A Self-Configurable IoT Agent System based on Environmental Variability

Doctoral Consortium

Nathalia Nascimento
Laboratory of Software Engineering
Pontifical Catholic University of Rio de Janeiro
Rio de Janeiro, Brazil
nnascimento@inf.puc-rio.br

ABSTRACT

This thesis develops a self-configurable system to design agents for Internet of Things (IoT) applications. The proposed approach goes beyond existing methods by supporting handling variability in IoT agents according to environmental changes. As part of the research, we have designed a software framework, prototyped several IoT applications, and conducted simulation and machine learning experiments. We find that (1) IoT agents vary according to the physical, software behavior and analysis architecture; (2) the configuration of the set of agents can be adjusted and reconfigured through feedback evaluative machine learning; and (3) reconfiguring a set of agents dynamically in accordance with environmental variants leads to better performance.

KEYWORDS

Embodied agents; Internet of things; Self-configurable system; Neuroevolution; Human-in-the-loop

ACM Reference Format:
Nathalia Nascimento. 2018. A Self-Configurable IoT Agent System based on Environmental Variability. In *Proc. of the 17th International Conference on Autonomous Agents and Multiagent Systems (AAMAS 2018), Stockholm, Sweden, July 10–15, 2018,* IFAAMAS, 3 pages.

1 INTRODUCTION

Many open research challenges in the area of variability modeling and management have been identified [6]. Some of these open challenges are related to variability handling in Internet of Things (IoT) applications. As the Internet of Things promotes the use of sensors to collect data about the system context, a continuous deployment based on feedbacks obtained from system execution and the system context must be considered. In addition, to achieve IoT requirements, the development of software to live in an open and highly dynamic world is becoming mandatory. This has caused configurable systems to address context changes during system execution by postponing variability binding to run-time. Consequently, Metzger and Pohl [6] state that, to create IoT applications, there is a need of new approaches to deal with autonomic principles and variability at the same time, human-in-the-loop adaptations, and run-time quality assurance.

Proc. of the 17th International Conference on Autonomous Agents and Multiagent Systems (AAMAS 2018), M. Dastani, G. Sukthankar, E. André, S. Koenig (eds.), July 10–15, 2018, Stockholm, Sweden.

Because the adaptive and evolutive nature of agent-based systems, when combined with simulation and machine-learning techniques, the use of learning agents has been proposed as an appropriate approach to modeling IoT applications [4]. According to existing experiments and our experience with the IoT domain [3–5], we introduced possible variants of an IoT agent in [7]. In [7], we identified three main variation points to handle in order to create an IoT agent: (i) the body variability (i.e. number, types and brands of sensors and actuators); (ii) the complexity of the behavior of the agent, which varies based on the physical components that are operated by the agent (e.g. if this agent is able to communicate, the number of signals agents are able to exchange); and (iii) the agent architecture that the agent uses to sense the environment and behave accordingly (e.g. if this agent architecture is a neural network, in terms of its architectural variability the type of activation function and the number of neurons should be considered).

In this paper, our focus is to investigate how a collection of IoT agents can be configured to deal with the system's requirements and environmental changes. To meet the system's requirements (e.g. performance), this configuration consists of testing the behavior of agents by using different subsets of body, behavior and analysis architecture features. To deal with environmental changes, this configuration consists of evolving or adapting the set of agents by adding or removing features.

For this purpose, we propose a self-configurable IoT agent approach based on feedback-evaluative machine-learning. The approach involves: (i) a variability model for IoT agents; (ii) generation of sets of customized agents; (iii) feedback-evaluative machine-learning; (iv) modeling and composition of a group of IoT agents; (v) a history of environmental changes and IoT agent's features; and (vi) a feature-selection method based on both manual and automatic feedback.

2 PROBLEM

In the literature, there are many examples [2, 10] of approaches that apply feature selection to handle variability in learning agent-based systems. Most of these approaches load all features into the agents and then, uses learning techniques to test subsets of features while the agents interact with the environment. If the environment is dynamic, the solution is adjusted until the agent finds a generalist subset that makes it able to deal with a set of environment variants. Consequently, they do not address the following problems: (i) the current set of features loaded into the agents causes a restrictive search space to the learning algorithm, making it difficult to find a subset to meet the system's requirements (e.g. desired performance); (ii) the set of agents cannot be reconfigured to provide specialized and better solutions to specific environmental changes; and (iii) the user changes some requirements of the agent-based system, making it necessary to add new unpredicted features to the feature model, as described in [9].

2.1 Motivating Example: Environmental Changes

We selected one example from the IoT domain: a smart street light application. In this application scenario, we consider a set of street lights distributed in a neighborhood. For more details concerning this application scenario, see [3]. Each street light represents an IoT agent, which needs to operate in an environment in which sometimes the background light can be bright and at other times dark. With respect to the environment background light, the application scenario has some variants: (i) night (background light is equal to 0.0); (ii) late afternoon (background light is equal to 0.5); and (iii) morning (background light is equal to 1.0). Each street light contains a lighting sensor, but its local brightness also interferes on the sensor measurement.

A neuroevolution algorithm tests different subsets of features in order to find that one that provides the better performance. However, depending on the set of features that was initially provided, the learning algorithm may be unable to find this appropriate solution. In this case, it is necessary to expand the search space for the learning algorithm, by adding new or changing the variants. In addition, during a first simulation, while the background light was always bright, the collection of agents found a solution that provided a performance, say X+1, during the morning. After the environment changed to the night, the agents' solution was adjusted to deal with this change. However, this new generic solution presents a performance, say X, during the morning, and the agent is unable to return to its previous configuration, as it does not maintain different versions of its configuration. This configuration history could enable the agent to maximize its performance during different parts of the day.

3 APPROACH

Our approach handles variability in two dimensions: agent and environment. In the agent dimension, our approach gives support to automatically adjust a set of agents to meet the non-functional requirements of a specific scenario setting, by selecting a subset of body, behavior and analysis architecture features. If the best subset of features for an initial configuration does not achieve all non-functional requirements, such as a performance value, an user can evaluate the results in order to add new features or select other variants to compose the set of features. In the environment dimension, a dataset contains the history of configurations that were selected to provide specialized and better solutions to specific environmental changes. Thus, if an environmental change is previously known, the system can self-reconfigure to an old configuration instead of retraining itself.

In short, our approach consists of an embedded feature-selection method [1] co-working with a control module that supports manual and automatic interaction. In addition, we support a manual and automatic controller to manage permissible run-time adaptations caused by environmental or system requirement changes. Consequently, if the neuroevolution algorithm does not find a good solution after testing many subsets of the current set of features, an user or an external machine learning module can interfere on the internal learning module execution in order to change or expand its search space.

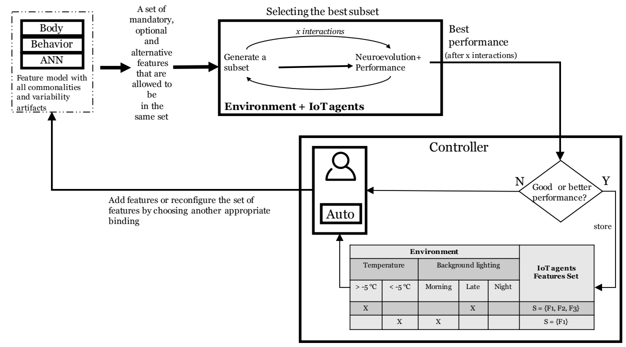

Figure 1: An adapted embedded feature-selection method approach to handle variability in IoT agents.

4 COMPLETED WORK

Currently, we have already developed a novel software framework to instantiate different applications based on IoT agents, named the Framework for the Internet of Things (FIoT) [4]. Using this framework, we prototyped four IoT agent-based applications [3–5, 8]. Based on these applications, we introduced possible variants of an IoT agent in [7].

5 CONTRIBUTIONS

We believe our approach is promising. Our contributions will be multi-fold: (i) a new software architecture to generate a set of customized agents that are able to lead with scenarios based on the Internet of Things; (ii) feature models for applications based on the Internet of Things with new specialized variants; (iii) mechanism to select the features that compose a group of agents; (iv) a mechanism based on a feedback loop and machine learning techniques to reconfigure an IoT-oriented multiagent system.

ACKNOWLEDGMENTS

The author would like to thank professors Dr. Carlos Lucena (PUC-Rio), Dr. Paulo Alencar (University of Waterloo) and Dr. Donald Cowan (University of Waterloo) for their guidance and support. The author would also like to thank CNPq, CAPES/Program 194/Process: 88881.134630/2016-01, and PUC-Rio for their support through scholarships and fellowships.

REFERENCES

[1] Girish Chandrashekar and Ferat Sahin. 2014. A survey on feature selection methods. *Computers & Electrical Engineering* 40, 1 (2014), 16–28.
[2] Inés del Campo, Victoria Martínez, Flavia Orosa, Javier Echanobe, Estibalitz Asua, and Koldo Basterretxea. 2017. Piecewise multi-linear fuzzy extreme learning machine for the implementation of intelligent agents. In *Neural Networks (IJCNN), 2017 International Joint Conference on*. IEEE, 3363–3370.

[3] Nathalia Moraes do Nascimento and Carlos José Pereira de Lucena. 2017. Engineering cooperative smart things based on embodied cognition. In *Adaptive Hardware and Systems (AHS), 2017 NASA/ESA Conference on*. IEEE, 109–116.

[4] Nathalia Moraes do Nascimento and Carlos José Pereira de Lucena. 2017. Fiot: An agent-based framework for self-adaptive and self-organizing applications based on the internet of things. *Information Sciences* 378 (2017), 161–176.

[5] Nathalia Moraes do Nascimento, Carlos José Pereira de Lucena, and Hugo Fuks. 2015. Modeling quantified things using a multi-agent system. In *Web Intelligence and Intelligent Agent Technology (WI-IAT), 2015 IEEE/WIC/ACM International Conference on*, Vol. 1. IEEE, 26–32.

[6] Andreas Metzger and Klaus Pohl. 2014. Software product line engineering and variability management: achievements and challenges. In *Proceedings of the on Future of Software Engineering*. ACM, 70–84.

[7] N. Nascimento, P. Alencar, C. Lucena, and D. Cowan. 2018. Machine Learning-based Variability Handling in IoT Agents. *ArXiv e-prints* (Feb. 2018). arXiv:cs.MA/1802.03858

[8] Carlos José Pereira de Lucena Nathalia Moraes do Nascimento, Marx Leles Viana. 2016. An IoT-based Tool for Human Gas Monitoring. In *Congresso Brasileiro de Informática em Saúde - CBIS 2016*, Vol. 15. CBIS 2016 (ISSN 2178-2857), 96–98.

[9] Amir Molzam Sharifloo, Andreas Metzger, Clément Quinton, Luciano Baresi, and Klaus Pohl. 2016. Learning and evolution in dynamic software product lines. In *Software Engineering for Adaptive and Self-Managing Systems (SEAMS), 2016 IEEE/ACM 11th International Symposium on*. IEEE, 158–164.

[10] Shimon Whiteson, Peter Stone, Kenneth O Stanley, Risto Miikkulainen, and Nate Kohl. 2005. Automatic feature selection in neuroevolution. In *Proceedings of the 7th annual conference on Genetic and evolutionary computation*. ACM, 1225–1232.