A Smart Grid and the Vehicle-to-Grid/Grid-to-Vehicle Problem

Subjects: Computer Science, Software Engineering

Contributor: Nikolaos I. Spanoudakis , Charilaos Akasiadis , Georgios Iatrakis , Georgios Chalkiadakis

The smart grid constitutes an important emerging application domain for artificial intelligence and multiagent systems (MAS). In the smart grid, energy and information both flow over electricity distribution and transmission networks in all possible directions. As such, buildings, as well as electric vehicles (EVs), become active energy consumers and/or producers, and the need for their effective integration into the system arises. Not only is the smart grid an electricity network with diverse consumers and producers, it is also a dynamic marketplace where heterogeneous devices appear and need to connect and interoperate.

open multiagent systems | smart grid | engineering multiagent systems (EMASs) | internet of things

1. Smart Grids and the Vehicle-to-Grid/Grid-to-Vehicle Problem

As electric vehicles (EVs) further penetrate energy markets globally, electricity demand patterns are subject to change at levels that might become disruptive to the stability and the reliability of the current electricity grids [1]. A way to mitigate this risk is by introducing “smart charging”, or grid-to-vehicle (G2V) capabilities, according to which the charging of EVs can be delayed and can take place at later time intervals than immediately after connecting to a charger [2], seeking, e.g., those with more renewable production, with less demand from other EVs, or with better pricing. In the opposite direction to that of G2V, vehicle-to-Grid (V2G) approaches can benefit from the capability of EV batteries to store energy, and thus coordinate their discharging to support situations of energy supply shortage [3].

Since the smart grid consists of multiple individual and economically minded entities, it is natural to model it as a multiagent systems (MAS) [4]. MASs provide a number of benefits in contrast to their centralized counterparts, such as faster computation times and scalability, since processing is performed in a decentralized fashion, and private data are not required to be shared.To date, many simulation tools and prototypes have been proposed that put forward V2G capabilities. Such approaches may either analyze low-level technical details regarding the operation of EVs, or are integrated into environments that include the respective individual stakeholders. Researchers now proceed to briefly review several representative such systems.

The study presented in [5] introduced EVLibSim, a Java-based simulator of the operation of charging stations. This tool offers a user interface (UI) that can be used to manage charging stations. Its capabilities include the creation, modification, and monitoring of charging stations given the application of particular scheduling algorithms. In addition to being used by domain experts to test potential scenarios of use, it focuses on charging stations only, without incorporating different types of stakeholders.

The work presented in [6] involved describing a MAS that supports the decision-making of EV drivers in regard to locating charging stations and charging opportunities in the city of Valencia. This system incorporates multimodal information from various sources, such as traffic monitoring systems, social networks, and pricing, in order to optimize the placement of charging stations. Such approaches are valuable during the design of charging infrastructure, but do not help in deciding what will happen next, after the infrastructure is deployed and becomes operational.

The survey presented in [7] involved a large-scale literature analysis of the MAS-based control of smart grids, providing information regarding the related technologies and standards, and the application of intelligent agents commercial projects. The authors in [8] proposed coalition formation techniques for EVs, providing services related to V2G and demand-side management. An MAS architecture was designed, and the implementation of simulations was performed, using the Java Agent Development Framework (JADE [9] 1). In that study, different kinds of intelligent agents were considered, i.e., EVs, aggregators that form EV coalitions, and a transmission system operator that acts as a mediator and regulator. Coalitions were formed with the objective of reaching minimum energy requirements for participating in the regulation market. However, this approach did not allow for more sophisticated selection processes, making it difficult to scale, and the presented evaluation involved only five EVs.

Another approach based on the use of JADE for the coordination of EV battery charging is that of [10]. In that study, individual EV driver preferences were taken into account, such as their willingness for V2G participation and the vehicle's charging availability. It was shown via experiments that the proposed MAS managed to satisfy EV owners' charging preferences individually, even in emergency conditions.

2. Frameworks and Internet of Things-Based Real-World Trials

In recent years, great progress has been achieved with respect to the delivery of real-world trials that offer V2G/G2V and which might incorporate simulators as well. To begin, XBOS-V [11] is a software-based open platform that can be utilized for controlling the charging of EVs connected to small buildings. The implementation of the standardized communication method for V2G, ISO 151182, provides the connection specifications for chargers and EVs. Relevant approaches are the open charge point interface (OCPI), the open charge point protocol (OCPP), and the open smart charging protocol (OSCP) [12]. OpenV2G [13] provides the required modules to implement the V2G public key infrastructure, e.g., to guarantee security for the EV and charging station connections, and also to allow for simulations to be executed. Another approach, the grid-integrated electric mobility model (GEM) [14], simulates both electricity and mobility aspects. However, this approach only allows for analysis to be conducted at a higher level, without referring to charging station recommenders, for example, and other particular stakeholders. Another tool that can be used to manage the charging of batteries is ACN-Sim [15], which can be utilized by individual end-users but not by large-scale grid operators.

Regarding the IoT domain, SYNAISTHISI is a research-oriented platform composed of open-source frameworks that can host dockerized services and can also act as a translator between various application-layer protocols [16]. Service dockerization allows scalable deployments to occur independently of the underlying operating systems of the hosts. Furthermore, the platform’s support for multiple protocols enables the orchestration of heterogeneous agents and services that may follow different implementation paths. Furthermore, the platform offers user authentication and authorization and restricts access to private and sensitive channels for the exchange of information, and also supports semantic annotations of exchanged information and available services.

3. Engineering Multiagent System for the Vehicle-to-Grid/Grid-to-Vehicle Application Domain

The fields of agent-oriented software engineering and multiagent systems engineering have produced a wealth of abstractions, methods, and techniques for developing MASs. A survey of this field is outside of the scope of this entry. A possible method is ASEME, the Agent Systems Engineering MEthodology [17]. ASEME can be naturally used in the design and modeling of IoT-based MAS systems, as well as in ambient intelligence applications [18][19][20][21]. It builds on statecharts and, more broadly, the unified modeling language (UML [22]) in order to perform system analysis and design models. It is agent-architecture- and agent-mental-model-independent, allowing the designer to select the architecture type and the mental attributes of the agent, thus supporting heterogeneous agent architectures. Moreover, ASEME puts forward a modular agent design approach and uses the so-called intra-agent and inter-agent control concepts. The former is implemented to coordinate the different modules that implement the agent’s capabilities, thus determining its behavior, whereas the latter allows for the control of the society of agents by defining the protocols that govern its coordination.

Importantly, in agent communication, there typically exist predefined message sequences that can be applied in several situations that share the same communication pattern regardless of the application domain [9]. Such message sequences are defined as protocols.

ASEME uses two relatively common abstractions for modeling agents: capability and functionality. Busetta et al. [23] view capability as “a cluster of plans, beliefs, events and scoping rules over them”. Braubach et al. [24] extended this idea and proposed that capabilities can contain sub-capabilities and have at most one parent capability. They defined the agent concept as an extension of the capability concept, aggregating capabilities. In the Prometheus methodology [25], each functionality identified in the analysis phase ends up being mapped to a capability in the design phase. In the agent modeling language (AML) [26], capability is a concept used to model an abstraction of a behavior in terms of its inputs, outputs, pre-conditions, and post-conditions. The behavior is the software component, and its capabilities are the signatures of the methods that the behavior realizes, accompanied by the method's pre-conditions and post-conditions. This approach is similar to that of service-oriented architectures, and thus considers the agent as an aggregation of services.

In ASEME, the agent coordinates its capabilities in the intra-agent control model. Capabilities are themselves decomposed to simple activities. For instance, one capability of a personal assistant agent is the ability to locate an appropriate charging station for its user's car. This task can be decomposed to specific activities, e.g., one activity is finding out which charging stations are in the user's vicinity, which charging sockets are available at each of them, and their free slots. Another activity is ranking the available slots according to the user's preferences. After identifying the activities associated with a capability, the next step is to connect them to a specific functionality, i.e., a specific method, algorithm, technology, or technique. This is an important managerial task as each activity can be connected to different functionalities. For example, the decision-making activity of ranking the available charging slots may be connected to an argumentation theory, implying an argumentation-based method, or to a utility function, implying a multi-criteria decision analysis method.

The inter-agent control model defines the capability of an agent to participate as a role in a specific protocol. ASEME allows the seamless integration of the inter-agent control model in the intra-agent control model as they follow the same formalism—i.e., statecharts [27]. The statecharts formalism does not exhibit the limitations (limited scalability, explosion of states) posed by other formalisms such as Petri-nets [28]. Therefore, in the research, researchers used the statechart formalism to define researchers' open protocols and design patterns. Finally, ASEME automatically generates portions of the agent code or provides guidelines for the programmers to transform their design models into implementation models.

References

1. International Energy Agency. Global EV Outlook: Towards Cross-Modal Electrification; IEA: Paris, France, 2018.
2. Danner, D.; Duschl, W.; de Meer, H. Fair charging service allocation for electric vehicles in the power distribution grid. In Proceedings of the e-Energy '19, Phoenix, AZ, USA, 25–28 June 2019; pp. 406–408.
3. Sarkar, R.; Saha, P.K.; Mondal, S.; Mondal, A. Intelligent Scheduling of V2G, V2V, G2V Operations in a Smart Microgrid. In Proceedings of the e-Energy '20, Virtual Event, 22–26 June 2020; pp. 417–418.
4. Karfopoulos, E.L.; Hatziargyriou, N.D. A Multi-Agent System for Controlled Charging of a Large Population of Electric Vehicles. IEEE Trans. Power Syst. 2013, 28, 1196–1204.
5. Rigas, E.S.; Karapostolakis, S.; Bassiliades, N.; Ramchurn, S.D. EVLibSim: A tool for the simulation of electric vehicles' charging stations using the EVLib library. Simul. Model. Pract. Theory 2018, 87, 99–119.

6. Jordán, J.; Palanca, J.; del Val, E.; Julian, V.; Botti, V. MASEV: A MAS for the Analysis of Electric Vehicle Charging Stations Location. In The PAAMS Collection, Proceedings of the Advances in Practical Applications of Agents, Multi-Agent Systems, and Complexity, Toledo, Spain, 20–22 June 2018; Demazeau, Y., An, B., Bajo, J., Fernández-Caballero, A., Eds.; Springer International Publishing: Cham, Switzerland, 2018; pp. 326–330.
7. Mahela, O.P.; Khosravy, M.; Gupta, N.; Khan, B.; Alhelou, H.H.; Mahla, R.; Patel, N.; Siano, P. Comprehensive overview of multi-agent systems for controlling smart grids. CSEE J. Power Energy Syst. 2020, 8, 115–131.
8. Kamboj, S.; Kempton, W.; Decker, K.S. Deploying power grid-integrated electric vehicles as a multi-agent system. In Proceedings of the 10th International Conference on Autonomous Agents and Multiagent Systems (AAMAS 2011), Taipei, Taiwan, 2–6 May 2011; pp. 13–20.
9. Bellifemine, F.L.; Caire, G.; Greenwood, D. Developing Multi-Agent Systems with JADE; Wiley Series in Agent Technology; John Wiley and Sons Ltd.: Hoboken, NJ, USA, 2007.
10. Papadopoulos, P.; Jenkins, N.; Cipcigan, L.M.; Grau, I.; Zabala, E. Coordination of the Charging of Electric Vehicles Using a Multi-Agent System. IEEE Trans. Smart Grid 2013, 4, 1802–1809.
11. Lipman, T.; Callaway, D.; Peffer, T.; von Meier Alexandra, E.A. Open-Source, Open-Architecture Software Platform for Plug-In Electric Vehicle Smart Charging in California; California Energy Commission: Sacramento, CA, USA, 2020.
12. Van Aubel, P.; Poll, E. Security of EV-charging protocols. arXiv 2022, arXiv:2202.04631.
13. Käbisch, S.; Peintner, D.; Heuer, J.; Schmutzler, J.; Gröning, S.; Lauterbach, M. The OpenV2G Project. Available online: https://openv2g.sourceforge.net (accessed on 18 March 2023).
14. Sheppard, C.; Jenn, A. Grid-Integrated Electric Mobility Model (GEM) v1.0; US Department of Energy: Washington, DC, USA, 2021.
15. Lee, Z.; Johansson, D.; Low, S.H. ACN-Sim: An Open-Source Simulator for Data-Driven Electric Vehicle Charging Research. In Proceedings of the e-Energy ’19, Phoenix, AZ, USA, 25–28 June 2019; ACM: New York, NY, USA, 2019; pp. 411–412.
16. Akasiadis, C.; Pitsilis, V.; Spyropoulos, C.D. A Multi-Protocol IoT Platform Based on Open-Source Frameworks. Sensors 2019, 19, 4217.
17. Spanoudakis, N.I.; Moraitis, P. The ASEME Methodology. Int. J.-Agent-Oriented Softw. Eng. 2022, 7, 79–107.
18. Spanoudakis, N.; Moraitis, P. Engineering Ambient Intelligence Systems Using Agent Technology. IEEE Intell. Syst. 2015, 30, 60–67.
19. Zambonelli, F. Key Abstractions for IoT-Oriented Software Engineering. IEEE Softw. 2017, 34, 38–45.

20. Savaglio, C.; Ganzha, M.; Paprzycki, M.; Bădică, C.; Ivanović, M.; Fortino, G. Agent-based Internet of Things: State-of-the-art and research challenges. Future Gener. Comput. Syst. 2020, 102, 1038–1053.
21. Fortino, G.; Savaglio, C.; Spezzano, G.; Zhou, M. Internet of things as system of systems: A review of methodologies, frameworks, platforms, and tools. IEEE Trans. Syst. Man Cybern. Syst. 2020, 51, 223–236.
22. Unified Modeling Language, Superstructure, V2.1.2; Technical Report formal/07-11-02; Object Management Group: Needham, MA, USA, 2007.
23. Busetta, P.; Howden, N.; Rönnquist, R.; Hodgson, A. Structuring BDI agents in functional clusters. In Proceedings of the International Workshop on Agent Theories, Architectures, and Languages, Orlando, FL, USA, 15–17 July 1999; Springer: Berlin/Heidelberg, Germany, 1999; pp. 277–289.
24. Braubach, L.; Pokahr, A.; Lamersdorf, W. Extending the capability concept for flexible BDI agent modularization. In Proceedings of the International Workshop on Programming Multi-Agent Systems, Utrecht, The Netherlands, 26 July 2005; Springer: Berlin/Heidelberg, Geimany; pp. 139–155.
25. Padgham, L.; Winikoff, M. Developing Intelligent Agent Systems: A Practical Guide; Wiley Series in Agent Technology; Wiley: Hoboken, NJ, USA, 2004; p. 240.
26. Trencansky, I.; Cervenka, R. Agent Modeling Language (AML): A comprehensive approach to modeling MAS. Informatica 2005, 29, 391–400.
27. Harel, D.; Naamad, A. The STATEMATE Semantics of Statecharts. ACM Trans. Softw. Eng. Methodol. 1996, 5, 293.
28. Mazouzi, H.; Seghrouchni, A.E.F.; Haddad, S. Open Protocol Design for Complex Interactions in Multi-agent Systems. In Proceedings of the First International Joint Conference on Autonomous Agents and Multiagent Systems: Part 2, AAMAS ’02, Bologna, Italy, 15–19 July 2002; ACM: New York, NY, USA, 2002; pp. 517–526.