Collaborative Homes

Fano Ramparany[1], Iago F. Trentin[1,2], Julien Cumin[1] and Olivier Boissier[2]

[1]*Orange Innovation, 28 chemin du Vieux Chêne, 38244 Meylan, France*

[2]*Mines Saint-Etienne, Univ Clermont Auvergne, CNRS, UMR 6158 LIMOS, Institut Henri Fayol, F - 42023 Saint-Etienne, France*

Abstract

In this article, we introduce our work on using multi-agent technologies for implementing context aware services, for the benefit of urban communities.

Our claim is that linked data and multi-agent technologies are key assets for enabling advanced context aware services for the benefit of users living in smart environments. To explore and assess our vision, we have developed a demonstrator that we also describe in this article. In our demonstrator, each house of a collaborative neighborhood is monitored by a dedicated agent, which exploits the content of the house context database. Such agents are able to reason on the current context and plan their actions accordingly. Their are also able to align their behaviour to features that are usually ascribed to human beings such as the ability to respect social rules and human values. The context database consists of a personal and private knowledge graph, that models the home context information.

To implement our demonstrator we used the Home'In smart home platform [1] and the JaCaMo multi-agent platform [2] and prototyped a collaborative security service.

Video: http://ramparany.free.fr/atac/collaborativeHomesFinal.mp4
Source Code: http://ramparany.free.fr/atac/code.tar.gz

Keywords

multi-agent systems, linked data, context aware services

1. Introduction

When talking about smart homes, the qualifier of smart denotes that the home is instrumented with connected sensors and actuators. These devices provide real time data about what is going on in the home. By analysing these streams of data we can infer higher level context information including the identity, location and activities of persons present. Thus, agent technology will valorize this context information by providing more awareness to smart home services. More precisely, software agents bring reasoning and collaboration capabilities, which they intimately interwine with their autonomy, pro-activity, reactivity. In addition, they act as intelligent assistants which enable people to make the best use of their home resources.

Potential use cases we envision include resource sharing (e.g. tools and furniture loaning). social interaction fostering (e.g. collaborative safety and security), opportunistic behaviours (e.g. last minute car sharing).

All the Agents Challenge (ATAC 2021)
fano.ramparany@orange.com (F. Ramparany); iagofelipetrentin@gmail.com (I. F. Trentin); julien1.cumin@orange.com (J. Cumin); olivier.boissier@emse.fr (O. Boissier)

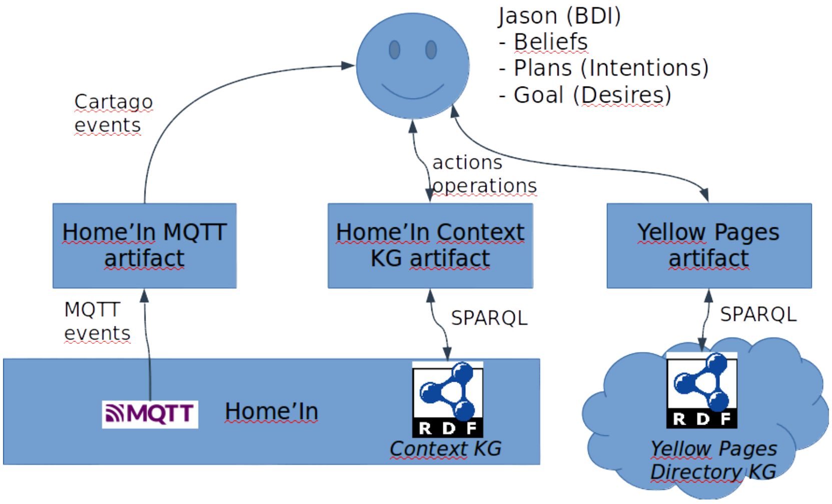

Figure 1: Home agent architecture

2. Implementation

Figure 1 presents the architecture of our home agent. As we will elaborate in section 4, context modeling requires a powerful representation language able to cope with the heterogeneity of context information. Semantic web and linked data technologies are key enablers to handling such semantic heterogeneity. The Home'In platform integrates a Context Knowledge Graph manager.

The Context Knowledge Graph implements the context model of each home as a RDF graph using the Jena [3] library. This RDF graph is based on the following main ontologies: FoaF [4], SOSA [5]. These ontologies are commonly used in the IoT domains. You can refer to [6] for a detailed description of the Context Knowledge Graph manager. As detailed later in section 4, this Knowledge Graph (KG) is updated on the fly, by processing device measurements that are published to the Home'In MQTT bus in realtime.

A Jason Home agent monitors the home activity through a Cartago artifact which focuses on this same MQTT bus, and submits SPARQL queries to the Context KG manager to get an accurate view of the home context through another dedicated Cartago artifact. This agent detects and recognizes specific situations. It has a library of plans which it executes to handle these situations.

Another Jason agent, called the yellow page agent, provides contact information (phone number, address,...) of services and businesses in the city. This information is also stored as a Knowledge Graph.

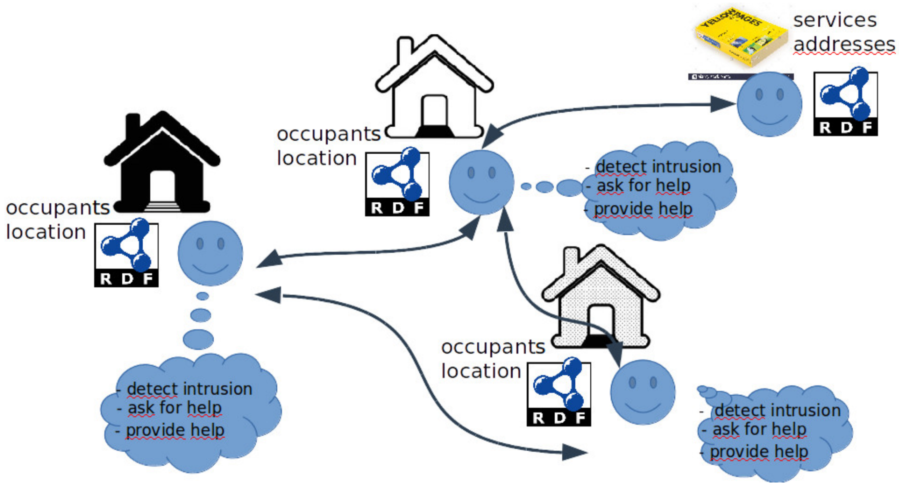

Figure 2: Prototype deployment architecture

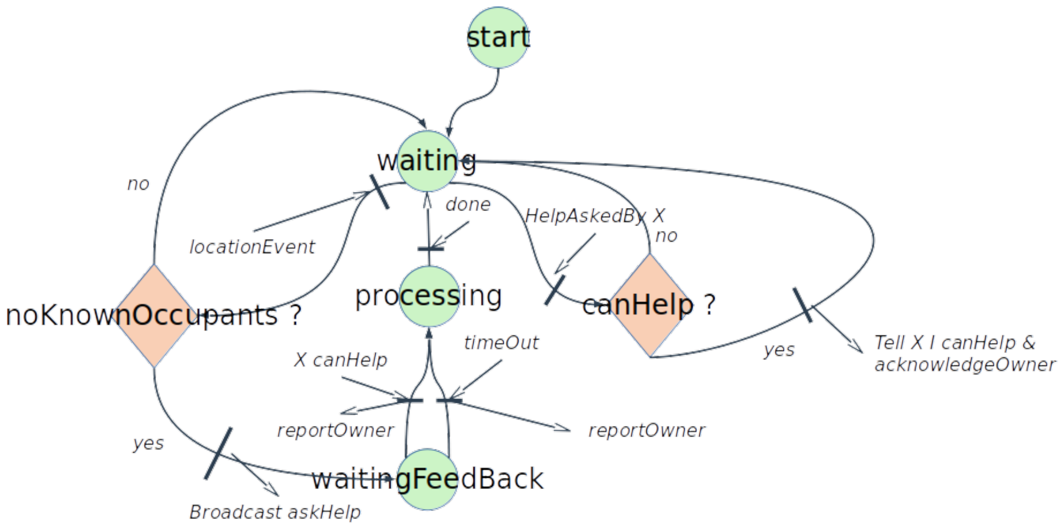

Figure 3: Home agent behaviour

3. Use Case

Figure 2 presents the prototype we have developped. It runs three deliberative home agents which continuously analyze the context KG of their respective homes. Whenever one home agent detects an intrusion, it triggers a plan which requests help from other home agents. If the plan requires the police intervention, the home agent retrieves the police contact information from the yellow page agent.

Figure 3 depicts the behaviour of a home agent. It is presented using a finite state machine formalism. The three home agents share the same behaviour and implementation. This ensures a high level of consistency in the way these agents interact with each other. In this finite state machine diagram, circles define agent states and diamonds define agent actions and decisions. Upon external events that the agent is monitoring and depending on the agent's own decision, the agent state might change. States transitions are depicted as arrows that link initial states to target states. A perpendicular segment in the middle of the arrow indicates that the transition has been partly caused by an external event.

Let us detail each states the home agent run through during its lifetime and the actions and decisions it might make while being in each state:

waiting state: At bootstrap, the home agent is waiting for any external event. In this state there are two events it is sensitive to: A location event and a request for help from another home agent. In case it receives a location event corresponding to a presence in the home, the agent queries the Context Manager whether there are any known occupants in. In case the answer is negative, it concludes that there is an intrusion and it triggers the corresponding plan, otherwise, it stays in the waiting state as there is nothing to worry about. The SPARQL query that the home agent submit to the Context Manager is:

```
PREFIX sosa: <http://www.w3.org/ns/sosa/> PREFIX homein: <http://orange.smart.home/HomeIn#>
ASK WHERE {
 ?o1 sosa:resultTime ?maxt .
 ?o1 sosa:hasFeatureOfInterest ?x .
 ?o1 sosa:hasResult ?place .
 FILTER(?x != homein:unknown)
 FILTER(?place != homein:outside)
 {
   SELECT (MAX(?t) AS ?maxt) ?x WHERE {
     ?o a homein:LocationObservation .
     ?o sosa:hasFeatureOfInterest ?x .
     ?o sosa:resultTime ?t .
   } GROUP BY ?x }}
```

Note that this SPARQL query contains and embbeded SELECT query, which purpose is to retrieve the latest location observation for each person. Once done, it checks if that person is known (i.e. is different from unknown) and in the home (i.e. in a place different from outside). In case an intrusion has been detected, the agent launches the corresponding plan. In Jason the plan is expressed as:

```
+onlyUnknownOccupantDetected : true <-
  ?name(Me);
  .broadcast(achieve,pleaseGetHelpRequestFrom(Me));
  -+state(waitingFeedback);
  !!timer30000.
```

As you can see, by executing this plan, the agent broadcasts a request for help. It then puts itself in a waitingFeedback state, and sets up a timer, so that in absence of any feedback after 30 seconds, it will engage in an emergency backup plan that we don't detail here. While in the waiting state, if the home agent receives a request for help, it checks wether there is somebody in. This is done by querying the Context Manager with a SPARQL query, similar to the previous one. If the response is positive, the home agent informs the occupant present about the situation and asks whether she/he is willing to help out in the safety operations. It then waits for the occupant's reply. If the reply is positive, it lets the breached home agent know which neighbor is ready to help.

waitingFeedback state: In that state, the agent is waiting for help proposals from other home agents. As mentioned it waits for at most 30 seconds. If within these 30 seconds it receives a proposal for help, it informs its occupants about the situation. As soon as both the occupants of the breached house and the occupants that are willing to help out have been identified by their respective home agents, they can establish a standard communication using their phone to organize the home safety operations. The home agent then puts itself in the processing state.

processing state: In this state, the occupants and the helpers are handling the situation. So the home agent asks the home occupant to confirm that it can return to the waiting state, once the situation has been made safe again.

4. Discussion

Let us highlight the main features of our prototype. As mentioned in section 3 **our agents are deliberative and informational**. Our prototype involves **four linked data sources**. Three of them provide context information of their respective home. The fourth one is the yellow page agent which provides a directory service.

As mentioned in section 2, **the content of smart home context information is heterogeous by nature**. It contains information about people, information about equipment, environmental information, etc. We use the KG formalism to manage this heterogeneity. In addition, the content of home context graphs and the directory graph managed by the yellow page agent use two different ontologies.

Our agents have heterogeneous behaviours: whereas home agents are deliberative, the yellow page agent is informational. However, through their interaction, they constitute a inter-mediation layer for coping with the information heterogeneity mentioned earlier, as each agent knows when and how to exploit its linked data source, and how to share it with other agents.

The home is equipped with connected sensors (thermometer, pressure sensors and other environmental sensors) whose measurements are collected and processed to update the context KG in real time. This enables the home agent to perform a timely context evaluation and quick decision. This is essential in our collaborative security use case. In addition to this realtime **dynamicity**, new furnitures, sensors, actuators and equipment, can be easily inserted in the home environment at a longer term temporal granularity, say at a yearly rate. This insertion will be done by simply adding new nodes and relations to the KG which correspond to new instances of the ontologies classes and their relationship with existing nodes. New types of devices can even be introduced by adding nodes corresponding to new classes to our ontology. Agent could even be involved in this process by detecting objects they don't recognize as a type they know.

In our prototype, the three home agents and yellow page agent form **a community that collaborate** to ensure collective safety in the entire neighborhood through a form of **social interaction among agents**. The breached home broadcasts a request for help to the other homes, which will positively answer depending on who is present. Home agents eventually submit critical decisions they make to occupants for final approval before the appropriate actions are performed.

References

[1] H. Rahimi, I. F. Trentin, F. Ramparany, O. Boissier, SMASH: a Semantic-enabled Multi-agent Approach for Self-adaptation of Human-centered IoT, 2021. URL: https://hal-emse.ccsd.cnrs.fr/emse-03251346, submitted to PAAMS 2021.

[2] O. Boissier, R. H. Bordini, J. F. Hübner, A. Ricci, A. Santi, Multi-agent oriented programming with jacamo, Science of Computer Programming 78 (2013) 747–761. URL: https://www.sciencedirect.com/science/article/pii/S016764231100181X. doi:https://doi.org/10.1016/j.scico.2011.10.004, special section: The Programming Languages track at the 26th ACM Symposium on Applied Computing (SAC 2011) & Special section on Agent-oriented Design Methods and Programming Techniques for Distributed Computing in Dynamic and Complex Environments.

[3] A. Seaborne, 2010, Jena, a Semantic Web Framework, URL: http://wiki.apache.org/incubator/JenaProposal.

[4] D. Brickley, L. Miller, FOAF Vocabulary Specification 0.97, Namespace document, FOAF Project, 2010. URL: http://xmlns.com/foaf/spec/20100101.html.

[5] K. Janowicz, A. Haller, S. J. D. Cox, D. L. Phuoc, M. Lefrançois, Sosa: A lightweight ontology for sensors, observations, samples, and actuators., CoRR abs/1805.09979 (2018). URL: http://dblp.uni-trier.de/db/journals/corr/corr1805.html#abs-1805-09979.

[6] F. Ramparany, Q. H. Cao, A semantic approach to iot data aggregation and interpretation applied to home automation, in: 2016 International Conference on Internet of Things and Applications (IOTA), 2016, pp. 23–28. doi:10.1109/IOTA.2016.7562689.