Programming Agent-based Mobile Apps: The JaCa-Android Framework

JAAMAS Track

Angelo Croatti
Alma Mater Studiorum – Università di Bologna
a.croatti@unibo.it

Alessandro Ricci
Alma Mater Studiorum – Università di Bologna
a.ricci@unibo.it

ABSTRACT

A relevant application domain for agent-based software is given by smart mobile applications. In this context, the impressive progress of technologies makes it possible to explore the use of agent-oriented programming languages and frameworks based on cognitive architectures. Accordingly, here we provide a description of JaCa-Android, a framework based on the JaCaMo platform that allows for designing and programming smart mobile apps using BDI-based cognitive agents within the A&A conceptual model.

KEYWORDS

Agents; BDI; Android; JaCaMo; Personal Agents

ACM Reference Format:
Angelo Croatti and Alessandro Ricci. 2021. Programming Agent-based Mobile Apps: The JaCa-Android Framework: JAAMAS Track. In *Proc. of the 20th International Conference on Autonomous Agents and Multiagent Systems (AAMAS 2021), Online, May 3–7, 2021*, IFAAMAS, 3 pages.

1 INTRODUCTION

In the literature, relevant application domains for agent-based technologies and multi-agent systems (MAS) require the possibility to run software agents on mobile and wearable devices [15, 18, 19, 23]. Main examples include personal assistant agents [20], context-aware applications [3] and smart environments [26]. From an engineering point of view, the availability of frameworks that make it possible to exploit agent-oriented programming [27] and multi-agent programming [8, 9] to this purpose is a relevant issue. In fact, this would make it possible to exploit the level of abstraction and the effectiveness of agent programming languages and frameworks for these applications.

In this paper we present the approach used by JaCa-Android, an open-source agent-oriented framework[1] to develop agent-based mobiles apps. The JaCa-Android framework is based on the JaCaMo platform [6, 7]. Briefly, using JaCa-Android, a mobile app is designed as a personal assistant agent, based on the belief-desire-intention (BDI) architecture [30] and using the Agents and Artifacts (A&A) conceptual model [22] to design the mobile application environment and the user context. In the simplest case, a JaCa-Android mobile app can be designed and programmed as a cognitive BDI agent equipped with an environment providing the facilities to manage the user interface (UI), to access the user context and the device context, as well as to exploit the whole Android framework and services. Beyond the simplest case, the framework allows scaling in line with mobile app complexity, both by allowing the design of the app as a full-fledged JaCaMo MAS and by allowing for the integration of the mobile app with larger distributed multi-agent systems, eventually running on different nodes.

2 MOBILE APPS AS AGENTS

The main objective of the JaCa-Android framework is to provide an effective agent-oriented approach in designing and programming mobile apps, to support the development of agent-based applications on mobile (and wearable) devices. Among others, main features of JaCa-Android are:

- to provide a proper (agent-oriented) level of abstraction in modelling and exploiting the mobile environment, hiding as much as possible low-level details concerning the Android context yet retaining the full power of its functionalities;
- to scale with complexity, that is provide an API that would make it possible to implement simple tasks/programs by means of simple agents, and introducing complexity (multiple agents, complex environments) incrementally, by need;
- to support an easy and flexible integration with existing libraries/services/technologies available in the host (mobile) environment;
- to support a seamless integration within distributed MAS applications, running on heterogeneous systems communicating by means of the Internet network (so that mobile and wearable devices are one of the different subsystems).

To deal with these features, in JaCa-Android a mobile app is designed as a cognitive BDI agent explicitly conceived as a *personal assistant* of the user (despite the specific job/task/application logic of the app). Accordingly, the personal assistant agent can be designed in terms of goals, beliefs, and plans. To interact with the user and with the external world and to accomplish its goals, JaCa-Android provides the agent with a toolbox of predefined *artifacts* – besides the ones that the developer can implement using the CArtAgO programming model [24] — a trivial example is reported in Figure 1. Depending on the specific app needs, the agent can dynamically instantiate and work with a proper set of artifacts.

User Interaction and Context. The first aspect to consider when designing a personal assistant agent is how the agent can observe and interact with the user. At the basic level, the agent must be able to perceive/observe the *user inputs* and the *user context*, as well as to *communicate* with him. The user is part of the environment that the agent perceives. In our case, a developer needs strong flexibility in choosing what kind of user input/context could be

[1] Available here: `https://github.com/pslabunibo/jaca-android`

Proc. of the 20th International Conference on Autonomous Agents and Multiagent Systems (AAMAS 2021), U. Endriss, A. Nowé, F. Dignum, A. Lomuscio (eds.), May 3–7, 2021, Online.

```
!setup.

+!setup
 <- makeArtifact("gps", "GPSService", Id);
    focus(Id);
    makeArtifact("sms", "SMSService", _).

+pos(Lat,Lon) : nearbyHome(Lat,Lon,Res) & Res == True
 <- sendSMS("Arrived at home.").
```

Figure 1: A trivial example of an app programmed as agent.

relevant depending on the specific application. In our case, this flexibility is achieved by means of artifacts, that is: the agent can instantiate the proper artifacts, eventually at runtime, depending on what it needs to perceive. The agent-artifact interaction model mirrors the agent action and perception capabilities. An artifact provides a set of operations that can be conceived as actions on the agent side. So, by instantiating an artifact, the set of actions that the BDI agent can do is extended with the set of operations provided by that artifact.

Integration with the Device Context and Application Services. Like in the case of the user context, also device context and further application services provided by the mobile environment – being either from the Android framework itself or from applications installed on the device – can be modeled and accessed by properly designed artifacts. To interact with existing application services, artifacts can be used to wrap and hide low-level platform-dependent code for enabling interaction between different apps and make it possible to exploit services provided by one app from another app.

Scaling with Complexity. In JaCa-Android is possible to design the mobile app as a multi-agent system, involving more than one agent running on the mobile device, eventually sharing the mobile environment: this is obtainable by spawning the mobile app as a full JaCaMo system. Also, it is possible to consider the mobile app and the personal assistant agent as part of a larger distributed multi-agent system. In this case, JaCa-Android allows the exploitation of the JaCaMo distributed infrastructure support that allows for an agent running on some device/node – so in its own MAS – to dynamically join and interact with an existing JaCaMo-based MAS.

3 JACA-ANDROID ARCHITECTURE

The JaCa-Android framework runs on top of a JaCaMo runtime, extending it with the required abstractions to make it possible the design and the execution of MASs over the Android platform. In Figure 2 an abstract representation of the complete framework is reported, starting from the Android Framework. The JaCaMo runtime actually is based on Jason runtime – running the BDI agents – and CArtAgO runtime – managing the artifact-based environment [6, 7]. Like in the case of JaCaMo, the JaCa-Android runtime is fully Java-based and runs as a normal Android app.

Generally speaking, the JaCa-Android framework can be used as a general purpose framework for developing and running multi-agent systems on the top of the Android platform. Besides this general view, the framework has been explicitly designed to facilitate the development of personal assistant agents, running on mobile (and wearable) devices. To this purpose, the framework

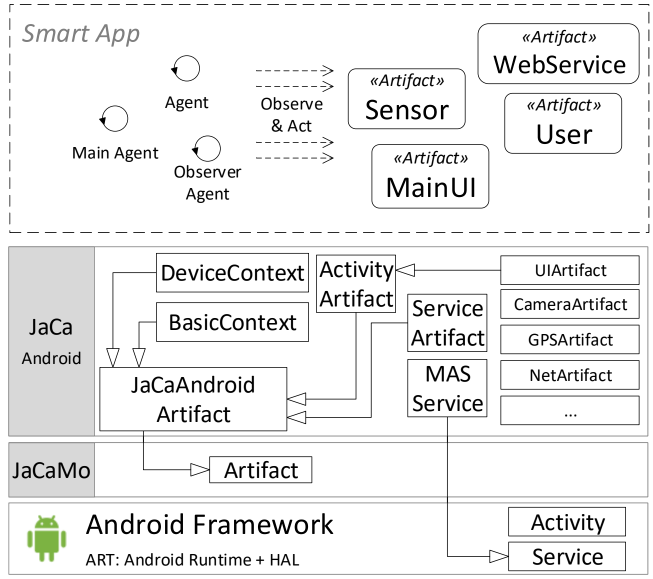

Figure 2: JaCa-Android coarse-grained architecture.

provides a layer of artifacts with functionalities useful to support context-awareness. Two main examples are `BasicContext` and `DeviceContext` artifacts. The first one allows agents to get information about the user context, including where the user is located (i.e. the geo location obtained by the GPS), what the user is looking at (obtained by the device camera), to mention two main ones. The second one allows agents to observe changes in device sensors values, e.g., changes in the orientation (given by the accelerometer or the gyroscope), changes related to the light sensor, and so on.

4 RELATED WORKS

In the literature, several works have been proposed to enable the development of general-purpose mobile apps/systems using existing and well-known agent programming platforms. Agent-based technologies applied in the context of mobile computing started with *mobile agent* (MA) technologies [4, 10, 14, 16, 28]. First concrete examples in agent literature include 3APL-M [17]. In the context of the development of intelligent agents on mobile devices, we can cite the Agent Factory Micro Edition (AFME) [21] and the ANDROMEDA platform [1, 2]. Also, another attempt to combine agents and Android is the JADE-LEAP project [5, 29] Finally, a work for porting the ASTRA agent-oriented language on the Android platform [25] is available for developing context-sensitive apps.

5 FINAL REMARKS

This paper offers a glimpse of the JaCa-Android framework in terms of (1) the approach proposed for the design of agent-based mobile apps, and (2) its internal architecture. This is an extended abstract of the full paper published at JAAMAS [13]. Finally, it is worth clarifying that JaCa-Android has been currently adopted in developing real-world case studies, both in the Healthcare domain [11] and Mixed Reality contexts [12], for evaluation purposes.

REFERENCES

[1] Jorge Agüero, Miguel Rebollo, Carlos Carrascosa, and Vicente Julián. 2008. Developing intelligent agents on the Android platform. In *Sixth European Workshop on Multi-Agent Systems (EUMAS 2008)*. 1–14.

[2] Jorge Agüero, Miguel Rebollo, Carlos Carrascosa, and Vicente Julián. 2009. Does Android Dream with Intelligent Agents?. In *International Symposium on Distributed Computing and Artificial Intelligence 2008 (DCAI 2008)*, Juan M. Corchado, Sara Rodríguez, James Llinas, and José M. Molina (Eds.). Springer Berlin Heidelberg, Berlin, Heidelberg, 194–204.

[3] Matthias Baldauf, Schahram Dustdar, and Florian Rosenberg. 2007. A Survey on Context Aware Systems. *Int. J. Ad Hoc Ubiquitous Comput.* 2, 4 (June 2007), 263–277.

[4] P. Bellavista, A. Corradi, and C. Stefanelli. 2001. Mobile agent middleware for mobile computing. *Computer* 34, 3 (March 2001), 73–81.

[5] Federico Bergenti, Giovanni Caire, and Danilo Gotta. 2014. Agents on the Move: JADE for Android Devices.. In *WOA 2014 (CEUR Workshop Proceedings, Vol. 1260)*, Corrado Santoro and Federico Bergenti (Eds.). CEUR-WS.org.

[6] Olivier Boissier, Rafael H Bordini, Jomi F. Hübner, and Alessandro Ricci. 2020. *Multi-Agent Oriented Programming: Programming Multi-Agent Systems Using JaCaMo.* MIT Press.

[7] Olivier Boissier, Rafael H. Bordini, Jomi F. Hübner, Alessandro Ricci, and Andrea Santi. 2013. Multi-agent oriented programming with JaCaMo. *Science of Computer Programming* 78, 6 (2013), 747–761.

[8] Rafael H. Bordini, Mehdi Dastani, Jürgen Dix, and Amal El Fallah Seghrouchni (Eds.). 2005. *Multi-Agent Programming: Languages, Platforms and Applications.* Number 15 in Multiagent Systems, Artificial Societies, and Simulated Organizations. Springer.

[9] Rafael H. Bordini, Mehdi Dastani, Jürgen Dix, and Amal El Fallah Seghrouchni (Eds.). 2009. *Multi-Agent Programming: Languages, Tools and Applications.* Springer.

[10] M. Breugst, S. Choy, L. Hagen, M. Höft, and T. Magedanz. 1999. *Grasshopper — An Agent Platform for Mobile Agent-Based Services in Fixed and Mobile Telecommunications Environments.* Springer Berlin Heidelberg, Berlin, Heidelberg, 326–357.

[11] Angelo Croatti, Sara Montagna, Alessandro Ricci, Emiliano Gamberini, Vittorio Albarello, and Vanni Agnoletti. 2019. BDI Personal Medical Assistant Agents: The Case of Trauma Tracking and Alerting. *Artificial Intelligence in Medicine* 96 (May 2019), 187 – 197.

[12] Angelo Croatti and Alessandro Ricci. 2018. A Model and Platform for Building Agent-Based Pervasive Mixed Reality Systems. In *Advances in Practical Applications of Agents, Multi-Agent Systems, and Complexity: The PAAMS Collection*, Yves Demazeau, Bo An, Javier Bajo, and Antonio Fernández-Caballero (Eds.). Springer International Publishing, Cham, 127–139.

[13] Angelo Croatti and Alessandro Ricci. 2020. Mobile Apps as Personal Assistant Agents: the JaCa-Android Framework for programming Agents-based applications on mobile devices. *Autonomous Agents and Multi-Agent Systems* 34, 2 (July 2020). https://doi.org/10.1007/s10458-020-09474-7

[14] Amal El Fallah Seghrouchni and Alexandru Suna. 2005. *Claim and Sympa: A Programming Environment for Intelligent and Mobile Agents.* Springer US, Boston, MA, 95–122.

[15] Martin Griss, Reed Letsinger, Dick Cowan, Michael Van Hilst, and Robert Kessler. 2002. *CoolAgent: Intelligent Digital Assistants for Mobile Professionals - Phase 1 Retrospective.* Technical Report. HPL Technical Report.

[16] Dag Johansen, Kåre J. Lauvset, Robbert van Renesse, Fred B. Schneider, Nils P. Sudmann, and Kjetil Jacobsen. 2002. A TACOMA Retrospective. *Softw. Pract. Exper.* 32, 6 (May 2002), 605–619.

[17] Fernando Koch. 2005. 3APL-M Platform for Deliberative Agents in Mobile Devices. In *Proceedings of the Fourth International Joint Conference on Autonomous Agents and Multiagent Systems* (The Netherlands) *(AAMAS '05)*. ACM, New York, NY, USA, 153–154.

[18] Fernando Koch and Iyad Rahwan. 2004. The Role of Agents in Intelligent Mobile Services. In *Proc. of the 7th Pacific Rim International Workshop on Multi-Agents, PRIMA 2004* (Auckland, New Zealand), Vol. 3371. 115–127.

[19] Fernando Luiz Koch and Iyad Rahwan. 2004. Classification of agents-based mobile assistants. In *Proceedings of the AAMAS Workshop on Agents for Ubiquitous Computing (UbiAgents)* (New York, USA).

[20] Pattie Maes. 1994. Agents That Reduce Work and Information Overload. *Commun. ACM* 37, 7 (July 1994), 30–40.

[21] C. Muldoon, G. M. P. O'Hare, R. Collier, and M. J. O'Grady. 2006. Agent Factory Micro Edition: A Framework for Ambient Applications. In *Computational Science – ICCS 2006*, Vassil N. Alexandrov, Geert Dick van Albada, Peter M. A. Sloot, and Jack Dongarra (Eds.). Springer Berlin Heidelberg, Berlin, Heidelberg, 727–734.

[22] Andrea Omicini, Alessandro Ricci, and Mirko Viroli. 2008. Artifacts in the A&A meta-model for multi-agent systems. *Autonomous Agents and Multi-Agent Systems* 17 (3) (Dec. 2008).

[23] Anand Ranganathan and Roy H. Campbell. 2003. A Middleware for Context-aware Agents in Ubiquitous Computing Environments. In *Proceedings of the ACM/IFIP/USENIX 2003 International Conference on Middleware* (Rio de Janeiro, Brazil) *(Middleware '03)*. Springer-Verlag, Berlin, Heidelberg, 143–161.

[24] Alessandro Ricci, Michele Piunti, and Mirko Viroli. 2011. Environment Programming in Multi-agent Systems: An Artifact-based Perspective. *Autonomous Agents and Multi-Agent Systems* 23, 2 (Sept. 2011), 158–192.

[25] S. Russell, O. Doyle, and R. W. Collier. 2017. Developing Android applications using agent-oriented programming. In *2017 12th International Conference on Intelligent Systems and Knowledge Engineering (ISKE)*. 1–7.

[26] Joao Santos, Joel J.P.C. Rodrigues, Bruno M.C. Silva, Joao Casal, Kashif Saleem, and Victor Denisov. 2016. An IoT-based mobile gateway for intelligent personal assistants on mobile health environments. *Journal of Network and Computer Applications* 71 (2016), 194 – 204.

[27] Yoav Shoham. 1993. Agent-Oriented Programming. *Artificial Intelligence* 60 (1993), 51–92.

[28] Alexandru Suna and Amal El Fallah-Seghrouchni. 2005. A Mobile Agents Platform: Architecture, Mobility and Security Elements. In *Programming Multi-Agent Systems*, Rafael H. Bordini, Mehdi Dastani, Jürgen Dix, and Amal El Fallah Seghrouchni (Eds.). Springer Berlin Heidelberg, Berlin, Heidelberg, 126–146.

[29] Yu Weihong and Yan Chen. 2013. The Development of Jade Agent for Android Mobile Phones. In *Proceedings of the 2012 International Conference on Information Technology and Software Engineering*, Wei Lu, Guoqiang Cai, Weibin Liu, and Weiwei Xing (Eds.). Springer Berlin Heidelberg, Berlin, Heidelberg, 215–222.

[30] Michael Wooldridge and Nicholas R. Jennings. 1995. Intelligent agents: theory and practice. *The Knowledge Engineering Review* 10, 2 (1995), 115–152.