A Multi-Agent System for Resource Privacy: Deployment of Ambient Applications in Smart Environments

(Extended Abstract)

Ferdinand Piette[1,2], Costin Caval[1],
Amal El Fallah Seghrouchni[1], Patrick Taillibert[1] and Cédric Dinont[2]
[1] Sorbonne Universités, UPMC Univ Paris 06, LIP6, Paris, France
{costin.caval, amal.elfallah, patrick.taillibert}@lip6.fr
[2] Institut Supérieur de l'Électronique et du Numérique, Lille, France
{ferdinand.piette, cedric.dinont}@isen-lille.fr

ABSTRACT

In this paper, we present Multi-Agent Systems (MAS) as a well-adapted paradigm for designing software for the configuration, deployment and monitoring of distributed applications in the domain of Ambient Intelligence (AmI). We show how privacy is enhanced by hiding information using the agent architecture and organisation. We introduce privacy policies that can allow or prevent the sharing of resource information. This results in privacy by design.

Keywords

Multi-agent system; Ambient Intelligence; Privacy management; Deployment; Goal-driven agents

1. DEPLOYMENT OF APPLICATIONS

Ambient Intelligence (AmI) research focuses on the improvement of human interactions with smart applications and proposes frameworks and platforms that facilitate the development of context-aware and dynamic applications. In this domain, it is often assumed that an underlying interoperable hardware and energy infrastructure already exists. Meanwhile, the main challenge of the Internet of Things (IoT) is to achieve full interoperability of interconnected devices while enabling advanced services and guaranteeing the trust, privacy and security of communications [1]. Because of the heterogeneity of such systems, it is difficult to have horizontal communication between connected devices. Present applications using connected devices on the IoT infrastructure are vertically connected from the device to a server that collects the data. This approach raises privacy questions and inevitably leads to a Big Brother effect: the user does not own his data any more. To allow horizontal connections between devices, we need mechanisms that reason on the heterogeneity of systems in order to automatically deploy smart applications provided by AmI on an existing hardware infrastructure provided by the IoT. To address these issues, we propose a distributed approach to solve the deployment problem using a multi-agent architecture which, as will be seen, helps to ensure resource privacy.

Appears in: *Proceedings of the 15th International Conference on Autonomous Agents and Multiagent Systems (AAMAS 2016), J. Thangarajah, K. Tuyls, C. Jonker, S. Marsella (eds.), May 9–13, 2016, Singapore.*

2. MULTI-AGENT MODELLING

We identify several necessary specificities to build the deployment software. It has to dynamically deploy and undeploy distributed AmI applications in an environment that is also dynamic. It also has to manage the context, like the location of the user, in order to choose the most relevant devices to deploy applications. Resource and information privacy is an important requirement in the AmI domain. At last, autonomy and robustness of the system are very important specificities. If a part of the system failed, the other parts should continue to work normally. To solve the dynamic deployment problem, we extend our previous work [4] where the available hardware infrastructure and the requirements of the deployable applications are described by graphs. Nodes represent hardware entities or relations between these entities and properties can be attached to each node. A graph matching algorithm can then be used on the available infrastructure graph to find the entities that can support the running of the application. However, this previous solution was centralised, which makes it unsuitable for real systems that need to take into consideration, among others, privacy and scalability. For the other specificities (context management, autonomy, robustness and privacy) of the deployment software, we identified Multi-Agent Systems (MAS) as a suitable solution. This paradigm possesses good properties to facilitate a local processing of the data, guarantee the autonomy of the different parts of the hardware infrastructure and so handle some aspects of privacy. The context management, autonomy and robustness are well studied in the agent literature [3]. In this paper, we focus on the encapsulation of resource privacy, using the restriction of information sharing and agent organisation.

2.1 Classes of Agent

The system is made of four classes of agent:

- An ***Infrastructure Agent*** deals with a part of the global hardware infrastructure which is represented as a graph [4] that is never shared with other agents. The agent uses a graph-matching algorithm to propose complete or partial solutions for the deployment of applications. This class of agent has several goals:

keep the infrastructure graph up to date; propose solutions for the deployment of applications, considering the available hardware infrastructure, but also the sharing and privacy policy (see Sec. 2.2); and deploy or undeploy functionalities of an application.

- An ***Infrastructure Super Agent*** is a representative of a group of *Infrastructure Agents*, acting as a proxy between the group and the other agents.
- An ***Application Agent*** manages an entire application during its runtime. It has a graph-based description of the application. The goals of this class of agent are to guarantee the consistency of the application and deploy or undeploy functionalities of the application if necessary. The functionalities of the application can be deployed on several parts of the infrastructure, managed by associated *Infrastructure Agents* with which the *Application Agent* will need to interact.
- At last, the ***User Agent*** is the interface between the agents of the deployment software and the user. This one can request the deployment or undeployment of applications.

In order to enhance the resource privacy, the agents only have a local view of the system. Indeed the graph representation of the available hardware infrastructure managed by an *Infrastructure Agent* is only known by this agent and is never shared with others. Moreover, the architecture used helps to keep a clear separation between the users, interacting with the associated *User Agents*, the applicative part, managed by the *Application Agents*, and the hardware part, monitored by the *Infrastructure Agents*.

2.2 Organisation and Interactions

Infrastructure Agents can be grouped behind an *Infrastructure Super Agent* which, as stated before, acts as a proxy for the agents of the group. From an outside view, this *Infrastructure Super Agent* is seen as a normal *Infrastructure Agent*, resulting in a multi-scale organisation that helps to improve privacy. It is then easier to abstract groups of agents and make them invisible from the outside.

To improve resource privacy, we also introduced sharing policies to regulate the access of the *User Agents* (and implicitly their users) to the infrastructure. These policies include completely blocking the access by the *Infrastructure Agent* for unknown users. Otherwise, the authorization levels can be, for example : (1) Administrator: the agent has full access to the resources proposed by the *Infrastructure (Super) Agent*, can manage the authorisation levels, reconfigure the *Super Agent* and add or remove *Infrastructure Agents* to the *Super Agent*; (2) Regular user: the agent has access to the resources of the *Infrastructure (Super) Agent* but it cannot reconfigure authorisation levels or agent organisation and; (3) Guest: the agent has a restricted access to the resources. Only the resources considered as non critical by an administrator can be shared. These authorisation levels are not limited to three and can be modified by one of the administrators of the *Infrastructure Agent*. As the *Application Agents* are created by *User Agents*, they both share the same authorisation level.

2.3 Design and Implementation

We used goal-driven agents for gains in the autonomy and robustness of the application. The goal-based representation adopted is the Goal-Plan Separation (GPS) approach [2]. This approach helped handle agent complexity through a multi-level description, from top level abstract behaviours with *goals plans* that describe relationships between goals, to low level *action plans* that detail the sequences of concrete actions.

We developed a demonstration model of the deployment software based on an apartment replica in which we executed various scenarios applied to home care for dependent persons. These scenarios use commercial connected devices tweaked to be horizontally connected, thanks to the deployment software. This realisation helps us asses the difficulties of handling the heterogeneity of hardware entities. We now have a concrete base for the implementation of a complete middleware that handles applications through an AppStore for Smart Homes and deploys automatically these applications in a real environment, using the available hardware devices, and including mechanisms to ensure management of resource privacy.

3. CONCLUSION AND FUTURE WORK

The MAS proposed in this paper contains four classes of goal-directed agent that reason on the deployment of applications on a hardware infrastructure. The use of MAS made it possible to introduce privacy measures at architecture and organisation levels, on top of which we added a user-defined privacy policy mechanism. This was an important criterion for the choice of the agent paradigm since in the AmI domain there are often different infrastructure owners that need to ensure the privacy of their resources. The separation between the applicative and the infrastructure layers, together with the decentralised approach also enhance the robustness of the solution. The clearly delimited entities, with either virtual (the applications) or physical (users, infrastructure elements) correspondents, guided the agentification. The next steps of this work will be to consider data privacy by defining data privacy policies in order to facilitate the local processing and storage of the data, with the user deciding which kinds of data he authorises to come out of his own infrastructure. This will impact the reasoning on the deployment, since the hardware entities have to be chosen in order to comply with this new data privacy policy.

REFERENCES

[1] L. Atzori, A. Iera, and G. Morabito. The internet of things: A survey. *Computer Networks*, 54(15):2787 – 2805, 2010.

[2] C. Caval, A. El Fallah Seghrouchni, and P. Taillibert. Keeping a clear separation between goals and plans. In F. Dalpiaz, J. Dix, and M. van Riemsdijk, editors, *Engineering Multi-Agent Systems*, volume 8758 of *Lecture Notes in Computer Science*, pages 15–39. Springer International Publishing, 2014.

[3] K. Kravari and N. Bassiliades. A survey of agent platforms. *Journal of Artificial Societies and Social Simulation*, 18(1):11, 2015.

[4] F. Piette, C. Dinont, A. El Fallah Seghrouchni, and P. Taillibert. Deployment and configuration of applications for ambient systems. *Procedia Computer Science*, 52:373 – 380, 2015. The 6th International Conference on Ambient Systems, Networks and Technologies (ANT-2015).