Towards Configuration Technologies for IoT Gateways

Alexander Felfernig[1] and **Seda Polat Erdeniz**[1] and **Paolo Azzoni**[2] and **Michael Jeran**[1] and **Arda Akcay**[1] and **Charalampos Doukas**[3]

Abstract. The AGILE project aims to create Internet of Things (IoT) gateway technologies that support many devices, protocols, and corresponding administration and software development activities. In this context, there are scenarios that require the support of configuration technologies. The major goal of this short paper is to provide an overview of application scenarios and related configuration technologies that will be developed within the scope of the AGILE project.

1 Introduction

Configuration is a process in which agents (users or external systems) can specify requirements and the configuration system (often denoted as the *configurator*) provides feedback in terms of solutions and/or explanations [6, 17, 18]. Configuration can be interpreted as a type of design activity where a product is composed (configured) from a set of instances corresponding to predefined component types such that the resulting configuration (solution) is consistent with a given set of constraints [17, 18]. Requirements can be regarded as specifications of intended properties of the product (i.e., specific constraints), for example, a specific *application* or *mail server version* should be included in the *system configuration*. In such contexts, system feedback for a user is provided in terms of configurations, reconfigurations, and explanations for situations in which no solution could be found. There is a multitude of application examples of knowledge-based configuration, for example, in the automotive domain, railway interlocking systems, financial services, operating systems, and software product lines [6].

Configuration services for the Internet of Things (IoT) domain [2] is a new application area. The IoT is an emerging paradigm that envisions a networked infrastructure enabling different devices (things) to be interconnected at anyplace and anytime. In this paper we discuss two basic scenarios that will be supported by software components to be developed in the AGILE research project.[4] First, *ramp-up configuration* services will be developed that help to determine an initial configuration for the whole IoT gateway infrastructure. One of the major tasks of such gateways[5] is to bridge devices to corresponding applications on the basis of different communication protocols such as Hue and Zigbee. For example, in the smarthome domain, a configuration would determine the set of sensors, connection protocols, and apps needed to make a gateway operable for the user. Second, we will develop technologies that help to *optimize configuration and reconfiguration* of communication protocols in such a way that user requirements (e.g., performance requirements) and side conditions (e.g., available bandwith) can be taken into account.

The contributions of this paper are the following. First, we introduce *Internet of Things* (IoT) as a new application domain of knowledge-based configuration technologies. Second, we provide an overview of example IoT application scenarios that are in the need of configuration support. Finally, we summarize research objectives and technological approaches that will be followed in the AGILE project.

The remainder of this paper is organized as follows. In Section 2 we provide an overview of basic configuration functionalities that will be provided in the context of AGILE ramp-up configuration scenarios. In Section 3, we focus on a runtime scenario in which we want to optimize the selection of communication protocols with regard to optimization criteria. In Section 4 we provide an overview of related work. With Section 5 we conclude the paper.

2 "Ramp-Up Configuration" in AGILE

AGILE gateways will be deployed in different domains such as *health monitoring*, *animal monitoring in wildlife areas*, *air quality and pollution monitoring*, *enhanced retail services*, *smart homes*, and *port area monitoring*. Each application scenario requires a pre-configuration which estimates the needed hardware and software components / devices to be deployed in the ramp-up phase of the system. We denote this type of configuration *ramp-up configuration* since each scenario requires a specific set of hardware components and software components (including apps) to "ramp-up" the system.

AGILE Air Pollution Monitoring. Environmental pollution has become an issue of serious international concern and is increasingly stimulating the development and adoption of solutions to monitor and reduce the effects of pollution. This is an interesting and challenging market, with both potential economical outcomes and a strong societal impact. The convergence of hardware integration, reduction of sensor costs, IoT and M2M technologies introduces a new panorama where it is really possible to deliver low cost, high quality monitoring systems with a capillary coverage of the territory. This convergence leads to a new era of solution for environmental pollution monitoring. Air quality and pollution monitoring stations are complex systems that, depending on the application context, deliver added value pollution monitoring services based on a delicate equilibrium between the adoption of the most appropriate sensors, their correlation, the selection of the correct algorithms and the configuration of their hardware and software parameters. A wrong or imprecise selection and configuration of these elements leads to misleading, wrong, and completely useless results and services. *Air Pollution Monitoring* is in the need of configuration support since the measuring equipment has to be pre-selected and parametrized in the

[1] Graz University of Technology, Austria, email: {alexander.felfernig, spolater, mjeran, aakcay}@ist.tugraz.at
[2] Eurotech Group, Italy, email: paolo.azzoni@eurotech.com
[3] Create-Net, Italy, email: cdoukas@create-net.org
[4] AGILE (An Adaptive & Modular Gateway for the IoT) is an EU-funded H2020 project 2016–2018 – see agileiot.eu.
[5] *Raspberry Pi* is one of the hardware platforms used in AGILE – see www.raspberrypi.org.

line of the environmental conditions that exist, for example, in a city.

Further AGILE Scenarios. The basic task of a configurator in *health monitoring* is to figure out which measuring devices are needed (including their parametrization) to be able to monitor and analyze specific body functions. In the *animal monitoring* scenario it is important to figure out which infrastructure can be used to complete predefined data collection tasks. In such scenarios, *reachability* of animals (and corresponding sensors) plays a major role in order to be able to complete data collection. Reachability depends on the selected drone types but also on the selected communication protocols which have different degrees of power consumption. *Enhanced retail services* that allow a personalized shopping experience in physical stores are in the need of configuration functionalities that indicate the amount and positioning of sensors (e.g., for indoor position detection) and displays that are needed to successfully support customers in their shopping experiences. In the *port area monitoring* scenario, configuration technologies are needed that help to select relevant sensors (e.g., gas, radioactivity, and water quality sensors) that are able to provide the needed data. In *smarthome* scenarios, the task of the configurator is to identify sensors, communication components, and protocols that are needed to provide the smarthome functionalities required by a customer. In this context, examples of customer requirements are rooms in the house and their type, maximum accepted price, and needed functionalities (e.g., presence monitoring and simulation, and video surveillance).

Knowledge Acquisition & Representation. When configuring, for example, *smart homes*, the configuration model includes information about the relationships between building properties and corresponding sensors (e.g., *if a room is a kitchen and includes an oven, then a corresponding temperature sensor has to be included for the room*) or between user preferences and the corresponding technical infrastructure (e.g., *if a user wants to save money, wireless communication is preferred*). A configuration for a given configuration task includes information about which components, devices, and drivers are part of the initial gateway installation.

In AGILE, we will evaluate the applicability of different types of configuration knowledge representations such as answer set programs (ASP) [13] and constraint-based representations [6, 9, 20]. Our aim is to *identify a knowledge representation language* that can be applied for each of the different application scenarios in order to provide a basic technology for supporting IoT ramp-up configuration tasks. The applicability of these languages will be primarily evaluated with regard to expressiveness and reasoning efficiency. Especially, ASP-based configuration approaches will be evaluated with regard to their applicability in typical gateway ramp-up scenarios.

Consistency Management of Knowledge Bases. Configuration knowledge bases can become inconsistent, i.e., the defined component types and constraints lead to the problem that no solution can be identified. Such a situation can occur in the context of regression testing [4] but also in situations where the conflict is induced by the configuration knowledge base itself. In such scenarios, configuration technologies in combination with model-based diagnosis [16] can be exploited to automatically identify the sources (e.g., constraints) of a given inconsistency [4]. Such functionalities will be included in a development environment for IoT configuration knowledge bases.

In the context of AGILE, we focus on the development of techniques that help to improve the efficiency of configuration knowledge engineering processes. Although automated debugging [4] is a useful means to reduce time efforts related knowledge base development and maintenance, the development and maintenance of related test cases (also denotes as examples [4]) is still costly. We will analyze the applicability of different testing approaches from software engineering and will especially focus on the the development of *mutation testing* approaches for knowledge bases [10]. In this context, a mutation will serve as a basis for generating tests that are, for example, accepted by the original knowledge base but should not.

Consistency Management of User Requirements. Consistency management not only plays a role in the context of knowledge base development and maintenance but also within the scope of a configuration process. A user of an AGILE configurator could articulate a set of requirements in such a way that no solution can be identified. Also in such a situation, model-based diagnosis approaches can be exploited to indicate sets of user requirements that have to be adapted such that at least one solution can be identified [4, 7, 12, 21]. A similar situation occurs in the context of reconfiguration, i.e., in a situation where hardware and software components of an IoT gateway have to be adapted. In this context, minimal changes have to be proposed that indicate how the existing configuration has to be adapted such that a consistent configuration can be determined that takes into account all reconfiguration requirements [8].

In AGILE, we focus on the development of personalization techniques that help to improve the diagnosis prediction quality, i.e., to identify those diagnoses that will be accepted by the user. Such personalized diagnoses will be determined on the basis of an analysis of the interaction behavior of users of similar gateway installations (available in gateway profile repositories). In this context we will develop learning-based approaches that help to calibrate search heuristics in order to improve efficiency and prediction quality of configuration and reconfiguration.

A simple example of our envisioned approach is the following. Let us assume the existence of a *configuration log* as the one shown in Table 1. The parameters req_i indicate user requirements and x_i indicate technical product parameter settings (consistent with the user requirements) accepted by the user u_i. The overall goal is to optimize the configurator search heuristics (e.g., variable and domain orderings) in such a way, that the prediction quality for the technical parameter settings is maximized. More precisely, we want to identify search heuristics that guide to solutions (configurations) that will be accepted by the *current* user. User interactions (see, e.g., Table 1) serve as a basis for learning. Prediction quality can be measured, for example, in terms of the user acceptance degree of parameter settings (configurations) proposed by the configurator. In this context we will evaluate different clustering techniques, i.e., to learn heuristics not on a global level, but depending on a specific cluster derived, for example, from the user requirements.

$user$	req_1	req_2	x_1	x_2	x_3	x_4
u_1	1	2	3	4	4	2
u_2	2	2	8	3	4	2
u_3	1	2	3	4	5	2
$current$	1	1	?	?	?	?

Table 1. Example configuration log as a basis for optimizing the prediction quality of configuration parameters (req_i represent user requirements and x_i represent technical product parameter settings accepted by previous users).

3 Runtime Configuration in AGILE

Modern embedded systems included in IoT scenarios support a rich set of connectivity solutions (e.g., 3G, LTE, TD-LTE, FDD-LTD, WIMAX, and Lora). In this context, configuration technologies play an important role in terms of suggesting optimal connectivity configurations. Such configurations include a collection of connectivity solutions that are needed to support a set of active applications (apps). Criteria that have to be taken into account are, for example, location information, available connectivity, performance and reliability requirements, contractual aspects, and costs.

In AGILE, runtime configuration must be performed on the gateway – in contrast, ramp-up configuration can also take place in the cloud. On the one hand we will evaluate different types of reasoning engines, for example, the CHOCO constraint solver[6] and the Sat4j boolean satisfaction library[7]. We will also take into account the application of rule engines[8], optimization libraries, and knowledge compression techniques [1] to assure efficiency of problem solving on the gateway level.

In AGILE, gateway configurations can be manually defined by users but also be determined on the basis of a configurator that is in charge of keeping the overall system installations consistent. A configurator (e.g., a constraint solver) can determine alternative configurations which have to be ranked. In order to determine a ranking for alternative configurations, a MAUT[9] approach can be used [22]. Examples of evaluation *dimensions* (dim) used in MAUT could be *performance*, *reliability*, and *costs*. Depending on the current gateway configuration and the usage context, a configurator can determine alternative (re-)configurations and rank them accordingly.

A simplified example of the application of a utility-based approach is the following. Table 2 includes an evaluation of connectivity protocol configurations $conf$ ($conf_a$ and $conf_b$) to be used on the gateway, for example, for different types of data exchange. The three evaluation dimensions used in this example are performance, reliability, and costs. Furthermore, Table 3 includes the personal preferences of two different gateway users (u_1 and u_2).

In order to determine the configuration that should be chosen for a specific user, we can apply a utility function (see, e.g., Formula 1).

$$utility(conf, u) = \Sigma_{d \in dim} interest(u, d) \times value(conf, d) \quad (1)$$

In this context, *utility(conf,u)* denotes the utility of the configuration $conf$ for the user u, *interest(u,d)* denotes the interest of user u in evaluation dimension d, and *value(conf,u)* denotes the contribution of configuration $conf$ to the interest dimension d. In the example, configuration $conf_a$ has a higher utility for user u_1 (107.0) whereas configuration $conf_b$ has a higher utility for u_2 (111.0). Note that for simplicity we omitted to sketch the determination of the evaluations depicted in Table 2 – for details see [5]. In order to increase the efficiency of runtime configuration, we will evaluate knowledge compression techniques that help to reduce search efforts as much as possible. For example, we will apply decision diagram techniques [1] to pre-calculate possible configurations and re-configurations.

Table 4 provides a summary of the configuration-related research objectives in AGILE. Within the context of ramp-up configuration scenarios we will identify knowledge representation mechanisms that allow an easy representation of the AGILE IoT domains introduced in Section 2. Furthermore, we will develop test case generation techniques that will help to make the development and management of test cases more efficient. For AGILE scenarios, we will develop concepts that support the learning of search heuristics to optimize configuration and reconfiguration processes. Furthermore, we will work on knowledge compression techniques [1] that help to make solution search on the gateway level as efficient as possible.

[6] choco-solver.org.
[7] sat4j.org.
[8] java-source.net/open-source/rule-engines.
[9] Multi-attribute utility theory.

4 Related Work

Although different from basic IoT scenarios [2], there exist applications that support the configuration of systems including hardware and software components. Falkner and Schreiner [3] introduce approaches to the configuration of railway interlocking systems as examples of complex industrial systems designed on the basis of constraint-based configuration technologies. Krebs et al. [11] show the application of configuration technologies in the area of car periphery supervision that includes detection of the car environment, the recognition of hazardous situations, and the handling of difficult traffic situations. Related applications are pre-crash detection, the detection of obstacles, and parking assistance. Related car configuration processes have to take into account existing hardware components and to combine these with the corresponding software units. Finally, Perera et al. [15] introduce an approach to the end-user-oriented configuration of IoT middleware components.

The afore mentioned approaches are in the line of the mentioned "ramp-up" scenario, i.e., infrastructures are configured before the system is operable. In contrast to the developments in [3, 11, 15], the "ramp-up" configuration approach that is currently developed in AGILE focuses on advanced testing methods for supporting configuration knowledge engineering and also on approaches to improve configurator usability by the inclusion of different types of personalized consistency restoration methods. Initial approaches to include recommendation methods into configuration problem solving are documented, for example, in [19]. These approaches do not take into account the issue of consistency management in a satisfactory fashion which will be a major focus of our work in the AGILE project. Finally, for an overview of different IoT smart solutions available on the market we refer to [14].

5 Conclusions and Future Work

In this paper we provide a short introduction to basic configuration scenarios of the AGILE project. We discussed the two scenarios of "ramp-up configuration" and "runtime optimization". Major challenges for our future work will be approaches to automated test case generation for configuration knowledge bases, efficient techniques to solve the "no solution can be found" problem in interactive configuration settings, and the personalization of related repair approaches.

REFERENCES

[1] H. Andersen, 'An introduction to binary decision diagrams', in *Lecture Notes for Efficient Algorithms and Programs*, pp. 1–35, (1999).

[2] L. Atzori, A. Iera, and G. Morabito, 'The Internet of Things: A survey', *Computer Networks*, **54**(15), 2787–2805, (2010).

[3] A. Falkner and H. Schreiner, 'SIEMENS: Configuration and Reconfiguration in Industry', in *Knowledge-based Configuration – From Research to Business Cases*, eds., A. Felfernig, L. Hotz, C. Bagley, and J. Tiihonen, pp. 199–210. Morgan Kaufmann, (2014).

[4] A. Felfernig, G. Friedrich, D. Jannach, and M. Stumptner, 'Consistency-based diagnosis of configuration knowledge bases', *Artificial Intelligence*, **152**(2), 213–234, (2004).

[5] A. Felfernig, G. Friedrich, D. Jannach, and M. Zanker, 'An environment for the development of knowledge-based recommender applications', *International Journal of Electronic Commerce (IJEC)*, **11**(2), 11–34, (2006).

[6] A. Felfernig, L. Hotz, C. Bagley, and J. Tiihonen, *Knowledge-based Configuration: From Research to Business Cases*, Elsevier/Morgan Kaufmann Publishers, 1st edn., 2014.

configuration	perfor-mance	reliability	costs
$conf_a$	9	5	2
$conf_b$	5	8	3

Table 2. Utility table: evaluation of configurations with regard to the interest dimensions *performance*, *reliability*, and *costs*.

user	performance	reliability	costs
u_1	10	3	1
u_2	5	7	10

Table 3. Example user preferences w.r.t. interest dimensions *performance*, *reliability*, and *costs*.

configuration topic	research objective
appropriate knowledge representations	knowledge representations for easy modeling and efficient configuration search
efficiency of knowledge base development and maintenance	automated test case generation and mutation testing
personalized consistency management	personalized configuration based on learning search heuristics and knowledge compression techniques

Table 4. Overview of AGILE research objectives.

[7] A. Felfernig, M. Schubert, and C. Zehentner, 'An Efficient Diagnosis Algorithm for Inconsistent Constraint Sets', *Artificial Intelligence for Engineering Design, Analysis, and Manufacturing (AIEDAM)*, **25**(2), 175–184, (2011).

[8] A. Felfernig, R. Walter, and S. Reiterer, 'FlexDiag: AnyTime Diagnosis for Reconfiguration', in *16th International Workshop on Configuration*, pp. 105–110, Vienna, Austria, (2015).

[9] G. Fleischanderl, G. Friedrich, A. Haselböck, H. Schreiner, and M. Stumptner, 'Configuring large systems using generative constraint satisfaction', *IEEE Intelligent Systems*, **13**(4), 59–68, (1998).

[10] Y. Jia and M. Harman, 'An Analysis and Survey of the Development of Mutation Testing', *IEEE Transactions on Software Engineering*, **37**(5), 649–678, (2011).

[11] T. Krebs, L. Hotz, and A. Günter, 'Knowledge-based Configuration for Configuring Combined Hardware/Software Systems', in *Proceedings of PuK'2002*, pp. 1–6, Freiburg, Germany, (2002).

[12] J. Marques-Silva, F. Heras, M. Janota, A. Previti, and A. Belov, 'On computing minimal correction subsets', in *IJCAI*, pp. 615–622, (2013).

[13] V. Myllärniemi, J. Tiihonen, M. Raatikainen, and A. Felfernig, 'Using answer set programming for feature model representation and configuration', in *Workshop on Configuration*, pp. 1–8, Novi Sad, (2014).

[14] C. Perera, C. Liu, and S. Jayawardena, 'The Emerging Internet of Things Marketplace From an Industrial Perspective: A Survey', *IEEE Transactions on Emerging Topics in Computing*, **3**(4), 585–598, (2015).

[15] C. Perera, A. Zaslavsky, M. Compton, P. Christen, and D. Georgakopoulos, 'Semantic-driven configuration of internet of things middleware', in *9th International Conference on Semantics, Knowledge & Grids (SKG)*, pp. 66–73, Beijing, China, (2013).

[16] R. Reiter, 'A theory of diagnosis from first principles', *Artificial Intelligence*, **32**(1), 57–95, (1987).

[17] Daniel Sabin and Rainer Weigel, 'Product configuration frameworks - a survey', *IEEE Intelligent Systems*, **13**(4), 42–49, (1998).

[18] M. Stumptner, 'An overview of knowledge-based configuration', *AICOM*, **10**(2), 111–125, (1997).

[19] J. Tiihonen and A. Felfernig, 'Towards Recommending Configurable Offerings', *International Journal of Mass Customization*, **3**(4), 389–406, (2010).

[20] E. Tsang, *Foundations of Constraint Satisfaction*, Academic Press, London, 1993.

[21] R. Walter, A. Felfernig, and W. Küchlin, 'Constraint-based and sat-based diagnosis of automotive configuration problems', *Journal of Intelligent Information Systems*, 1–32, (2016).

[22] D. Winterfeldt and W. Edwards, *Decision Analysis and Behavioral Research*, Cambridge University Press, 1986.