An Ontology-Based Reasoning Framework for Context-Aware Applications

Christoph Anderson(✉), Isabel Suarez, Yaqian Xu,
and Klaus David

Chair of Communication Technology, University of Kassel,
Wilhelmshöher Allee 73, 34121 Kassel, Germany
{anderson,isabel.suarez,david}@uni-kassel.de
yaqian.xu@comtec.eecs.uni-kassel.de
http://www.comtec.eecs.uni-kassel.de/

Abstract. Context-aware applications process context information to support users in their daily tasks and routines. These applications can adapt their functionalities by aggregating context information through machine-learning and data processing algorithms, supporting users with recommendations or services based on their current needs. In the last years, smartphones have been used in the field of context-awareness due to their embedded sensors and various communication interfaces such as Bluetooth, WiFi, NFC or cellular. However, building context-aware applications for smartphones can be a challenging and time-consuming task. In this paper, we describe an ontology-based reasoning framework to create context-aware applications. The framework is based on an ontology as well as micro-services to aggregate, process and represent context information.

Keywords: OWL · Android · Ontology · Context · Framework

1 Introduction

In the last few years, mobile phones have evolved from devices, used for voice communication and sending text messages only, to powerful smartphones with multiple embedded sensors and communication interfaces such as WiFi, Bluetooth, NFC or cellular. With their increasing processing capabilities and sensors such as accelerometers, gyroscopes, and magnetometers, smartphones are now being used for internet browsing, social networking, playing games, watching videos or listening to music [4]. Other fields of smartphone applications are activity recognition and context-awareness. In the area of activity recognition, for example, smartphones are used to recognize Activities of Daily Living (ADL) [13,15]. Context-aware applications can be utilized in the field of Ambient-Assisted-Living (AAL) and home automation. These types of applications monitor the environment to aggregate context information to provide recommendations or services to the user. Typically, this information is extracted

H. Christiansen et al. (Eds.): CONTEXT 2015, LNAI 9405, pp. 471–476, 2015.
DOI: 10.1007/978-3-319-25591-0_34

from sensor data, represented as time series, by using machine-learning and data processing algorithms. Existing frameworks in this field can be divided into two categories based on the way contexts are represented. The first category includes frameworks that describe context information without semantic e.g. as plain programming objects such as strings or class objects using object-oriented models [1,5,12,14]. However, these object-oriented models are not suitable for knowledge and data sharing in heterogeneous pervasive environments [2]. The second category consists of frameworks supporting semantic representations of context information [8,10,11]. These frameworks exploit ontologies, first-order logics or other description technologies to represent context information semantically. Building context-aware applications on smartphones that represent context information semantically as well as aggregate and process contexts through sensor information is still a challenging task.

In this paper, we present an ontology-based framework to create context-aware applications. In addition to the ontology, we extend the current state of the art frameworks by integrating micro-services to aggregate and process context information. By supporting reasoners such as Pellet [17], HermiT [6] and JFact [18], the framework can deduce complex contexts from already aggregated context information by using the reasoning paradigm of the Web-Ontology-Language (OWL). In the following sections, we present the architecture of our framework including a schematic overview over its components. A discussion about the limitations of the framework and a conclusion with a summary of future work is given at the end of this paper.

2 Framework-Architecture

The framework is based on an OWL ontology to model and represent context information. By using an ontology, complex contexts can be deduced by using the reasoning paradigm of OWL. In combination with the ontology, the framework exploits micro-services to aggregate and process context information from embedded smartphone or environmental sensors. Micro-services refer to standalone applications providing a background service only. These background services implement one or multiple functionalities such as context classification, prediction or sensor data aggregation, which can be used by applications or other micro-services. A schematic view of the framework architecture is given in Fig. 1.

2.1 Micro-Services

The micro-service architectural style is a mechanism, where a dedicated application provides one or multiple services, each one of them running in its process. In our framework, services communicate over an Inter-Process-Communication interface (IPC) with other services or the framework core. Figure 1 depicts three possible types of micro-services. Since the framework has to exploit sensors to aggregate and process sensor data, the first type of micro-services (*Sensing*) is

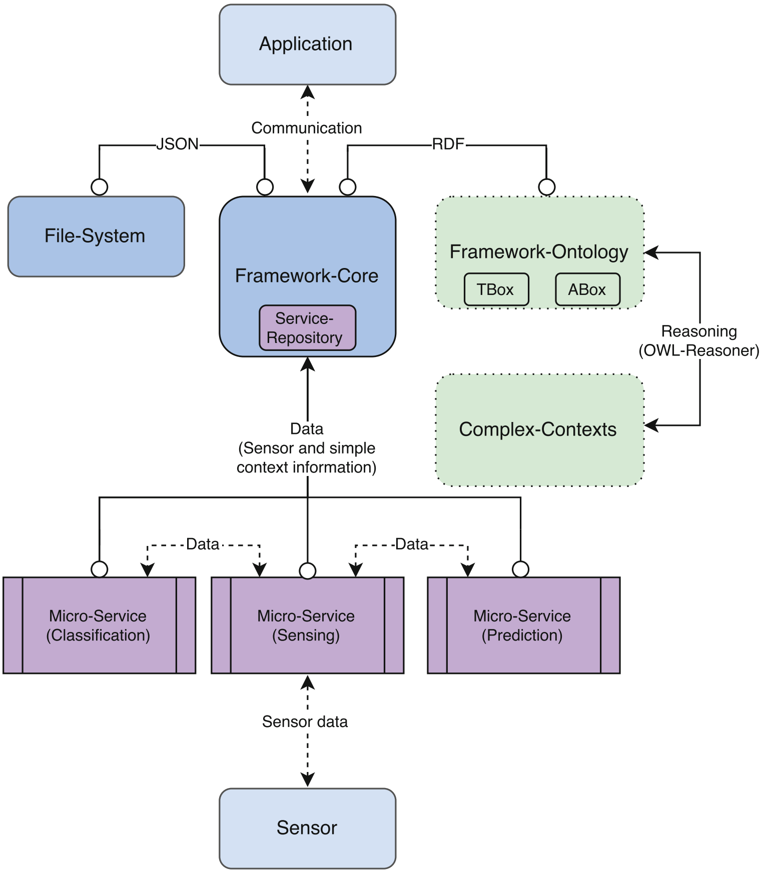

Fig. 1. Schematic view of the framework architecture

used to establish a connection to different sensors. Either embedded smartphone sensors such as the accelerometer, gyroscope or magnetometer or environmental sensors such as movement sensors or door contacts can be used. Establishing a connection to environmental sensors, for example, can be done by implementing a micro-service requesting a web-service abstracting sensor information e.g. in a smart home environment. Already aggregated sensor data can be distributed to other micro-services or the framework core by IPC. The second type of micro-services (*Classification*) is used to classify context information from sensor data. Existing approaches such as [9] can be implemented as services to classify context information such as standing, sitting, walking, and lying. The third type of micro-services (*Prediction*) is used for context prediction. Generally, existing context prediction algorithms such as [16] or [7] exploit context histories to build prediction models. These algorithms predict future contexts providing further information about the next context of the user. To support context prediction algorithms, the framework provides access to already aggregated context information in form of a context history stored on the smartphone's file-system.

2.2 Framework Core

The framework core is the central component of the framework. It maintains the ontology and provides an interface to establish IPC connections to installed micro-services. The interface controls the life-cycle of micro-services. For this reason, the core maintains a service-repository of installed micro-services in the form of a database. The repository is needed to establish connections to installed services. If the user installs or removes a service, the framework core updates the corresponding entry in the database. Additionally, the framework core process information from micro-services such as aggregated contexts, raw sensor data or context predictions. While most context classification or prediction approaches rely on sensor or context histories to classify or predict new context information, the framework core saves this information on the smartphone's local file-system. Already classified contexts are added to the ontology.

2.3 Framework Ontology

The framework uses an OWL ontology to model and represent context information. The ontology describes the terminology of the context model in the form of conceptual classes and relationships between these classes. These classes and relations define the structure of the underlying model and are stored inside a terminological box (TBox). Instances of conceptual classes, their attributes and relationships are stored inside an assertional box (ABox). With the reasoning paradigm of OWL ontologies, the framework can deduce complex contexts from already aggregated information. In a smart environment, for example, the ontology can infer that a user is making coffee by using dense sensing [3].

3 Limitations

An experimental evaluation of the framework was carried out by implementing a context-based communication filter for Android based smartphones. The filter blocks certain communication such as phone calls, text messages or emails by aggregating and processing context information of the user in a smart home environment located in our department. Similar to the evaluation in [19], we experienced memory and performance issues on the tested smartphones (Google Nexus 4, 5 and 6) when reasoning larger ontologies[1]. Besides memory and performance issues, applications using the framework may have time constraints when deducing new context information from already aggregated contexts. These time constraints may be violated when using reasoners that are not optimized for large

[1] More than 6.000 Axioms.

ontologies or when using power-limited smartphones. Besides the limitations caused by different ontology sizes and reasoners being used, efforts regarding the privacy and security of context information have to be made. Although, context information is stored on the smartphone's local file-system, the exchange of data e.g. using environmental sensors, has to be secured by using current state of the art encryption technologies such as TLS (Transport Layer Security).

4 Conclusion

Context-aware applications process context information to support users in their daily tasks and routines. Building context-aware applications for smartphones that represent context information semantically as well as aggregate and process contexts through smartphone or environmental sensor is still a challenging task. In this paper, we presented an ontology-based framework to create context-aware applications. By utilizing an ontology, context information can be described semantically. In addition to the ontology, we implemented micro-services to aggregate and process context information from embedded smartphone or environmental sensors. By utilizing reasoners such as Pellet, HermiT or JFact, the framework can deduce complex contexts from already aggregated context information by using the reasoning paradigm in OWL. In the future, we plan to publish the framework, making it available to the community. Also, we plan to take into account semantic rule languages such as SWRL (Semantic-Web-Rule-Language), as well as other reasoners to extend the functionalities of the framework.

Acknowledgements. This work has been co-funded by the Social Link Project within the Loewe Program of Excellence in Research, Hessen, Germany.

References

1. Carlson, D., Schrader, A.: Dynamix: an open plug-and-play context framework for android. In: IOT, pp. 151–158. IEEE (2012)
2. Chen, H., Perich, F., Finin, T., Joshi, A.: Soupa: standard ontology for ubiquitous and pervasive applications. In: The First Annual International Conference on Mobile and Ubiquitous Systems: Networking and Services, 2004, MOBIQUITOUS 2004, pp. 258–267, August 2004
3. Chen, L., Hoey, J., Nugent, C., Cook, D., Yu, Z.: Sensor-based activity recognition. IEEE Trans. Syst. Man Cybern. Part C Appl. Rev. **42**(6), 790–808 (2012)
4. Falaki, H., Mahajan, R., Kandula, S., Lymberopoulos, D., Govindan, R., Estrin, D.: Diversity in smartphone usage. In: Proceedings of the 8th International Conference on Mobile Systems, Applications, and Services, MobiSys 2010, pp. 179–194. ACM, New York (2010). http://doi.acm.org/10.1145/1814433.1814453
5. Ferreira, D., Kostakos, V., Dey, A.K.: Aware: mobile context instrumentation-framework. Front. ICT **2**(6), 1–9 (2015)
6. Glimm, B., Horrocks, I., Motik, B., Stoilos, G., Wang, Z.: Hermit: an owl 2 reasoner. J. Autom. Reasoning **53**(3), 245–269 (2014). http://dx.doi.org/10.1007/s10817-014-9305-1

7. Gopalratnam, K., Cook, D.J.: Active lezi: an incremental parsing algorithm for sequential prediction. Int. J. Artif. Intell. Tools **13**(04), 917–929 (2004). http://www.worldscientific.com/doi/abs/10.1142/S0218213004001892
8. Gu, T., Pung, H.K., Zhang, D.: Toward an osgi-based infrastructure for context-aware applications. IEEE Pervasive Comput. **3**(4), 66–74 (2004)
9. Kwapisz, J.R., Weiss, G.M., Moore, S.A.: Activity recognition using cell phone accelerometers. SIGKDD Explor. Newsl. **12**(2), 74–82 (2011). http://doi.acm.org/10.1145/1964897.1964918
10. Meditskos, G., Dasiopoulou, S., Efstathiou, V., Kompatsiaris, I.: Sp-act: a hybrid framework for complex activity recognition combining owl and sparql rules. In: 2013 IEEE International Conference on Pervasive Computing and Communications Workshops (PERCOM Workshops), pp. 25–30, March 2013
11. Paspallis, N., Papadopoulos, G.A.: A pluggable middleware architecture for developing context-aware mobile applications. Pers. Ubiquit. Comput. **18**(5), 1099–1116 (2014). http://dx.doi.org/10.1007/s00779-013-0722-7
12. Raento, M., Oulasvirta, A., Petit, R., Toivonen, H.: Contextphone: a prototyping platform for context-aware mobile applications. Pervasive Comput. IEEE **4**(2), 51–59 (2005)
13. Roy, N., Misra, A., Cook, D.: Infrastructure-assisted smartphone-based adl recognition in multi-inhabitant smart environments. In: 2013 IEEE International Conference on Pervasive Computing and Communications (PerCom), pp. 38–46. IEEE Computer Society, March 2013
14. Sridevi, S., Bhattacharya, S., Pitchiah, R.: Context aware framework. In: Sénac, P., Sénac, P., Seneviratne, A., Seneviratne, A., Ott, M., Ott, M. (eds.) MobiQuitous 2010. LNICST, vol. 73, pp. 358–363. Springer, Heidelberg (2012). http://dx.doi.org/10.1007/978-3-642-29154-8_40
15. Shoaib, M., Scholten, H., Havinga, P.: Towards physical activity recognition using smartphone sensors. In: 2013 IEEE 10th International Conference on Ubiquitous Intelligence and Computing, Autonomic and Trusted Computing (UIC/ATC), pp. 80–87, December 2013
16. Sigg, S., Haseloff, S., David, K.: An alignment approach for context prediction tasks in ubicomp environments. Pervasive Comput. IEEE **9**(4), 90–97 (2010)
17. Sirin, E., Parsia, B., Grau, B.C., Kalyanpur, A., Katz, Y.: Pellet: a practical owl-dl reasoner. Web Semant. Sci. Serv. Agents World Wide Web **5**(2), 51–53 (2007). http://www.sciencedirect.com/science/article/pii/S1570826807000169. software Engineering and the Semantic Web
18. Tsarkov, D., Tsarkov, D., Horrocks, I., Horrocks, I.: FaCT++ description logic reasoner: system description. In: Furbach, U., Furbach, U., Shankar, N., Shankar, N. (eds.) IJCAR 2006. LNCS (LNAI), vol. 4130, pp. 292–297. Springer, Heidelberg (2006)
19. Yus, R., Bobed, C., Esteban, G., Bobillo, F., Mena, E.: Android goes semantic: Dl reasoners on smartphones. In: Bail, S., Glimm, B., Gonçalves, R.S., Jiménez-Ruiz, E., Kazakov, Y., Matentzoglu, N., Parsia, B. (eds.) ORE. CEUR Workshop Proceedings, vol. 1015, pp. 46–52 (2013). CEUR-WS.org