

# Paper 5

*by* Paper Aradea

---

**Submission date:** 08-Oct-2020 03:47PM (UTC+0700)

**Submission ID:** 1408912279

**File name:** B1.\_Paper\_SICIR-IOP\_2019.pdf (731.86K)

**Word count:** 3346

**Character count:** 18568

PAPER • OPEN ACCESS

## Self-adaptation modeling for service evolution on the Internet of Things (IoT)

To cite this article: Aradea *et al* 2019 *IOP Conf. Ser.: Mater. Sci. Eng.* **550** 012026

View the [article online](#) for updates and enhancements.



**IOP | ebooks™**

Bringing you innovative digital publishing with leading voices to create your essential collection of books in STEM research.

Start exploring the collection - download the first chapter of every title for free.

## Self-adaptation modeling for service evolution on the Internet of Things (IoT)

Aradea<sup>1</sup>, I Supriana<sup>2</sup>, K Surendro<sup>2</sup>, and H Mubarak<sup>1</sup>

<sup>1</sup>Department of Informatics, Faculty of Engineering, Siliwangi University

<sup>2</sup>School of Electrical Engineering and Informatics, Bandung Institute of Technology

Email: aradea@unsil.ac.id; iping@informatika.org; endro@informatika.org;  
husni.mubarak@unsil.ac.id

**Abstract.** One of the most important concepts of the internet of things is related to software services, where the system must be able to provide real-time data collected from various environments. So that software services represent a variety of physical and virtual real-world objects that can grow very fast. This condition is related to the ability of self-adaptation software services. However, the existing model of self-adaptation is generally not paying attention to the requirements for service evolution. The objective of this paper is to introduce self-adaptation modeling techniques which consist of, first, domain modeling of the internet of things to represent real-world context; second, developing an inference engine for context inference. As a form of evaluation, this model is applied to the patient monitoring system that will relate to the concept of self-adaptation of various systems, devices, actors, and environment. The case study results show the system's ability to anticipate changes in context and growth needs.

### 1. Introduction

A software service is required to have smart solutions, such as managing various resources, data and others. The internet of things (IoT) is a concept that can answer the challenge. Currently there are several definitions of IoT, but basically aims to expand the benefits of internet connectivity that can integrate real-world objects as physical and virtual representations on an ongoing basis. The IoT concept offers many advantages, such as efficiency and effectiveness of application, access speed, service accuracy, and more. The various advantages offered will not have a significant impact if only temporary. This is related to how far the concept offered can meet the dynamism and diversity of its environment.

Self-adaptation model has been present and can be a solution to issues related to complexity of the system, such as demands for autonomy, flexibility, scalability, reliability, speed, etc. [1]. Self-adaptation is a system that automatically can take appropriate action based on its knowledge of what is happening in the system [2], the system can modify its behavior in response to changes in the system itself or its environmental changes [3]. However, the study of self-adaptation related to the requirements of system evolution still requires further study [4], because a number of current researches are mostly aimed at requirements specification, monitoring, and verification, while studies on evolution for self-adaptation systems still lack of research [5-6].

The self-adaptation system has gained much attention from researchers, such as Morandini [7] and Dalpiaz [8] developing an adaptive system by taking the case of a patient monitoring system. The proposed model highlights the importance of modeling variability and adaptability at run-time. With the

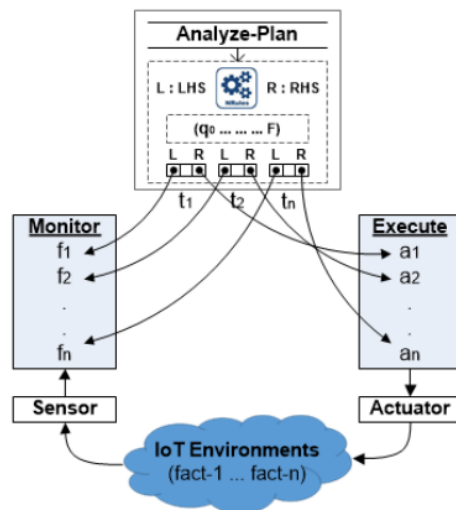


same case study, we modified it according to the proposed model, and added system evolution capabilities not yet covered in both works. In addition, researchers in this field have exploited many autonomic computing concepts through various approaches, such as Gabriel [9] focusing on quality of service specifications through context-awareness models, Dhaminda [10] adopting a natural language approach to goal pattern model, Arciani [11] utilizing abstract state machines to achieve context awareness and self-awareness, Knauss [12] uses machine learning approaches to introduce contextual requirements, as well as Paz [13] through formal methods of proposing dynamic adaptation mechanism planning. The concept adopted by these researchers was adopted in the proposed model. However, we add a domain problem representation through goal modeling and evolutionary functions, so that the model relates to the requirements engineering area in dealing with the variation of operational changes and allows it to accommodate its growth. Based on this fact, this paper proposes a self-adaptation modeling technique in the IoT concept. Section 2 of this paper describes the proposed model, section 3 application of the proposed model into a case study, section 4 discusses related work, and section 5 contains conclusions and future work.

## 2. Self-adaptation model

All elements and activities in the system environment require modeling and control that can represent their behavior in various conditions in the real world [14]. Self-adaptation model is proposed to realize a software service that is capable of reflecting every element within its environment and has the adaptability to its changes. The model consists of (a) domain model, is a basic function for the IoT context and application logic, (b) control model, is an inference engine to manage domain model through adaptation logic.

The approach used in this model is an extension of previous research conducted by Supriana (2016) and Aradea (2017) [15] [16]. Domain models were developed by adopting a goal-based approach [17] through model extensions [4], mapped to the concept of autonomic computing [18]. The model generally has the main activity, ie monitors, analyzers, planners, executors, and knowledge (MAPE-K). Adaptation patterns are controlled using an autonomic manager. The autonomic manager contains the rule engine in the form of event-condition-action (ECA) rules, and in our version, the concept is extended by using the rule editor model, where the rule changes can be done directly on the knowledge base.



**Figure 1.** Self-adaptation model.

Figure 1 shows the self-adaptation model developed. Each computational process is transformed into the model domain based on the mapping rules [19] through a component-based software approach [20], as well as utilizing several design patterns [21]. Each root goal in the model domain with AND-decomposition is defined as a component of the analyze & plan (AP), while each sub-goal can be defined as a monitor component (M) and/or an execute component (E), Knowledge (K) is represented as rule-based systems with the following basic structures:

WHEN<event>; *state transition*  
 IF<condition>; *conditions to trigger action*  
 THEN<action>; *action when the event takes place*  
 VALID-TIME<time\_period>; *a period of time during adaptation needs to be applied*

The sensor monitors the context information that becomes the input set as <event> and is captured by the context processor, where  $\Sigma$ : context information is a set of facts ( $f_1, f_n$ ) captured from an IoT environment event and can determine state changes. The controller evaluates the <condition>, referring to a certain <event> that occurs in the state set, and this controller will trigger the performer action component to do the <action> adaptation through the transition function to the <condition> that applies at a certain <time\_period>. Q: the action ( $a_1, a_n$ ) behavior is a set of state system behavior, from the initial state ( $q_0$ ) to the state that can be the target (F).  $\delta$ : the transition ( $t_1, t_n$ ) function as the time of the adaptation action based on an evaluation of the condition to change the initial state ( $q_0$ ) to the target state (F). The quality of the inference engine depends on the selection of one state, where the state selected as the adaptation action in the right hand side (RHS) class is determined by the expression in the class left hand side (LHS) which is the compatibility between the rules and facts. The strategy developed for this mechanism is the forward strategy, which is the reuse of existing components and matches the required ones.

### 3. Case study: patient monitoring systems

In order to evaluate the model, we developed a patient monitoring system (PMS) case study. The PMS services have made significant advances and are supported by medical technologies to generate various devices for software services. For example smart-home scenario that can help patients in carrying out their daily activities. This system will connect with various devices, such as cameras, meal sensors, medicine sensors, thermometers, oxymeters, smart shirts and other devices. In addition, this system will connect with other systems such as hospitals, nutrition consultancy services, clinics, etc., and other important things will involve various professions as system actors, such as doctors, nurses, social workers and other relevant actors. This fact shows that the PMS services has a variety of interrelated elements, and eventually will cause various problems related to software services.

Case study of PMS is a case problem have been discussed by fellow researchers [7, 8]. In this discussion, case descriptions are modified in accordance with the proposed model. The goal modeling for patient monitoring system can be seen in figure 2. The system has three sub-goals to be achieved, ie feeding schedule, medication schedule, and patient service. This system will connect with several sensors, including meal sensors associated with real-time monitoring plan, food ordering, and activation alarm-1, other sensors are medicine sensors associated with the plan to monitor patient prescriptions and activation of alarm-2. Coordination of all plans will be regulated through the achievement of patient service goals.



Figure 2. Patient monitoring system modeling.

The adaptation mechanisms required for this case consist of (a) adaptation to changes and growth of context information related to patient feeding activities, (b) adaptation to changes and growth of context information related to the activity of taking medicine for the patient. The system requirement of this case can be illustrated as follows:

- There are a number of properties to monitor: (a) external properties: a number of sensors, system actors, and other related systems within the IoT environment, (b) internal properties: functional and non-functional systems, represented as goals and soft goals in figure 2.
- The system state at runtime is a combination of both values of the property. Violations will be analyzed based on the symptom of each detected event.
- There are several rules required in this case, namely (a) a set of rules for organizing components, when new events arise and the unavailability or incompatibility of context information, (b) a set of rules for handling the addition and/or changes of new functions.

Based on the modeling of figure 2, the three sub-goals "follow medical instructions" are AND-decomposition, meaning that the variables in each goal are dependent on each other. This can be represented by setting up detectable events or symptoms, such as mealtime events, deliver food events, medicine supply events, and more. Each event can be set through the rule engine, as follows:

- Mealtime events are generated when changes are detected or unavailable of context information, thus having to perform system reconfiguration.
- Deliver food events are generated when the patient's food unavailability is detected, thus having to order food from outside service providers.
- Medicine supply events are generated when an unsuitable supply of drugs is detected, thus calling an assistant or nurse

The setting of mealtime events is manifested based on the "eating time" component function as a monitor component (M), which determines the function of the execute (E) "deliver food" and "eat meals" components. The decomposition of this sub-goal is OR-decomposition, thus indicating the variability of the context information (*eating\_time*), which consists of breakfast, lunch and dinner. While delivering food events will be related to order availability for patients (*order\_food*), whether available or not. In addition, the system components are also associated with the resources of the meal sensor, and it may



be possible unexpected events or errors (*event\_error*), such as sensor damage, lost connections, etc. Based on the description, then obtained three system variables, namely: "eating\_time", "order\_food" and "event\_error", so the plan can be set as follows: *plan (eating\_time, order\_food, event\_error)*. The settings of the system behavior relate to the rule specification, in this case for example set as follows:

**Rule-1:**

*if (eating\_time = time.breakfast or time.lunch or time.dinner) and (order\_food = true) and (event\_error = null) then plan = activate alarm*

**Rule-2:**

*if (eating\_time = time.breakfast or time.lunch or time.dinner) and (order\_food = false) and (event\_error = null) then plan = create new service\_action*

**Rule-3:**

*if (eating\_time = null) and (event\_error = null) then plan = change service\_action*

**Rule-4:**

*if (event\_error = not null) then plan = change service\_action*

**Rule-5:**

*if not [criteria] then plan = change service\_action*

This plan is realized to overcome variability in fulfilling goal and softgoal. For example, goal achievement for feeding schedule and scheduled to take medicine will use the "patient service" component function as a component of analyze and plan (AP), as it contributes positively (++) to "availability" and "relevance" softgoals, compared to analyze and plan (AP) each that only contribute positively (+), will even be affected negative contribution (-) when detected change. The rule specifications can be mapped into ECA rules (table 1).

**Table 1.** ECA-feeding schedule.

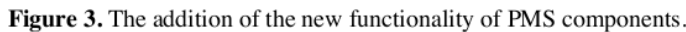
Event (E)	Condition (C)	Action (A)
<i>mealtime_events</i>	<i>(eating_time = time.breakfast or time.lunch or time.dinner) ; (order_food = true) ; (event_error = null) ;</i>	<i>P<sub>1</sub> = activate_alarm</i>
<i>mealtime_events</i> <i>deliver_food_events</i>	<i>(eating_time = time.breakfast or time.lunch or time.dinner) ; (order_food = false) ; (event_error = null) ;</i>	<i>P<sub>2</sub> = create new service_action</i>
<i>mealtime_events</i>	<i>(eating_time = null) ; (event_error = not null) ; not [criteria] ;</i>	<i>P<sub>3</sub> = change service_action</i>

Adaptation action in table 1 is an adaptation for changes in context information, for example:

- *P<sub>1</sub> = activate\_alarm*, set alarm messages based on prevailing conditions. In this case the system will connect with the repository, sensor, etc.

As for fulfilling the evolution of the system and as a follow-up of adaptation, the system can handle its growth. For example:

- *P<sub>2</sub> = create new service\_action*, adding a new function to order patient food from a catering service provider based on information/ advice on nutrition consultancy services. In this case the system will connect with other systems within the IoT environment.
- *P<sub>3</sub> = change service\_action*, function to call the assistant/ nurse when the patient's feeding schedule is not available, or contact the service desk when interruption occurs. In this case the system will connect with other system actors within the IoT environment



## 4. Conclusion

The results of the case studies show the proposed model allows to anticipate changes in the range of environmental entities. In addition, the model also takes into account its evolutionary needs, so the system can be expanded easily and as needed. We believe this model can contribute to the IoT domain, where the system must be viewed based on its life cycle that will continue to grow. As future work, we plan to extend the context inference strategy to formulate conflict resolution and apply it to more complex cases.

- [1] Supriana I and Aradea 2015 Automatically relation modeling on spatial relationship as self-adaptation ability *International Conference on Advanced Informatics: Concept, Theory and Application (ICAICTA)* vol. **2** (Chonburi-Thailand: IEEE Xplore) p 157–162
- [2] Ganek A G and Corbi T A 2003 The dawning of the autonomic computing era *IBM Systems Journal* **42**(1) 5–18
- [3] Cheng B H C, Eder K I, Gogolla M, Grunske L, Litoiu M, Muller H A, Pelliccione P, Perini A, Qureshi N, Rumpe B, Schneider D, Trollmann F and Villegas N 2014 Using models at runtime to address assurance for self-adaptive *LNCS Springer* 101–136
- [4] Aradea, Supriana I and Surendro K 2018 Self-adaptive software modeling based on contextual requirements *Telecommunication, Computing, Electronics and Control (TELKOMNIKA)* **16**(3) 1276–88
- [5] Sucipto S and Wahono R S 2015 A Systematic Literature Review of Requirements Engineering for Self-Adaptive Systems *Journal of Software Engineering* **1**(1) 17–27
- [6] Knauss A 2015 *The Capture and Evolution of Contextual Requirements: The Case of Adaptive Systems* Dissertation (Hannover: University of Victoria)



- [7] Morandini M, Penserini L, Perini A and Marchetto A 2017 Engineering requirements for adaptive systems *Requirements Engineering* **22**(1) 77–103
- [8] Dalpiaz F, Giorgini P and Mylopoulos J 2013 Adaptive socio-technical systems: a requirements-based approach *Requirements Engineering* **18**(1) 1–24
- [9] Tamura G, Villegas N, Muller H, Duchien L and Seinturier L 2013 Improving context-awareness in self-adaptation using the DYNAMICO reference model *8<sup>th</sup> Int. Syp. Software Engineering for Adaptive and Self-Managing Systems* 153–62
- [10] Abeywickrama B D, Bicocchi N and Zambonelli F 2012 SOTA: towards a general model for self-adaptive systems *Proc. 2012 IEEE 21st International WETICE* 48–53
- [11] Arcaini P, Riccobene E, Scandurra P 2015 Modeling and Analyzing MAPE-K Feedback Loops for Self-adaptation *SEAMS 15' Proc. 10<sup>th</sup> Int. Sym. Software Eng. For Adaptive and Self Managing Systems* (USA: IEEE Press) 13–23
- [12] Knauss A, Damian D, Franch X, Rook A, Müller HA and Thomo A 2015 ACon: A learning-based approach to deal with uncertainty in contextual requirements at runtime *Journal of Information and Software Technology* **70** 85–99
- [13] Paz A and Arboleda H 2016 A model to guide dynamic adaptation planning in self-adaptive systems *Electronic Notes in Theoretical Computer Science* **321** 67–88
- [14] Aradea, Supriana I and Surendro K 2014 An overview of multi agent system approach in knowledge management model *International Conference on Information Technology Systems and Innovation (ICITSI 2014) - Proceeding* (Bandung) 62–69
- [15] Supriana I, Surendro K, Aradea and Ramadhan E 2016 Self-adaptive cyber city system *International Conference on Advanced Informatics: Concepts, Theory and Applications (ICAICTA)* IEEE Malaysia
- [16] Aradea, Supriana I, Surendro K and Darmawan I 2017 Integration of self-adaptation approach on requirements modeling *Recent advances on soft computing and data mining. SCDM. Advances in Intelligent Systems and Computing* vol. 549 ed T Herawan, R Ghazali, N Nawi, M Deris (Cham, Switzerland: Springer) p 233–43
- [17] Bresciani P, Perini A, Giorgini P, Giunchiglia F and Mylopoulos J 2004 Tropos: An agent-oriented software development methodology *Journal of Autonomous Agents and Multi-Agent Systems* **8**(3) 203–36
- [18] Kephart JO and Chess DM 2003 The vision of autonomic computing *Computer* **36**(1) 41–50
- [19] Nakagawa H, Ohsuga A and Honiden S 2012 Towards dynamic evolution of self-adaptive systems based on dynamic updating of control loops *Sixth International Conference on Self-Adaptive and Self-Organizing Systems (SASO)*, (IEEE Press) pp 59–68
- [20] Hirsch D, Kramer J, Magee J and Uchitel S 2006 Modes for software architectures *In EWSA* pages 113–26
- [21] Abuseta Y and Swesi K 2015 Design patterns for self adaptive systems engineering *International Journal of Software Engineering & Applications (IJSEA)* **6**(4)

# Paper 5

## ORIGINALITY REPORT

7 %

SIMILARITY INDEX

2 %

INTERNET SOURCES

5 %

PUBLICATIONS

9 %

STUDENT PAPERS

## PRIMARY SOURCES

1

Submitted to Universiti Teknologi Malaysia

Student Paper

4 %

2

Y Hasegawa. "Enhanced Faraday rotation of cube-shaped EuS nanocrystals with a magnetic coercive field", IOP Conference Series Materials Science and Engineering, 02/01/2009

Publication

3 %

Exclude quotes Off

Exclude bibliography On

Exclude matches < 2%