Multi-agent Negotiation of Decentralized Energy Production in Smart Micro-grid

Alba Amato, Beniamino Di Martino, Marco Scialdone, and Salvatore Venticinque

Abstract. SmartGrid is an electricity network that can intelligently integrate the actions of all users connected to it in order to efficiently deliver sustainable, economic and secure electricity supplies. In this context the CoSSMic project aims at fostering a higher rate of self-consumption of decentralized renewable energy production, using innovative autonomic systems for management and control of power micro-grids on users behalf. To achieve this goal we have designed an ICT framework that integrates different appliances such as smart meters, solar panels, batteries, etc., providing a common platform to support sharing of information and negotiation of energy exchanges between power producers and storages in accordance with policies defined by owners, weather forecasts, and habits and plans of participants.

Keywords: Multi-Agent Systems, Smart Grid, Energy Market, XMPP.

1 Introduction

A SmartGrid is an electricity network that employs innovative products and services together with intelligent monitoring, control, communication, and self-healing technologies to better facilitate the connection and operation of generators of all sizes and technologies. It allows consumers to play a part in optimizing the operation of the system. It significantly reduces the environmental impact of the whole electricity supply system. CoSSMic (Collaborating Smart Solar-powered Micro-grids - FP7-SMARTCITIES-2013) is an ICT European project that aims at fostering a higher rate for self-consumption (<50%) of decentralized renewable energy production by

Alba Amato · Beniamino Di Martino · Marco Scialdone · Salvatore Venticinque
Department of Industrial and Information Engineering,
Second University of Naples, Via Roma 29, Aversa, Italy
e-mail: {alba.amato,beniamino.dimartino,
marco.scialdone,salvatore.venticinque}@unina2.it

D. Camacho et al. (eds.), *Intelligent Distributed Computing VIII*,
Studies in Computational Intelligence 570, DOI: 10.1007/978-3-319-10422-5_17

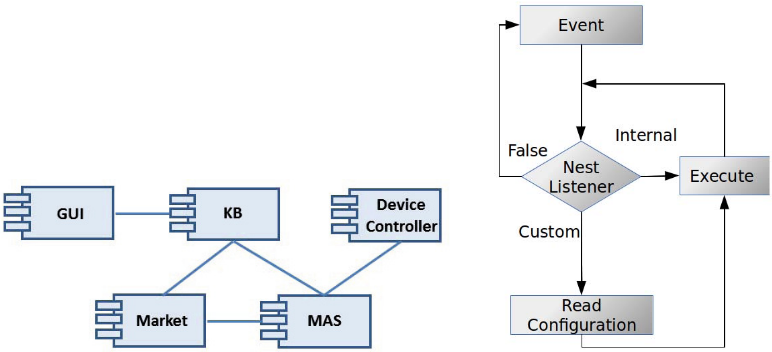
innovative autonomic systems for management and control of power micro-grids on users' behalf. This will allow power optimization of household and neighborhood and of sales to the network. In addition CoSSMic will provide a higher degree of predictability of power deliveries for the large power companies, and it will satisfy the requirements and achieve the benefits discussed above. To obtain a more profitable distribution of energy for utilities and customers it is necessary a system able to trade energy allocation and price information. Besides a 'smart' system has to manage the information's exchange within the network, the integration of renewable energies, the strategy to reduce costs or for optimization of energy consumption and distribution. To obtain these goals we propose a modular, vendor agnostic, agent based architecture that uses the publish/subscribe paradigm. Cloud services will be provided to connect distributed installations, to allow for power monitoring and updating policies by users. The agent based approach was chosen to run each software instance autonomously. The design and development of such a framework should support the communication among agents over a peer-to-peer overlay to negotiate the scheduling of power sources to energy storages. In this paper we address the problem of designing a framework that supports agents based P2P negotiation among power consumers and producers for the optimal scheduling of energy exchange.

2 Related Work

The scientific community investigates different priorities in the field of smart grids. Much effort has been spent on the investigation in this field of agents technology [7]. In [6] authors consider how consumers might relate to future smart energy grids, and how exploiting software agents to help users in engaging with complex energy infrastructures. In [1] authors describe a Message Oriented Middleware (MOM) to simplify distributing applications across heterogeneous operating systems, programming language, computer architectures, networking protocols, and at the same time reducing the complexity on the interconnection functionalities and providing a high level of scalability based on RabbitMQ, Data Distribution Service (DDS) and the Extensible Messaging and Presence Protocol (XMPP). In [5] multi-agent resource allocation in a competitive peer-to-peer environment is addressed making use of micro-payment techniques, along with concepts from random graph theory and game theory. It provides an analytical characterization of protocol and specifies how an agent should choose optimal values for the protocol parameters. In [4] authors claim that agent and peer-to-peer based decentralized self-management can change the future of energy markets in which the power grid plays a core role. Our contribution, and in particular the CoSSMic project, supports negotiation among end users on real power grid.The framework will be validated on real infrastructures by trials that involve inhabitants of two different European countries. Both software and hardware will be integrated and customized to be compliant with existing installations. Finally we have experience in building network of agents both in smart cities applications [2] and for negotiation and brokering of computational resources in Cloud markets [3].

3 Architecture

The stakeholders of CoSSMic Framework include Users, Devices and Power Suppliers (GenCO). The CoSSMic User interacts with the framework by a Graphical User Interface (GUI) according to three high level use cases (UCs). The *Management* UC allows to configure and manage the available devices and to manage and control at a higher level, through rules and policies, the energy flows. The *Monitoring* UC provides facilities to supervise and to get useful information for eventually reconfiguring the devices and scheduling the allocation of power. *Reporting and statistics* integrate information from several sources, including power companies, weather reporting and forecasting and to encourage the growth of the neighborhood network. CoSSMic Devices will use the Platform providing metering and management services. CoSSMic will exchange power with GenCO when the MicroGrid cannot satisfy its requirements in the case of over- or under- production of energy within the CoSSMic neighborhoods. The CoSSMic platform will run on embedded computer systems that will be provided to end users as a black box, to be plugged into the power network and connected to Internet. The Platform will be installed in every household and will join a community of other instances within the neighborhood. Instances of the platform communicate by a P2P overlay and with the Cloud to eventually exploit advanced services. Each platform instance will communicate with other households only for the energy negotiation. Components of the CoSSMic Platform are shown in Figure 1 (a).



(a) CoSSMic Platform (b) Control Agent Life Cycle

Fig. 1 Control Agent Life Cycle and Contract Net Interaction Protocol

A Graphical User Interface (GUI) allows users to interact with electronic devices through graphical icons and visual indicators. The Knowledge Base (KB) is an information repository that provides means for information to be collected, organized, shared, searched and utilized. The Multi Agent System (MAS) allows for the deployment of agents of consumers and producers that will participate in the energy distribution. The Market supports the energy negotiation. Agents can publish calls for proposals, accept offers and negotiate with other agents. The Device Controller allows agents to send real time commands to electric devices in the smart house. Agents will act on user behalf and their main task is to negotiate energy. They can be classified according to two categories. *Consumers* buy energy for passive devices. *Producers* can sell energy. In this category there are, for example, power generators, solar panels, wind turbines, but in CoSSMic we only consider solar panels. Those devices, which are able both to produce and consume energy will be defined *Prosumers*. In this category there will be also storages, which are represented by a couple of agents belonging to the two different classes.

The consumer agents, once found one who offers enough power to meet their own needs, will kick off the negotiation. If an agent cannot find enough energy to satisfy its needs in the neighborhood market, it will contact a GenCo.

4 Implementation

To develop and deploy agents, we used the JADE multi-agent system[1]. Figure 2 shows the implementation of our agents-based architecture. The GUI, described previously, allows the user to interface with the system. The various devices in the home can send information about electricity consumption through wireless interfaces (for example UHF or Zigbee) to the mediator. Mobile devices (e.g. electric cars), instead, send information through the CoSSMic Cloud. In both cases the information, through the Mediator, reach the agent platform whose main actors are: *User Agent*: an agent associated to the user that interfaces with the GUI and with the DB Manager; *Event Bus*: handles the various possible events. Control Agents subscribe to this event bus in order to receive events from devices; *DB Manager*: interfaces with the Knowledge Base that, as we said, can collect, manage and share information; *Control Agent*: the most important of all, it manages the electric energy of all devices in the household. In particular, in our MAS we have two types of control agent. A *Consumer Agent* is associated to each electric device that absorbs electricity. Its task is to obtain the energy required from the device to operate. A *Producer Agent* is associated to each electric device that produces electricity and it tries to sell this energy to consumer agents. In Figure 1 (b) the control agent life cycle is shown. When an event occurs, the CA calls all the registered listeners. Some event and listeners are built in, but the developer is able to define new custom events and custom listeners.

A configuration file is used to define those events, and related listeners, which the agent must react to. In our prototype we already provide the Negotiation Handler as a listener that can be used to start a new negotiation on a specific event. In the configuration itself the developer can choose what type of negotiation protocol to be used and the related negotiation strategy. We already support a negotiation protocol based on the FIPA Contract Net Interaction Protocol. In our test, the initiator starts the negotiation when the listener receives an event indicating that the

---

1 `http://jade.tilab.com/`

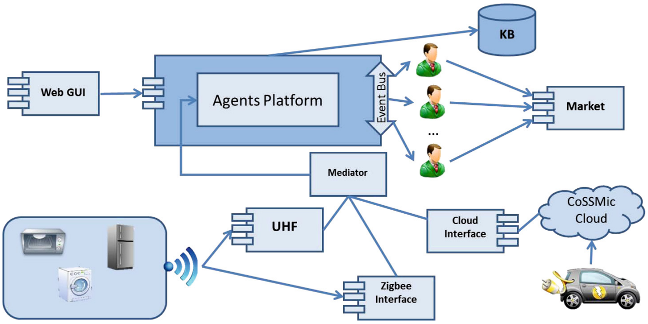


Fig. 2 Implementation of CoSSMic Platform

energy level is below a certain threshold and the initiator sends a CFP to a number of power producers who are already running. At this point there are two possible situations. The responders send a message with the energy price, the initiator accepts one proposal and rejects all the others, or the initiator rejects all the proposals or more simply there are not proposals. In the last case the agent changes its own behaviour from initiator to responder, waiting for call for proposal by new producers. A straightforward negotiation strategy could be the acceptance of the proposal that offers the lower price among the ones which are below a thresholds. Of course different strategies can be defined and configured by the developer.

At transport layer to support P2P negotiation protocols we use the eXtensible Messaging and Presence Protocol (XMPP)[2]. Our solution involves the use of a chat room that acts as a market for a neighborhood where agents enter and publish their requests for sale or purchasing energy. The prototype is implemented using TIGASE, an open source project to develop a XMPP server implementation in Java. Furthermore, to realize the communication between agents through XMPP, we use Smack[3], an Open Source XMPP (Jabber) client library for instant messaging and presence. Using an XMPP server, it is possible to create rooms - open or reserved - where users can meet and communicate. In the implementation of the prototype, we use a persistent Chat Room (in this way, if everyone leaves the chat or if the server restarts, the room remains and stores all messages sent) as Energy Market, in which the control agents can enter autonomously. Once entered the market room, producer agents can make proposals. If a consumer agent believes that an offer is satisfactory, it starts a negotiation with the agent who made the proposal.

[2] http://www.xmpp.org

[3] http://www.igniterealtime.org/projects/smack/

5 Conclusion

The paper presents a modular software architecture for supporting agents to collect information about local energy production and storage resources of neighborhoods of individual houses and to schedule the energy flows using negotiation protocols. Besides the implementation of a preliminary prototype is described. The introduction of the publish/subscribe paradigm as a solution for service data exchange within a smart grid was introduced together with the usage of the XMPP protocol to allow secure interaction with other systems allowing the extension of the architecture to other platforms with little effort. Experimental activities are an ongoing work together with the implementation of optimal negotiation strategies and new protocols. We also plan, as a future work, to use a NoSQL database for metering and monitoring all smart micro grids participating in the trials of the project.

Acknowledgements. This work has been supported by CoSSMic (Collaborating Smart Solar-powered Micro-grids - FP7-SMARTCITIES-2013).

References

1. Alkhawaja, R.A., Ferreira, L.L., Albano, M., Garibay-Martínez, R.: Qos-enabled middleware for smart grids. Tech. rep., Polytechnic Institute of Porto, ISEP-IPP (2012)
2. Amato, A., Di Martino, B., Venticinque, S.: Semantically augmented exploitation of pervasive environments by intelligent agents. In: Proceedings of the 2012 10th IEEE International Symposium on Parallel and Distributed Processing with Applications, ISPA 2012, pp. 807–814 (2012)
3. Amato, A., Liccardo, L., Rak, M., Venticinque, S.: Sla negotiation and brokering for sky computing. In: Proceedings of the 2nd International Conference on Cloud Computing and Services Science, CLOSER 2012, pp. 611–620 (2012)
4. Brazier, F., Ogston, E., Warnier, M.: The future of energy markets and the challenge of decentralized self-management. In: Beneventano, D., Despotovic, Z., Guerra, F., Joseph, S., Moro, G., de Pinninck, A.P. (eds.) AP2PC 2008/2009. LNCS (LNAI), vol. 6573, pp. 95–103. Springer, Heidelberg (2012)
5. Peleg, Y., Rosenschein, J.S.: Agents and peer-to-peer computing: Towards P2P-based resource allocation in competitive environments. In: Beneventano, D., Despotovic, Z., Guerra, F., Joseph, S., Moro, G., de Pinninck, A.P. (eds.) AP2PC 2008/2009. LNCS (LNAI), vol. 6573, pp. 129–140. Springer, Heidelberg (2012)
6. Rodden, T.A., Fischer, J.E., Pantidi, N., Bachour, K., Moran, S.: At home with agents: Exploring attitudes towards future smart energy infrastructures. In: Proceedings of the SIGCHI Conference on Human Factors in Computing Systems, CHI 2013, pp. 1173–1182. ACM, New York (2013)
7. Rogers, A., Ramchurn, S.D., Jennings, N.R.: Delivering the smart grid: Challenges for autonomous agents and multi-agent systems research. In: AAAI (2012)