ISSN 1870-4069

Design of a Language for IoT Service Composition

Isaac Machorro-Cano[1], Giner Alor-Hernández[1], José Oscar Olmedo-Aguirre[2], Lisbeth Rodríguez-Mazahua[1], Mónica Guadalupe Segura-Ozuna[3]

[1] Tecnológico Nacional de México / I. T. Orizaba, Orizaba, Veracruz, Mexico
[2] CINVESTAV-IPN, Electrical Engineering, Mexico City, Mexico
[3] Universidad del Papaloapan (UNPA), Tuxtepec, Oaxaca, Mexico
imachorro@gmail.com, galor@itorizaba.edu.mx,
lrodriguez@itorizaba.edu.mx,oolmedo@cinvestav.mx,
msegura@unpa.edu.mx

Abstract. In the Internet of Things (IoT) ecosystem, multiple smart devices communicate among them and with people. Similarly, they are primarily characterized by remarkable detection and processing capabilities. On the other hand, a service composition (SC) task involves performing the orchestration or choreography of services. SC is frequently studied in the context of Web services (WS), where a series of standards have been developed and used in real-world implementations to support SC. Unfortunately, these standards are inadequate in the IoT paradigm, since IoT devices are based on data/events and the resources are restricted. This research work proposes the design of a language for IoT SC, while simultaneously discussing important literature on SC, business process, IoT service orchestration, and IoT service choreography. Finally, as a proof-of-concept, we present a case study of IoT service composition in the healthcare domain for patients with overweight or obesity.

Keywords: business process, IoT, orchestration, service composition.

1 Introduction

The IoT is the important evolution of the Internet where heterogeneous devices and machines are interconnected among them and with people. Recently, the microcontrollers used to communicate over the Internet have gained popularity, thus giving rise to a wide variety of smart and networked devices, such as digitally enhanced objects, motion detectors, health surveillance devices, electric meters, and even street lights. All these devices are chiefly characterized by their detection, processing, and network connection capabilities [1].

In order to support the real-time communication of smart objects through the Internet, some web protocols are being implemented in real time. These protocols are compatible with smart objects, Web open source standards such as the devices profile for web services and Constraint Application Protocols (CoAPs). In parallel, the services offered by smart objects are accessed directly on the Web and interact with a large number of conventional web services to form a new generation of ubiquitous applications [2]; however, there seems to be a problem regarding the service composition (SC) of smart objects.

SC is a basic principle of service-oriented computing (SOC) [3] where different services are combined together in order to satisfy complex user requirements. Two important aspects of SC are service orchestration and service choreography [4]. SC is frequently studied in the context of web services and business processes, where a series of standards are developed and used in real-world implementations to support it. However, the current characteristics of IoT systems (e.g. devices are based on da-ta/events), as well as issues such as resource-restriction, make some of the techniques developed for Web SC inadequate when applied in the IoT, thereby implying that new SC mechanisms should be developed by taking into account the new requirements of IoT systems.

Real-time support for web-based protocols has paved the way for the arrival of new IoT applications. Moreover, SC seeks to reuse several services from existing components by joining them in a creative way; the idea is that when applied to the IoT context, streamlines the development of IoT applications. Likewise, SC applied in the IoT allows combining services from multiple smart objects to satisfy complex user needs across a wide range of application domains, and it is used to create innovative applications in a more efficient way [5].

This research work proposes an IoT-based language for service composition that considers both service orchestration and service choreography mechanisms. The remainder of the paper is structured as follows: Section 2 discusses relevant literature on SC applied in the IoT, service orchestration and choreography and business pro-cesses (BP). On the other hand, Section 3 proposes the design of our language for IoT SC, whereas Section 4 introduces a case study of SC for the IoT in the healthcare domain, specifically for patients with overweight or obesity. Finally, section 5 pre-sents the research conclusions and suggestions for future work.

2 Related Works

In their work, Yang and Li [4] proposed a strategy for selecting and aggregating sensory data to address the issue of IoT information service composition. The strategy considers the modeling and evaluation of the quality of service (QoS) in the IoT. Moreover, the authors implemented a binary genetic algorithm to identify the best SC solutions. On the other hand, Dar et al. [6] discussed the design and implementation of ROA, a generic architecture model with tools for integrating end-to-end systems and IoT-based business processes. Additionally, Ren et al. [7] presented a service selection model that highlights the global synergy effect based on collaboration requirements. The validity and advantages of the model and the algorithm were tested through the smart car manufacturing simulation experiment in the cloud.

Rapti et al. [8] proposed a decentralized service composition model for pervasive IoT environments that relies on artificial potential fields (APFs), while Dijkman et al. [9] proposed a framework for developing business models in IoT applications. The framework was created from a literature survey on existing business model frameworks. Then, the authors adapted these frameworks based on surveys to 11 companies that develop IoT applications. Conversely, Vidyasankar [10] proposed a transaction model and a correction criterion for executions of SC in the IoT. The proposal defines relaxed atomicity and isolation properties for transactions in a flexible manner and can thus be adapted for multiple IoT applications. Additionally, Ju et al. [11] presented a generic business model for IoT services that took as its basis a literature analysis and interviews with eight experts working for IoT companies. Similarly, the authors identified the key components of an IoT business model: key partners, key resources, key activities, and value propositions.

Baker et al. [12] developed a multi-cloud IoT SC algorithm called E2C2, which seeks to create a conscious energy composition plan by looking for and integrating the least possible number of IoT services. To test E2C2's performance, the authors evaluated it against four SC algorithms in multiple cloud environments (i.e. All clouds, Base cloud, Smart cloud, and COM2). Meanwhile, Bergesio et al. [13] proposed an object-oriented model capable of orchestrating services and using the services in a stand-alone system to help users personalize smart spaces.

Furthermore, the model provides an automatic adaptation of a "personalization" when the environment is modified. On the other hand, Wen et al. [14] proposed a fog orchestrator to facilitate the centraliza-tion of a group of resources, map applications to specific requests, offer an automated workflow to physical resources, generate workload execution with control of runtime QoS, and create efficient directives over time to manipulate objects. In addition, Macker and Taylor [15] proposed the Network Edge Workflow Tool (Newt) for two use cases. In the first case, Newt was used to implement a causal workflow in a disaster response scenario, whereas in the second case, it was used to orchestrate William Shakespeare's Hamlet by distributing the actors across an emulated wireless environment and having them exchange information.

In their work, Duhart et al. [16] presented the Environment Monitoring and Management Agent (EMMA) framework, which relies on a set of elements for designing distributed architectures for receptive environments. The authors used the resource-oriented architecture (ROA). In addition, Chen and Englund [17] discussed a choreography platform for Internet-oriented services, which performs the choreography of heterogeneous services by means of an automatic synthesis of choreography dia-grams. The approach was particularly proposed for Cooperative Intelligent Transport Systems (C-ITS), where vehicles, infrastructure, and cloud services are interconnected and cooperate to achieve efficient transport solutions. Finally, Pahl et al. [18] pre-sented an architectural pattern with its underlying principles. The pattern combines IoT edge orchestration with a provenance mechanism, which relies on the chain of blocks for trusted orchestration administration (TOM) in the cloud.

As can be observed, the majority of the works do not draw upon traditional Web services or wearables for IoT service composition. The following section introduces our language for IoT service composition and thoroughly discusses its design and its basic elements that enable service orchestration.

3 Design of an IoT-based Language for Service Composition

Service composition is performed through either service orchestration or service choreography. Service orchestration is a centralized process for organizing interac-tions among the services of an activity or business process; however, orchestrators involved in a same service orchestration task rarely know each other. Similarly, note that service orchestration was designed to orchestrate both conventional web services described in WSDL (Web Services Description Language) and REST services (Rep-resentational State Transfer) that currently use the different providers of wearable devices.

In this paper, we present the design of a language for IoT service composition, which has the necessary elements to perform the orchestration of services. The lan-guage's service orchestration elements are explained below:

- **orchestrationIoT**: The beginning and the end of the services orchestration in the IoT.
- **definitionRoles**: Label to define business processes and their roles.
- **varorc**: Data/states to be used within the business processes.
- **colaborations**: Label for conversations in the business process.
- **exceptions**: Label for handling exceptions.
- **faultRecovery**: Label for error recovery.
- **eventCtrl**: Label for concurrent events.
- **busproflow**: Implement the business process flow.
- **invdoe**: Method for invocating data or events from the smart device.
- **recdoe**: Method for receiving data or events from the smart device.
- **repdoe**: Method for responding to data or events from the smart device.
- **deviceData**: Label used to obtain the description of the smart device (i.e. series, brand, model, size, weight, and status - active or inactive).

4 Case Study: Services Composition in Healthcare for Patients with Overweight or Obesity

This section presents a case study to represent the services' orchestration within the IoT-based service composition language. Namely, the case study addresses the composition of healthcare services for overweight/obese patients. The scenario is as follows:

- An adult patient with overweight/obesity needs to monitor and coordinate the services provided by the wearable provider and those from a smart scale. Likewise, the patient needs external services that would help them achieve their goal of controlling or losing weight. Note that all the service's data are visualized by the patient in real time.

Fig. 1 represents the scenario explained above. As can be observed, the patient's medical parameters (i.e. burned calories, physical activity, heart rate, weight, BMI) are collected by a wearable device and a smart scale, which are synchronized using a smartphone.

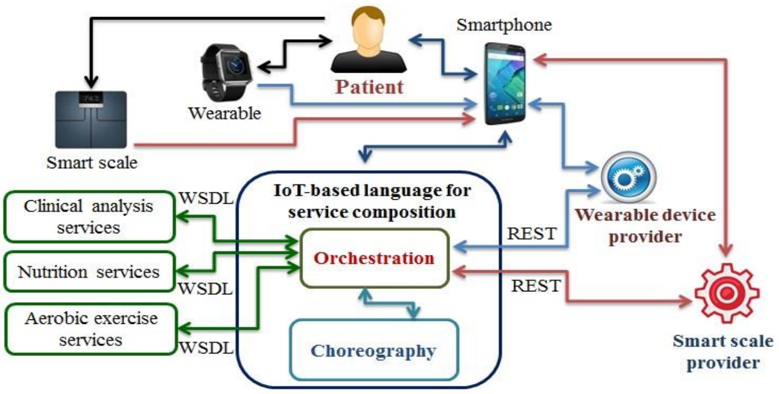

Fig. 1. Service orchestration for a patient with overweight or obesity.

The patient uses an application to visualize the services' composition. To perform the orchestration of the services (i.e. burned calories, physical activity, heart rate, weight, BMI), the SC mechanism relies on REST to request the medical parameters to both the wearable device provider and the smart scale provider. The SC mechanism does this to coordinate these services and establish an order of invocation, so that they can be visualized dynamically and in real time by the patient. Then, the external Web services (clinical analysis, nutrition services, and aerobic exercises) are orchestrationed. To this end, the SC mechanism relies on the WSDL to request information on each external Web service (e.g. availability, price, time, location). Then, using this information, the patient can select the most reliable clinical analysis laboratory, the best nutrition plan, and the most convenient type of aerobic exercise, according to their preferences and criteria.

The UML activity diagram of the previous scenario is depicted below in Fig. 2. Also, the figure shows how we mapped the labels for the services' orchestration to the language for IoT service composition.

- The <orchestrationIoT> tag is generated in the initial state of the diagram.
- The <definitionRoles> tag is generated according to the streets represented in the diagram (Language for IoT service composition, Web services, and REST-based Web service).
- The <varoc> tag is generated when the patient defines the amount of weight they want to lose.
- The <collaborations> tag is generated by identifying the smart devices and web services used to achieve weight loss or weight control.
- The <exceptions> tag is generated when exceptions occur (e.g. maintenance of the REST-based web service from the wearable provider).
- The <faultRecovery> tag is generated to handle exceptions or errors (e.g. incomplete data on the availability of web services).
- The <eventCtrl> tag is generated when the wearable device and the smart scale are concurrently working. This tag is represented in the diagram by the horizontal blue lines.

– The <invdoe>, <recdoe>, and <repdoe> tags are generated during the process of requesting the patient's medical parameters to the wearable device provider.
– The <deviceData> tag is generated during the process of receiving the patient's medical data.

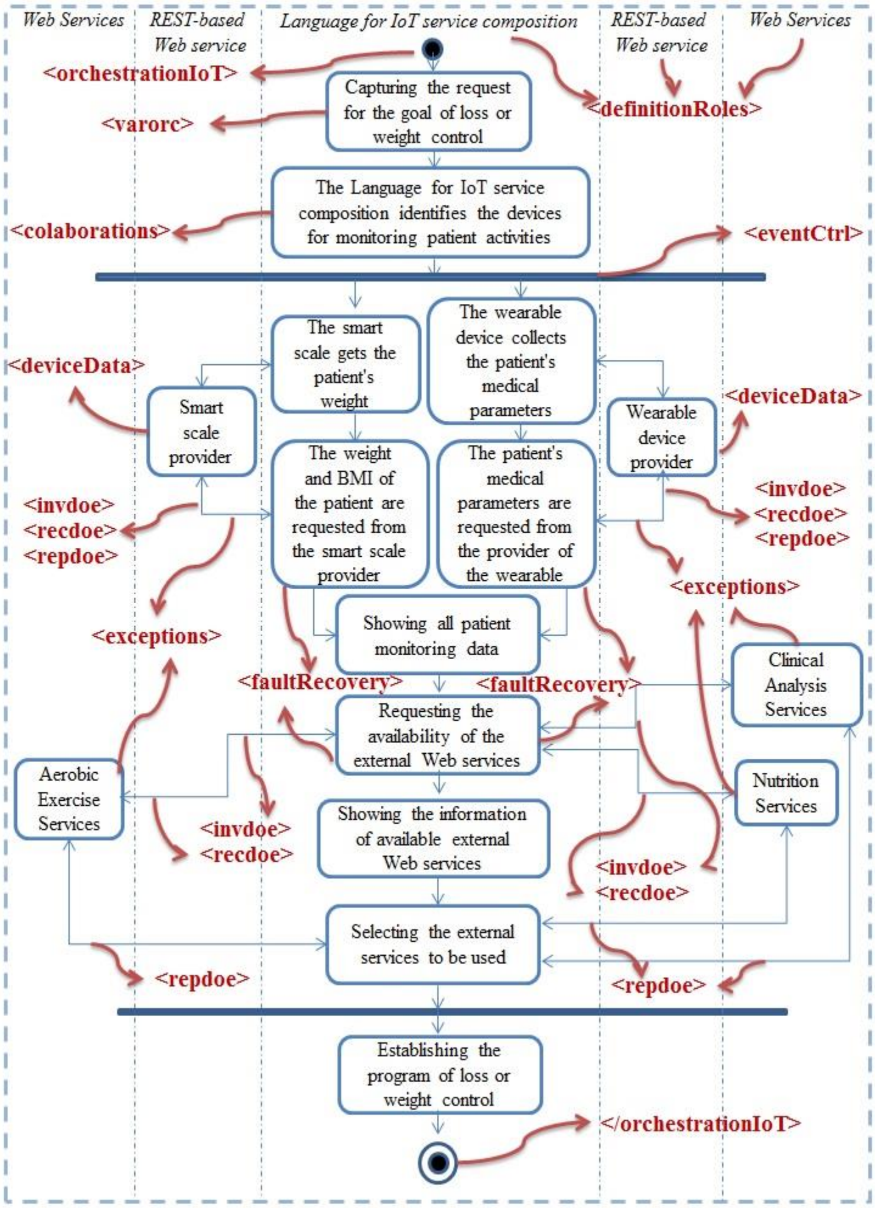

Fig. 2. UML-based activity diagram and the mapping process of the labels for the services orchestration

The language for IoT service composition is able to represent other scenarios, not only in the medical domain, but also in other contexts, such as home automation and the industrial sector.

5 Conclusions and Future Work

The increasing number of smart devices that nowadays communicate over the In-ternet has led to multiple heterogeneous devices connected in a network. Consequent-ly, there is a need for specific mechanisms to achieve the composition of all the ser-vices offered by these devices, and thus exploit the potential of the IoT. In this work, we propose the design of a language for IoT service composition. As a proof of con-cept, we present a case study for the orchestration of medical services for patients with overweight or obesity. The language requires all the involved smart devices to be intercommunicated through the Internet, and all the device providers to provide all the requested data.

As future work, we will seek to work on other case studies and thus expand the applicability of the language. Furthermore, we will intend to design the service choreography task and formalize the language to perform the composition of services in the IoT paradigm.

Acknowledgments. This work was supported by Tecnológico Nacional de México (TecNM) and sponsored by both the Mexico's National Council of Science and Technology (CONACYT) and the Secretariat of Public Education (SEP) through the PRODEP program (Programa para el Desarrollo Profesional Docente).

References

1. Luthra, S., Garg, D., Mangla, S.K., Berwal, Y.P.S.: Analizing challenges to Internet of Things (IoT) adoption and diffusion: An Indian context. Procedia Computer Science 15, pp. 733–739, doi: 10.1016/j.procs.2017.12.094 (2018)
2. Tokognon C.A., Gao B., Tian G.Y., Yan Y.: Structural Health Monitoring Framework Based on Internet of Things: A Survey. IEEE Internet of Things Journal 4, pp. 619–635, doi: 10.1109/JIOT.2017.2664072 (2017)
3. Pisching, M.A., Junquiera, F., Santos Filho, D.J., Miyagi, E.: Service Composition in the Cloud-Based Manufacturing Focused on the Industry 4.0. International Federation for Information Processing 450, pp. 65–72 (2015)
4. Yang, Z., Li, D.: IoT information service composition driven by user requirement. IEEE 17th International Conference on Computational Science and Engineering. Pp. 509–513, doi: 10.1109/CSE.2014.280 (2014)
5. Han, S.N., Khan, I., Lee, G.M., Crespi, N., Glitho, R.H.: Service composition for IP smart object using realtime Web protocols: Concept and research challenges. Computer Standards & Interfaces 43, pp. 79–90 (2016)
6. Dar, K., Taherkordi, A., Baraki, H., Eliassen, F., Geihs, K.: A resource oriented integration architecture for the Internet of Things: A business process perspective. Pervasive and Mobile Computing 20, pp. 145–159 (2015)

7. Ren, M., Ren, L., Jain, H.: Manufacturing service composition model based on synergy effect: Asocial network analysis approach. Applied Soft Computing 70, pp. 288–300, doi: https://doi.org/10.1016/j.asoc.2018.05.039 (2018)
8. Rapti, E., Karageorgos, A., Gerogiannis, V.C.: Decentralised Service Composition using Potential Fields in Internet of Things Applications. Procedia Computer Science 52, pp. 700– 06, doi: 10.1016/j.procs.2015.05.079 (2015)
9. Dijkman, R.M., Sprenkels, B., Peeters, T., Janssen, A.: Business models for the Internet of Things. International Journal of Information Management 35, pp. 672–678, doi: http://dx.doi.org/10.1016/j.ijinfomgt.2015.07.008 (2015)
10. Vidyasankar, K.: A Transaction Model for Executions of Compositions of Internet of Things Services. Procedia Computer Science 83, pp. 195–202 (2016)
11. Ju, J., Kim, M., Ahn, J.H.: Prototyping Business Models for IoT Service. Procedia Computer Science 91, pp. 882–890, doi: 10.1016/j.procs.2016.07.106 (2016)
12. Baker, T., Asim, M., Tawfik, H., Aldawsari, B., Buyya, R.: An energy-aware service composition algorithm for multiple cloud-based IoT applications. Journal of Network and Computer Applications 89, pp. 96–108 (2017)
13. Bergesio, L, Bernardos, A.M., Casar, J.R.: An Object-Oriented Model for Object Orchestration in Smart Environments. Procedia Computer Science 109C, pp. 440–447, doi: https://doi.org/10.1016/j.procs.2017.05.415 (2017)
14. Wen, Z., Yang, R., Garraghan, P., Lin, T., Xu, J., Rovatsos, M.: Fog Orchestration for Internet of Things Services. IEEE INTERNET COMPUTING. Pp. 16–24 (2017)
15. Macker, J.P., Taylor, I.: Orchestration and analysis of decentralized workflows within heterogeneous networking infrastructures. Future Generation Computer Systems 75, pp. 388–401, doi: http://dx.doi.org/10.1016/j.future.2017.01.007 (2017)
16. Duhart, C., Sauvage, P., Bertelle, C.: A Resource Oriented Framework for Service Choreography over Wireless Sensor and Actor Networks. International Journal of Wireless Information Networks, pp. 173–186, doi: 10.1007/s10776-016-0316-1 (2016)
17. Chen, l., Englund, C.: Choreographing services for smart cities: smart traffic demonstration. IEEE 85th Vehicular Technology Conference (VTC Spring), pp. 1–5, doi: 10.1109/VTCSpring.2017.8108625 (2017)
18. Pahl, C., Ioini, N.E., Helmer, S., Lee, B.: An Architecture Pattern for Trusted Orchestration in IoT Edge Clouds. 2018 Third International Conference on Fog and Mobile Edge Computing (FMEC), pp.1–8 (2018)