Spatial services for decentralised smart green energy management

Houssem Ben Mahfoudh*†, Giovanna Di Marzo Serugendo*, Nabil Abdennadher†,
Andreas Rumsch‡, and Andres Upegui†
*University of Geneva
Email: {Houssem.Benmahfoudh, Giovanna.Dimarzo}@unige.ch
‡Lucerne University of Applied Science and Arts
Email: Andreas.Rumsch@ihomelab.ch
†University of Applied Science of Western Switzerland, Geneva
Email: {Nabil.abdennadher, Andres.Upegui}@hesge.ch

***Abstract*—In neighbourhoods, the number of energy generators are growing. A main reason for this being the rise in people's energy needs and the possibility of local energy production. Future scenarios involve people's choices of energy type (mostly green) as well as sharing energy among neighbours. This article investigates the use of spatial services applied to a decentralised green energy management system. Spatial services are a new type of bio-inspired services spreading geographically across dispersed devices. They are agent-based and suitable for dynamically changing scenarios. We prototyped, implemented and deployed an energy management scenario involving a peer to peer innovative energy management system and a self-adaptive system. Preliminary results show the feasibility of the envisaged scenarios. Future work include investigating advanced peer to peer scenarios as well as enhancing current spatial services with reinforcement learning.**

I. Introduction

Nowadays cities equip neighbourhoods, city-centers and buildings with new energy generation sources and networking capabilities for making cities *smarter*. In order to face the high costs and the growing needs for energy, cities and citizens aim at installing renewable and sustainable systems. One of the efforts towards this aim, from the citizen perspective, is to install photo-voltaic systems (PV) that will provide electricity during the day. The main limitation of this source of energy is that, unlike hydroelectric power sources, its availability is not on-demand and therefore it is only available during daylight. Generally, the average household does not consume all the energy generated at peak hours, and in such a case two alternatives are usually available for the excess energy: it can be stored in batteries for consumption during the night, or it can be fed back through the power supply lines in order to contribute to the overall power generation.

The price of energy for households is typically based on two fixed price tariffs, one during the daytime, and one during the nighttime. For the energy supplier the price is based on the market price charged by energy suppliers of higher levels in the electricity market. This leads to a situation in which one has to pay a fixed price for consumed energy but gets a significantly lower price for excess energy fed into the grid; a price that is more or less bound to the market price. An approach to enable the system to trade energy in a more flexible way is to allow the households to decide where they want to buy their energy from. One solution is to design a meshed communication network that connects owners or even devices together. It acts as an option to sell energy directly among neighbours who are willing to pay a lower price than that of an established utility supplier. Thus, local producers benefit to get a higher price than they would get from the supplier. Selling the excess green energy, locally produced, directly to another consumer is advantageous for both parties: the energy price for the producer and consumer becomes higher and lower respectively, when compared to that of the utility supplier.

Both options for buying green energy from the utility or from a private producer do not necessarily result in direct consumption of the green energy produced. Instead, this strongly depends on the energy management of each part. Therefore, the approach should additionally be extended with an automated control of the appliances. To accommodate the highly changing fluctuations of energy consumption or production, we propose an agent-based autonomous system that adapts to different conditions, needs and preferences.

First, this paper lists related works followed by a description of our energy management system and energy model. Then, it introduces the underlying coordination model and spatial services. Finally, we present a prototyping and a deployment of our agent-based scenario, and move on to future works and conclusion.

II. Related works

Distributed energy and renewable energy sources are growing over time developing life quality and productivity of citizens. It offers many opportunities for the economic sector while creating a cleaner and more sustainable environment. The pre-existing initiative 'Change38' demonstrates the first steps in that direction [1]. In Change38, producers and consumers are connected via an application. A consumer can then decide to start an appliance. He is then informed about the amount of energy he has consumed from his producer. The producer gets paid extra for the energy he can sell to his connected customer. Peer to peer energy management system [2] is a hot topic in

978-1-5386-3669-5/18/$31.00 2018 IEEE

the research field consolidated by the solar panel expansion and the high energy storage capacities. Other works focus on designing a peer to peer trading network built on blockchain [3], [5], [11]. This aims to exchange energy between neighbours by negotiating price and quantity. A new decentralised energy currency NRG-coin was also revealed. It is used for billing and payment between neighbours [10].
Some agent-based solutions for smart energy management focus on negotiating the price of buying and selling power between two traders [13]. Agent-based simulations are used to validate the effect of dynamic pricing proposed by electricity power companies over a long period of time, and manage community-based shared storage batteries [9].
Our proposal provides an adaptive solution employing services, enacted by agents. These services spread over a geographic area, dynamically adapt to changing conditions, autonomously and continuously identifying providers and consumers of energy, availability and type of energy.

III. Energy Management

Current energy distribution infrastructures rely on a hierarchical architecture. The lowest level 7 is the energy distribution from a central transformer to the households: the local distribution network. The transformer acts as an energy supplier for the local distribution network (see Figure 1). Transformers are typically designed to transport energy in one direction but are able to work bidirectionally. Due to the increasing number of photo-voltaic (PV) systems, the amount of energy transported from the local distribution network to higher levels of the grid also increases. Therefore, the aim should be to consume energy where it is produced in the same local distribution network. But the centralised architecture imposes limitations to an energy management system among neighbours and does not provide an overall system autonomy. For an optimal solution we have to respect the grid structure

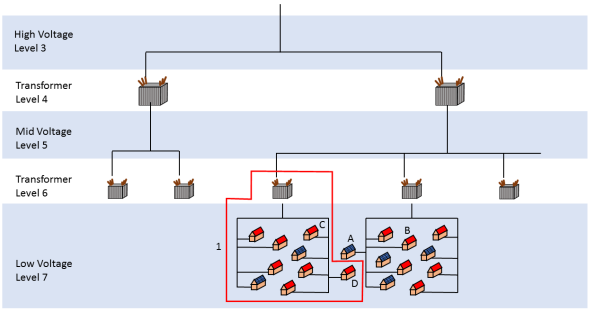

Fig. 1: Energy should be distributed between households connected inside one grid cell.

(see Figure 1). So the energy flows from the highest to lowest voltage level but also in the opposite direction if energy is produced in the lower levels. Households are on level 7, the lowest voltage level, and connected to a transformer station. All households connected to the same transformer are in the same local distribution network. In the case of a local distribution network containing houses with PV systems, these systems provide energy that other houses in the same network (cell) can use. For example house A produces energy that house B consumes (see Figure 1). Excess energy has to flow back via the transformer station (which is on level 6) to the superior level 5 (the medium voltage level). Inside level 5 the green energy can be distributed to other local distribution networks on level 7, for example from house A to house C, with an extra load of the transformers. Again, excess energy flows to a superior level. At the end, the produced energy has to be consumed at the moment of production.

IV. Energy Model

In an energy management system, green energy suppliers (*resp.* consuming appliances) have to be able to communicate their amount of excess green energy (*resp.* their need for energy). From the consuming side, appliances are switched on by the users at random points in time, as well as by the system itself for optimisation purposes. Furthermore, the needed energy is not constant during the process but changes depending on the concrete task, like heating water or drying clothes. From the producing side, fluctuating weather conditions or shadowing by other buildings result in a volatile energy production. All this results in a highly dynamic and spatially distributed system where the amount of produced, *resp.* consumed, energy can change rapidly for each location. Figure 2 shows the state of production and consumption at two different times and two different weather conditions. Knowing the location of producers and consumers is essential as both should be in the same local distribution network. Communication between the different households is necessary in order to know the consumers needs and the producers excesses. The next subsections describe the actors and the agents of our system, as well as the payload carried during communications.

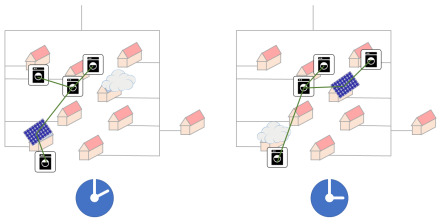

Fig. 2: The solar energy management system behaves like a spatial system

A. Actors

Our system consists of the following actors:
Utility company: the connecting infrastructure (levels 7 to level 3) belongs to a utility company that provides the transfer of energy from a house to another at a given price.
Energy producer: house owners play the role of energy producers, if their houses are equipped with an energy producing system (e.g. PV).
Energy consumer: house owners or tenants play the role of

energy consumers.

For the sake of simplicity, in this paper, we do not consider the price of transferring energy from one house to the other, we only consider that the closer the producer, the cheaper the price. We also consider scenarios where a given house owner can be both an energy producer (provided the house can produce energy) or an energy consumer. We also consider that one house hosts only one owner or one tenant.

B. Agents

Software agents [14], [15] are autonomous software entities that have sensing and acting capabilities, collaborate and interact with each other. Software agents work on behalf of energy producers and energy consumers. In each household, a computational node hosts one or more such agent that has the following capabilities:

Requesting / Proposing energy: communicating with neighbouring agents for requesting needed or proposing available energy;

Negotiating prices and contract: identifying the best offer for buying or selling energy based on current needs and communicating the choice to the corresponding agent;

Controlling appliances: optimising energy within a given household by monitoring and switching on/off the appliances.

C. Payload

In practice, agents will be able to share information about energy by specifying:

Price : Produced energy has a price which can be fixed by the owner. In this case, agents seeking for energy can first check the costs and compare them with other energy providers' prices. We can imagine it as a requests-offers local market.

Availability : Agents specify for how long they would like to request or provide energy.

Quantity : Agents need to specify how much energy they need or can provide. In some cases, fetching energy from many providers is a solution if no one can provide the total requested amount of energy.

Type of Energy : In a general scenario, energy from various sources and from different types (solar, wind, fossil, water, nuclear, etc) can be provided and requested.

Section : To avoid power cable overloads and infrastructure costs, we aim primarily at transfering energy within the same cell.

These five points will be reduced under a payload of the form: <P,A,Q,T,S>.

V. Coordination model

To coordinate the communication, negotiation and control activities of the agents, we use a coordination model. Each household hosts a computational node that contains a coordination platform (also called coordination middleware) implementing the coordination model we discuss here. Our work relies on the SAPERE model [7], a coordination model for multi-agent pervasive systems inspired by chemical reactions (see Figure 3). In chemical-based coordination models, coordination rules work as chemical reactions on the data tuples present in the coordination media (called shared tuple space). Coordination rules dynamically aggregate, bind, suppress, spread data present in the shared tuple space. Agents communicate with each other, or coordinate their own activities by injecting/retriving data tuples from their tuple space. The coordination platform then takes care of aggregating, spreading or evaporating the tuples.

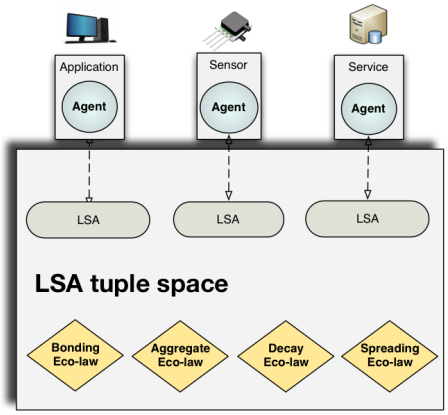

FIG. 3: SAPERE Coordination Model

The SAPERE coordination model is based on four main concepts:

1) *Software Agents*: active software entities representing the interface between the tuple space and the external world including any sort of device (e.g. sensors), service and application. They are the agents of our energy model. There may be one or more agents linked to each coordination platform.
2) *Live Semantic Annotations (LSA)*: LSAs are tuples of data used to store applications data. Values can change with time or be updated regularly. LSAs convey the payload of our energy model.
3) *Tuple space*: data space containing all the tuples in a node. Tuples of a node are shared among the agents of that node. There is one shared space for each computational node (in each house).
4) *Eco-laws*: chemical-based coordination rules, namely: Bonding (for linking an agent with a data that he referred to, was waiting for, concerns it, etc.); Aggregation (for combining two or more LSAs value, such as keeping maximum, minimum values, averaging values, filtering values, etc.); Decay or evaporation (regularly decreasing the pertinence of data and ultimately removing outdated data); Spreading (for propagating LSAs to neighbouring nodes).

Software agents are sensitive to LSAs being injected in the shared tuple space. We say they "bond" to these LSAs. The tuple values trigger some agent's behaviour, which then starts some computation. The result of this computation can be diverse and multiple: the agent instructs some actuator to provide some effect in the environment (e.g. turn off the heater); the agent

may inject a new data tuple in the tuple space (e.g. the quantity of energy requested); or update an LSA value (e.g. providing an updated value of the generated quantity of energy, or the monitoring value of an appliance). Coordination of the different agents occurs through this indirect retrieval and injection of data in the shared tuple space. Such kinds of models are efficient in dynamic open systems: agents communicate asynchronously without having a global knowledge of the system; new agents seamlessly join or leave the system at any moment.

A. Spatial services

Spatial services [12] are a new generation of services exploiting spatially distributed data (LSAs). They result from collective and decentralised interactions among multiple computing entities (agents). These entities coordinate with each other based on their local positions. In a peer to peer network, agents are able to communicate with other similar agents present in neighbouring spots. Agents' behaviour will be determined by bio-inspired mechanisms [4] [6] that are space- and time-related. These agents actually enact what we call spatial services. They are space-related since they spread across a series of geographically dispersed devices (across space); a set of correlated piece of data, which taken collectively have a specific meaning. They are also time-dependent, since deposited data can be subject to dynamic evaporation. Data present at a given location can aggregate and produce enriched information. Agents residing in computational nodes are sensitive to this information and may respond accordingly. Spatial services use self-organisation paradigms in order to ensure robustness and scalability. It also allows services to communicate in an autonomous way and react to environmental perturbations.

B. Spatial services for energy management

Computational nodes are attached to individual houses. Each node runs an instance of the SAPERE coordination platform. Agents inject, update LSAs or retrieve information from the tuple space. In our case, LSAs contains the payload. Agents also establish contracts among the involved parties and notify the transformer to transfer energy from the producer to the consumer. Different bio-inspired mechanisms provided as services [6] support the agents:

- Spreading : It diffuses information within a neighbourhood. It is a broadcast with a fixed propagation hops.
- Gradient : is based on the spreading pattern. It provides additional information about hops distance. Also, it aggregates information by applying some algebraic operation.
- Evaporation : is a pattern employed to mark information relevance. User's offers and requests become outdated and risk to overload the communication network. Adding a kind of timer to data ensures a trusted level of communication.
- Chemotaxis : based on the gradient index, it routes LSAs back to the source of the gradient.

The advantages of the spatial service approach are: (1) no need for a central node resulting in lower costs for a utility; (2) faster response and therefore more precise synchronisation between consumer and producer; (3) autonomous adaptation to changes such as arrival/departure of energy providers, or energy consumers, changes in energy types, fluctuations of available or requested energy levels; (4) increased system robustness thanks to its adaptation to infrastructure faults and changes; (5) adapted for mobile entities like electric cars.

VI. Case Study

Figure 4 depicts the case where six houses of a given cell/sector are connected via the utility company network for the energy and a peer to peer communication network for agents' communication. In this scenario, the peer to peer communication network has the same structure as the energy network. Every house hosts a node equipped with the coordination platform, which spreads, receives or routes requests. None of the agents in the nodes has a global view of

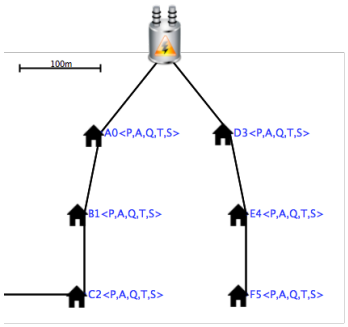

Fig. 4: A Neighborhood sector

the network. Agents can only sense their neighbours through information received through the communication network. In this section, we present a scenario where agents communicate to exchange energy. We define four types of LSAs:

- Settings LSA (S): <S, Node name, Payload>
- Request LSA (R): <R, Node name, Payload>
- Proposal LSA (P): <P, Node name, Payload>
- Contract LSA (C): <C, Node name, Payload>

Table I shows the local settings of the payload of the different nodes (i.e. available energy locally produced). A0 decides to spread a Request LSA asking for green energy. It requests 1 kWh of energy for 1 hour and for a price less than 0.09. In this example, A0 is directly connected to B1 and D3. According to Table I, these nodes do not satisfy A0 needs (the price proposed by B1 is too high and D3 does not provide energy). Spreading continues until the request reaches C2 and F5. These two nodes have enough energy and offer a cheaper price. Both nodes will send a Proposal LSA back to A0 and wait for its confirmation. A0 will select F5 as it provide the cheapest price. This generates a Contract LSA to accept the proposal. To ensure a continuous supply of energy, agents running in every node are ready to make new requests and modify contracts when problems occur (changing weather, electricity problems, etc). This makes the system self-adaptive to dynamic changes in the environment, such as changes in energy demand, in the production, or seamless to integration of new producers or consumers entering/leaving the system.

Node	P(currency unit)	A (hour)	Q(kWh)	T	S
A0	0.1	0	0	Solar	1
B1	0.12	2	1	Hydraulic	1
C2	0.08	4	2	wind	1
D3	-	-	-	-	1
E4	0.16	0.5	5	Solar	1
F5	0.06	3	1	wind	1

TABLE I: Local settings - available energy produced

VII. AGENT-BASED DESIGN

We explain here the different services and agents involved in our system. We consider three different types of agents, all present in all nodes.

LocalSettingsAgent: updates the local production values <P,A,Q,T,S> within a node. For example, for node C2 of Figure 4, this agent updates the payload values (as shown in the third line of Table I). To do so, it gathers sensor energy data, the type of energy provided by C2 (i.e. wind), monitors energy levels and regularly injects the updated tuple into the local SAPERE platform (Algorithm 1). This tuple represents the payload of C2 and corresponds to its local settings (S) <S, C2, 0.08, 4, 2, wind, 1>. In the case of A0 (see Table I), the payload mentions that A0 provides solar energy, but at this point in time, there is no such energy available to share with neighbours. For D3, its payload mentions that it is not equipped with any energy generator, and does not provide any available energy.

```
Result: LocalSettingsAgent
initialization;
while execution do
    update(payload);
end
```

Algorithm 1: LocalSettingsAgent

RequestAndContractEnergyAgent: This agent monitors the current need for energy in each household. As soon as the level of energy is below a certain threshold, it spreads a request for energy to the neighbouring houses. This request takes the form of a tuple that specifies the requested energy (Algorithm 2). For instance, this agent will inject in the SAPERE platform a Request LSA (R) with values corresponding to <R, A0, 0.09, 1, 1, wind, 1>. This request then spreads as a gradient through the communication network. Later, upon a potential response from neighbours (Algorithm 2), this agent decides upon the best strategy for actually contracting the requested energy to be consumed. A0 spreads a Contract LSA (C) to F5 <C, F5, 0.06, 1, 1, wind, 1> as it satisfies its needs in energy (see Figure 5).

ProposalEnergyAgent: This agent detects arriving requests for energy, compares it to the current local settings, and if a match is possible submits a proposal for providing energy (Algorithm 3). For instance, node C2, detects the Request LSA incoming from node A0 (see above), matches it with its current settings and injects a Proposal LSA (P) of the form <P, C2, 0.08, 1, 1, wind, 1> mentioning its settings. F5 will do the same with <P, F5, 0.06, 1, 1, wind, 1>. Each node that has sent a proposal will allocate the requested energy for a period of time. These two proposals will reach A0 using chemotaxis. Later upon a potential contract from a consumer, the agent chosen by A0 (i.e. here F5) actually transfers the energy to the neighbour and updates its local settings accordingly by updating the payload tuple, i.e. by injecting an updated Settings LSA (S) in its local node.

```
Result: RequestAndContractEnergyAgent
initialization;
while execution do
    if energyQuantity <energyThreshold then
        sendRequest();
    end
    if receiveProposal() then
        compareAndSelect(proposals);
        sendContracts();
    end
end
```

Algorithm 2: RequestAndContractEnergyAgent

```
Result: ProposalEnergyAgent
initialization;
while execution do
    if detectRequest() and satisfyRequest() then
        sendProposal();
        allocateEnergy(time);
    end
    if receiveContract() then
        sendEnergy();
        update(payload);
    end
end
```

Algorithm 3: ProposalEnergyAgent

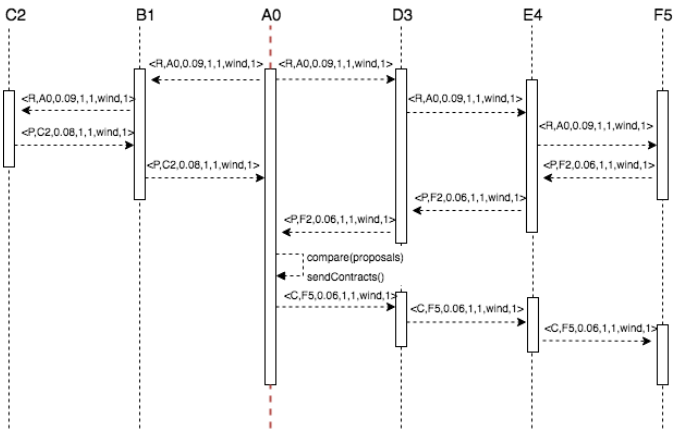

FIG. 5: Sequence diagram to establish contract

Spatial services are enacted by the different agents above, and rely on the gradient and chemotaxis spatial services to further propagate and respond to various requests. Spatial services are provided on request as energy delivery depends on the actors of the moment, the current energy supply, requests, etc.

VIII. Prototyping, implementation and deployment

We propotyped the scenario above with The ONE-SAPERE [8]. This is a prototyping tool that simulates nodes movements, and that runs in each node an actual instance of the SAPERE platform. Every node has several agents running on it. It allows to prototype and validate applications with realistic and large-scale scenarios and actual codes that will be later deployed in real settings. Figure 6 shows an image of the prototyping tool representing the five nodes (one for each house) of our case study. Request LSAs from A0 follow the communication network. A gradient service conveys these LSAs across the network. F5 and C2 Proposal LSAs flow back to A0 following the gradient. The chemotaxis service actually conveys these proposals. It is then similar for the Contract LSA sent to F5 by A0.

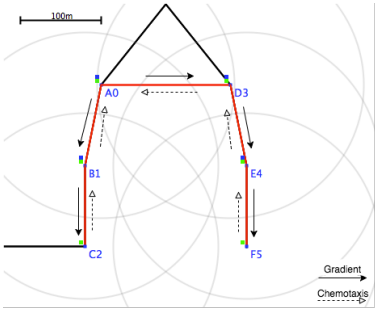

Fig. 6: Prototyping scenario

Finally, we implemented a peer to peer network (Figure 7 and video https://youtu.be/xvI_cK-qF6Y) hosting our system. It is composed of six smart nodes (raspberry pi each running the SAPERE platform) controlling each an OpenZwave bulb (for showing the level of energy consumed in each house), and a light sensor Gen 6 (acting as PV system). The agent working on behalf of the house on the right side of the figure plays the role of consumer and requests energy from its neighbours when its own PV is not able to furnish enough energy. Request LSAs reach the other 4 agents, that answer back with proposals. Since energy is less expensive from the producer on the left side of the figure (1CHF), the requester chooses this proposal. The house on the left side shares its energy with the one on the right side and thus reduces its own consumption (shown by a less bright bulb). The scenario further enfolds with the house on the left itself being in need of additional energy.

IX. Future work and Conclusion

We defined a decentralised green energy management system based on spatial services and network communication among nodes. In a large network, we don't need to spread requests until the last node. This introduces a new level of complexity, which is distance, that we intend to tackle with the gradient. We need also to consider the potential overloading of electric cables and energy transfer cost incurred. Reinforcement learning may be a good solution for adapting spreading parameters; like number of hops that we should use to get a response and avoid networks flooding. Spatial services applied to distributed energy systems provide a new vision for designing collaborative and adaptive energy management scenarios.

Fig. 7: Implementation

References

[1] In Situ Energy AG. Change38. https://www.change38.ch.

[2] Chao Long Meng Cheng Chenghua Zhang, Jianzhong Wu. Review of existing peer-to-peer energy trading projects. *ICAE2016*, 2017.

[3] Stamatis Karnouskos Martin Griesemer Dejan Ilic, Per Goncalves Da Silva. An energy market for trading electricity in smart grid neighbourhoods. *IEEE*, 2012.

[4] Giovanna Di Marzo Serugendo, Jose Luis Fernandez-Marquez, and Francesco Luca De Angelis. Engineering spatial services: Concepts, architecture, and execution models. In R. Ramanathan and K. Raja, editors, *Handbook of Research on Architectural Trends in Service-Driven Computing*, pages 136–159. IGI Global, 2014.

[5] Carolin Beer David Dauer Christof Weinhardt Esther Mengelkamp, Benedikt Notheisen. A blockchain-based smart grid: towards sustainable local energy markets, 2017.

[6] Jose Luis Fernandez-Marquez, Giovanna Di Marzo Serugendo, Sara Montagna, Mirko Viroli, and Josep Lluis Arcos. Description and composition of bio-inspired design patterns: a complete overview. *Natural Computing*, pages 1–25, 2012.

[7] Bernhard Anzengruber Gabriella Castelli Francesco L. De Angelis Giovanna Di Marzo Serugendo Simon Dobson Jose Luis Fernandez-Marquez Alois Ferscha Marco Mamei Stefano Mariani Ambra Molesini Sara Montagna Jussi Nieminen Danilo Pianini Matteo Risoldi Alberto Rosi Graeme Stevenson Mirko Viroli Juan Ye Franco Zambonelli, Andrea Omicini. Developing pervasive multi-agent systems with nature-inspired coordination. *Pervasive and Mobile Computing*, 17, Part B:236 – 252, 2015. 10 years of Pervasive Computing' In Honor of Chatschik Bisdikian.

[8] Giovanna Di Marzo Serugendo Graeme Stevenson Gabriella Castelli Jose Luis Fernandez-Marquez, Francesco L. De Angelis. The one-sapere simulator: A prototyping tool for engineering self-organisation in pervasive environments. *IEEE*, 2016.

[9] Rio Kanamori, Takuya Yoshimura, Shogo Kawaguchi, and Takayuki Ito. Evaluation of community-based electric power market with agent-based simulation. In *2013 IEEE/WIC/ACM International Conferences on Web Intelligence (WI) and Intelligent Agent Technology (IAT)*, pages 108–113, 2013.

[10] Kristof Van Moffaert Narcs Avellana Ann Nowe Mihail Mihaylov, Sergio Jurado. Nrg-x-change: a novel mechanism for trading of renewable energy in smart grids. *IEEE*, 2014.

[11] Davor Svetinovic Nurzhan Zhumabekuly Aitzhan. Security and privacy in decentralized energy trading through multi-signatures, blockchain and anonymous messaging streams. *IEEE*, 2016.

[12] Giovanna Di Marzo Serugendo, Nabil Abdennadher, Houssem Ben Mahfoudh, Francesco L. De Angelis, and Roberto Tomaylla. Spatial edge services. *Global IoT Summit*, 2017.

[13] Zhu Wang and Lingfeng Wang. Adaptive negotiation agent for facilitating bi-directional energy trading between smart building and utility grid. *IEEE TRANSACTIONS ON SMART GRID*, 4(2):702–710, June 2013.

[14] G. Weiss. *Multiagent systems. A modern approach to distributed artificial intelligence*. The MIT Press, 1999.

[15] M. Wooldridge. *An introduction to multi-agent systems*. John Wiley & Sons, 2002.