurs nécessaires à la détermination des actions de maintenance à effectuer sur le système. Pour cela, nous proposons le *framework* Pulse (Figure 1) qui est composé de quatre niveaux : l'importation des données, la gestion du temps et des modifications de l'architecture, la persistance et l'analyse des données.

Un prototype du *framework* est déjà opérationnel ¹ et permet de visualiser certains comportements du système. Il facilite ainsi l'interprétation du statut de chaque message et fournit des indicateurs pour l'analyse de l'interopérabilité. Le prototype est implémenté sur Moose, une plateforme d'analyse de données open source (Demeyer *et al.*, 2001; Ducasse *et al.*, 2009), dans laquelle le métamodèle Pulse est intégré.

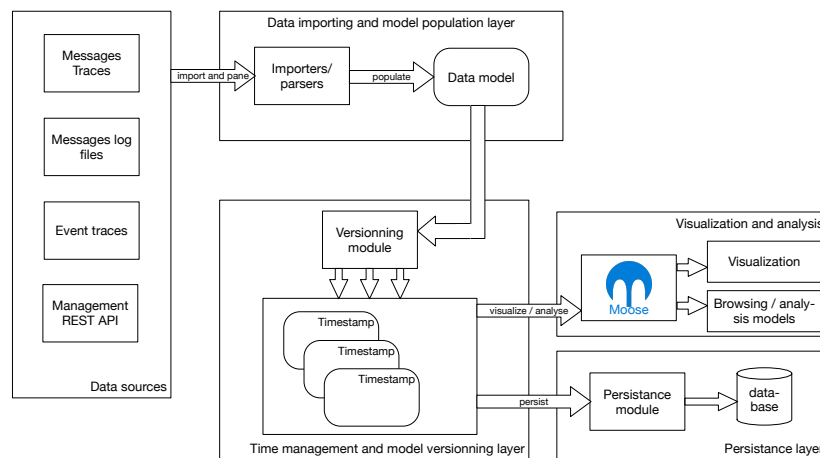


Figure 1. Le framework Pulse.

Le niveau "importation" collecte les métadonnées des messages transitant dans le système IoT et reconstitue l'architecture. Ces métadonnées permettent de suivre les messages échangés, d'identifier les entités actives, et d'améliorer la maintenabilité grâce à la détection des exceptions. Les métadonnées collectées sont organisées

1. <https://github.com/janniklaval/pulse>

pour être traitées de manière cohérente dans un métamodèle générique et extensible, présenté en Section 3.

Le niveau “gestion du temps” permet de traiter trois aspects du système : une représentation statique, considérant notamment les entités et les canaux par lesquels transitent les messages ; une représentation dynamique, qui concerne le cycle de vie des messages (connexions, messages et toute entité volatile) ; et la représentation du cycle de vie de l’architecture, où les entités sont créées, modifiées et supprimées. En conséquence, la structure Pulse intègre des fonctionnalités de modélisation dynamique de l’architecture en incluant un historique contenant les dates de création et de suppression des composants. Cette gestion du temps est implémentée avec Orion (Laval *et al.*, 2011), un outil interactif de prototypage pour la réingénierie.

Le niveau “analyse” permet de définir un ensemble d’indicateurs qui évaluent l’état du système au cours du temps ou à un instant donné pour visualiser et analyser les modifications et leur impact. Ces deux fonctionnalités nous permettent non seulement de détecter les problèmes au moment où ils se produisent, mais aussi d’identifier la source potentielle d’un problème en revenant à des états antérieurs.

3. Le Métamodèle Pulse

L’objectif du métamodèle Pulse est de représenter les trois aspects du système : la structure statique, le cycle de vie des messages et le cycle de vie de l’architecture.

Plusieurs protocoles peuvent être utilisés dans un même système d’information, un des objectifs de ce métamodèle est d’être suffisamment générique pour prendre en compte différents protocoles, et d’être extensible. Ce métamodèle est capable de prendre en charge les protocoles AMQP, MQTT, KAFKA et CoAP (Amokrane *et al.*, 2018; Laval *et al.*, 2020; Laval, 2021). Nous travaillons actuellement à son adaptation au *middleware* ROS (*Robot Operating System*). La Figure 2 illustre la structure actuelle du métamodèle.

Il rassemble des informations provenant de plusieurs sources. Si une console de gestion existe, les traces de messages fournies par cette console, cela permet d’identifier pour chaque message :

- Le nœud par lequel il transite, ainsi que les informations de connexion et d’hôte.
- L’échange dans lequel il a été publié ou consommé, les files d’attente vers lesquelles il est acheminé ou la file d’attente depuis laquelle il est consommé, ainsi que les clés de routage associées.
- L’utilisateur qui publie ou consomme le message.
- Son horodatage, son type (publié / reçu) et son mode de livraison (persistant ou non).

Les événements de création et de suppression de ressources, d’hôtes, d’utilisateurs et de permissions ; la création et la fermeture de connexions et de canaux et les tentative d’authentification de l’utilisateur peuvent également être récupérés par l’analyse des

4. Défis et verrous

Le prototype présenté montre la faisabilité et l'intérêt de notre approche. Son implémentation permettant d'analyser des systèmes AMQP d'une entreprise partenaire nous encourage à l'étendre vers d'autres architectures orientées événements. Les perspectives sont de plusieurs ordres, énoncés ci-après.

– Aide à la conception : Le métamodèle Pulse peut être utilisé dans la phase de conception des systèmes. L'analyse des échanges de données peut permettre d'extraire les caractéristiques des connecteurs permettant aux nouveaux constituants de se connecter au système d'échange. Les connecteurs sont des entités de première classe de l'architecture qui décrivent la sémantique de communication entre des constituants hétérogènes (Kutsche *et al.*, 2008). Ils sont donc des constituants à part entières avec des fonctionnalités leur permettant d'établir la communication entre d'autres constituants du système, par exemple le filtrage des informations sensibles ou la traduction du format du message dans un autre format. Dans ce sens, nous comptons utiliser Pulse dans un projet de configuration et de génération automatique des connecteurs permettant d'intégrer une entité dans un système d'information. Dans cette proposition, un modèle du connecteur est généré à partir d'une configuration du *feature model* représentant l'ensemble des fonctionnalités d'un connecteur désiré. Ainsi en plus de l'aspect données, le métamodèle sera étendu pour tenir compte de la structure du *feature model* et des fonctionnalités que ce dernier représente. D'autre part, les caractéristiques de chaque connecteur varient en fonction du besoin de communication: mode synchrone ou asynchrone, type de transport point-à-point ou *publish/subscribe*, etc. De plus les connecteurs doivent être générés automatiquement. Il faudra s'assurer que cette extension sera applicable dans un contexte évolutif et ouvert en tenant en compte des fonctionnalités variables du connecteur.

– Vérification de la qualité des échanges : obtenir des mesures quantitatives sur l'échange de données en temps réel nous permet d'élaborer une approche d'évaluation automatique de la fiabilité du système. Nous travaillons à définir un ensemble d'indicateurs en prenant en compte les caractéristiques et l'architecture du système IoT et les mesures associées issues de la norme (ISO/IEC 30141, 2018). Nous ne gardons que les indicateurs les plus pertinents et mesurables pour les associer adéquatement aux problèmes d'interopérabilité. Différentes techniques d'analyse peuvent être adoptées pour identifier les causes potentielles des problèmes par exemple les arbres de décision (Ringwald *et al.*, 2006) ou les méthodes à base de modèles (Khan *et al.*, 2010). L'objectif est d'utiliser le *framework* Pulse pour quantifier les échanges en temps réel et appliquer des algorithmes permettant de contrôler l'état du système en fonctionnement, mais également de fiabiliser la phase de conception.

– Simulation et banc de tests : la simulation d'un système IoT et de ses échanges est intéressante pour rendre plus rapides la conception et la maintenance par les tests. La simulation peut être complète ou hybride en intégrant du matériel dans la boucle (Hardware in the Loop - HIL). Elle permet de tester des scénarios difficilement testable en situation réel, dangereux ou trop coûteux.

Notre objectif est de pouvoir définir des scénarios, contraintes et métriques afin de mettre en évidence les points de ruptures des algorithmes évalués. Lors de ces simulations, nous apportons un intérêt particulier à l'étude des anomalies temporaires ou définitives et à la réponse que le système adoptera pour y faire face. Pour assurer la capacité d'exploration du simulateur, un ensemble de fichiers de configurations a été standardisé et documenté. Les travaux actuels permettent d'instancier un ensemble de capteurs réels et/ou virtuels interagissant ensemble. Le format de données envoyées vers le système IoT a été standardisé pour tous les capteurs afin d'assurer une forte interopérabilité. Pour garder une cohérence entre les capteurs du banc de test et les capteurs virtuels, la génération des données produites par ces capteurs s'appuie sur des algorithmes d'apprentissage automatique profonds. Ces modèles permettent de générer des données exploitables semblables à des données provenant de capteurs réels.

– Sûreté de fonctionnement du système : l'analyse de fiabilité et de la sûreté de fonctionnement peut être complétée avec les données de suivi. Nous envisageons le développement de pronostics de défaillance (*Prognostics and Health Management*) du système IoT. Le pronostic de défaillance, bien connu dans le domaine de la mécanique (Atamuradov *et al.*, 2017), transforme un ensemble de données recueillies sur l'équipement surveillé en indicateurs de santé du système, extrapolés dans le temps pour aider à sa maintenance. Nous envisageons de développer ce type de solutions sur les systèmes cyberphysiques. En couplant le *framework* Pulse avec des approches de pronostics de défaillance, nous espérons obtenir une gestion fine du cycle de vie des composants du système.

5. Conclusion

Les travaux menés dans l'analyse des données d'un système d'échanges de type EDA nous montrent que les systèmes IoT, bien que très flexibles, n'offrent pas de moyens de supervision avancés. Nos travaux s'attachent à aborder ces problèmes en considérant l'ensemble du cycle de vie du système (conception, analyse en fonctionnement, maintenance). Le *framework* Pulse et le métamodèle proposés ainsi que son prototype nous offre une base solide pour envisager d'aborder les défis présentés dans cet article. Chacun de ces défis est actuellement en cours d'investigation et lèvera sans doute de nouveaux verrous.

Dans cet article, nous n'avons pas considéré les verrous quant à l'exploitation de notre système. Notamment, l'un d'eux est que l'approche est actuellement centralisée. Or, nous pourrions envisager le *framework* comme un environnement de type système de systèmes. Ce changement d'architecture permettrait de passer à l'échelle grâce à une répartition de la charge de traitement des données. En même temps, des algorithmes de fusion de données ou de fusion de graphes devront être développés pour obtenir de cette supervision distribuée, une vision globale du système.

References

- Amokrane N., Laval J., Lanco P., Derras M., Moala N. (2018). Analysis of data exchanges, contribution to data interoperability assessment. In *2018 international conference on intelligent systems (is)*, p. 199-208.
- Atamuradov V., Medjaher K., Dersin P., Lamoureux B., Zerhouni N. (2017). Prognostics and health management for maintenance practitioners-review, implementation and tools evaluation. *International Journal of Prognostics and Health Management*, Vol. 8, No. 060, pp. 1–31.
- Brand T., Giese H. (2019). Generic adaptive monitoring based on executed architecture runtime model queries and events. In *2019 IEEE 13th international conference on self-adaptive and self-organizing systems (saso)*, p. 17-22.
- Demeyer S., Tichelaar S., Ducasse S. (2001). *FAMIX 2.1—the FAMOOS information exchange model*. Technical report, University of Bern.
- Ducasse S., Girba T., Kuhn A., Renggli L. (2009). Meta-environment and executable meta-language using smalltalk: an experience report. *Software & Systems Modeling*, Vol. 8, No. 1, pp. 5–19.
- Internet of Things (IoT) – Reference architecture*. International Standard. (2018). International Organization for Standardization.
- Khan M. M. H., Le H. K., LeMay M., Moinzadeh P., Wang L., Yang Y. *et al.* (2010). Diagnostic powertracing for sensor node failure analysis. In *Proceedings of the 9th ACM/IEEE international conference on information processing in sensor networks*, pp. 117–128.
- Kutsche R., Milanovic N., Bauhoff G., Baum T., Carlsburg M., Kumpe D. *et al.* (2008). Bizycle: Model-based interoperability platform for software and data integration. *Proceedings of the MDTPI at ECMDA*, Vol. 430.
- Laval J. (2021). The pulse project: a framework for supervising data exchanges in an IoT system. *Automation, Robotics & Communications for Industry 4.0*, pp. 74.
- Laval J., Amokrane N., Derras M., Moalla N. (2020, Nov). Analysis of data exchange among heterogeneous IoT systems. In *10th international conference on interoperability for enterprise systems and applications, Tarbes*.
- Laval J., Denier S., Ducasse S., Falleri J.-R. (2011). Supporting simultaneous versions for software evolution assessment. *Science of Computer Programming*, Vol. 76, No. 12, pp. 1177–1193.
- Mallek S., Daclin N., Chapurlat V. (2010). Towards a conceptualisation of interoperability requirements. In K. Popplewell, J. Harding, R. Poler, R. Chalmers (Eds.), *Enterprise interoperability iv*, pp. 439–448. London, Springer London.
- Ringwald M., Römer K., Vialetti A. (2006). *Snif: Sensor network inspection framework. department of computer science*. Technical report. ETH Zurich, Technical Report, 535.
- Sriraman B., Radhakrishnan R. (2005). Event driven architecture augmenting service oriented architectures. *Report of Unisys and Sun Microsystems*.