A Distributed Agent-based System For Coordinating Smart Solar-powered Microgrids

Shanshan Jiang
SINTEF
P.O. Box 4760 Sluppen
7465 Trondheim, Norway
Email: Shanshan.Jiang@sintef.no

Salvatore Venticinque
Dep. of Industrial and Information Engineering
Second University of Naples
Aversa, Italy
Email: salvatore.venticinque@gmail.com

Geir Horn
University of Oslo
P.O. Box 1080 Blindern
0316 Oslo, Norway
Email: Geir.Horn@mn.uio.no

Svein Hallsteinsen
SINTEF
P.O. Box 4760 Sluppen
7465 Trondheim, Norway
Email: Svein.Hallsteinsen@sintef.no

Matthias Noebels
ISC Konstanz
Rudolf-Diesel-Str. 15
D-78467 Konstanz, Germany
Email: matthias.noebels@isc-konstanz.de

***Abstract*—Renewable energy like solar power is crucial for the transition to more sustainable energy supply and use in the modern society. Buildings with rooftop solar panels form microgrids acting as prosumers and are usually not under the control of the regulated companies operating the public grids. Currently much work has focused on the self-consumption of individual microgrids. On the contrary, the CoSSMic system targets at a neighborhood of microgrids with the primary goal to maximize the self-consumption of the whole neighborhood by coordinating their energy use and storage. To address the challenge of the fluctuating and partly unpredictable nature of renewable energy, a novel hybrid control mechanism is proposed, where planning and scheduling based on predictions is supplemented by a reactive feedback loop to compensate the inability to predict accurately the rapid fluctuations in PV output due to passing clouds. To enable easy creation, evolution, and operation of the neighborhoods without the need for expensive central equipment and support, a Peer-to-Peer, multi agent, and negotiation based architecture has been designed and implemented to realize the control mechanism. Early evaluation has been based on user centered design and involvement, while final evaluation will be carried out using experiments and simulations based on one-year trials on two trial neighborhoods in Germany and Italy respectively.**

***Keywords*—Collaborating microgrids; Peer-to-peer; Multi-agent system; Coordination; Prediction**

I. Introduction

Smart energy systems and smart buildings play an important role in the vision of smart cities and the integration of renewable energy sources, like solar and wind, is crucial for the transition to more sustainable energy supply and use in the modern society. Solar energy is well suited to local deployment, creating so-called *microgrids* embedding rooftop photovoltaic (PV) panels, a number of power consuming devices and storage. They are typically confined in a smart home or office building, and although connected to the public grid, possibly via a smart meter, the microgrid is normally outside the control of the regulated companies operating the public grid.

The self-consumption and self-sufficiency of such local solar powered microgrids are hampered by the fluctuating nature of the solar power production caused by fluctuating radiation. Thus, it is often a time mismatch between generated and consumed power, because PV panels generate energy during the day, while people tend to use the energy during the morning and in the evening after work. Load shifting and local storage are key mechanisms to cope with this. Storage is expensive, but with the increasing popularity of electric vehicles (EV), the batteries of parked or charging electric vehicles can be used as additional storage.

Previous research, experimenting both with manual and automatic control of load shifting and use of storage, has shown that this can improve self-consumption considerably [1], [2], [3]. So far the focus has mostly been on improving the self-consumption of individual microgrids [1], [2], [3]. In contrast, the CoSSMic (Collaborating Smart Solar-powered Microgrids) project focuses on the coordination of the load shifting and use of storage resources in a group of neighboring microgrids, leveraging differences in energy needs and energy usage patterns between different kinds of buildings. For example schools and kindergartens typically have their energy consumption peaks and valleys at different times than homes. Such groups of power sharing microgrids will be referred to as *neighborhoods* in this paper.

The CoSSMic project aims to develop and experiment with an innovative autonomic ICT based system, controlling the energy usage, storage, and the exchange with the public grid in neighborhoods of collaborating microgrids. The goal is to maximize the self-consumption of the neighborhood. The CoSSMic system will be governed by preferences and constraints set by the building users through modern devices such as smartphones and touchpads. Furthermore, the system will leverage pricing signals and other demand side guidance provided by the electric power retailers and public grid operators. Storage can be provided by dedicated batteries, or by battery powered units connected temporarily for charging, e.g. electric cars. Weather forecasts are leveraged to predict both the output from the local PV panels and the energy consumption of the buildings, and thus enable near optimal *demand side management* [4].

This paper presents and discusses the control architecture of the CoSSMic system. It has largely been shaped by two central challenges: Firstly, in order to enable easy creation, evolution, and operation of the neighborhoods, a robust solution was sought where neighborhoods can be created without the need for any expensive central equipment and support, and where individual microgrids can dynamically join or leave a neighborhood, and thus being resilient to temporary failures without disrupting the overall operation of the neighborhood. A decentralized and distributed coordination of neighborhood is suitable as the microgrids are outside the control of the regulated authorities. Actually decentralized control instead of centralized control is a trend in the current smart grids development [5]. Multi-agent systems (MAS) with Peer-to-Peer (P2P) communication is a promising technology to handle this aspect [6]. Secondly, to cope with the inherent limited predictability of PV generation, the ambition of CoSSMic was to align the power consumption of the neighborhood both to the highly predictable fluctuations in solar radiation caused by the motion of the earth, and to the much less predictable variations in radiation caused by clouds, in particular the rapid fluctuations caused by fast moving clouds.

The novelty of the approach described in this paper is a combination of

- A hybrid control mechanism to cope with the unpredictability of the fluctuating insolation, and
- A P2P, negotiation-based, multi-agent system implementation of the control mechanism for coordination of the self-consumption of neighborhoods.

The paper is structured as follows: Section II presents some related work; Section III describes the idea of the hybrid control mechanism; Section IV presents the technical architecture, where Section IV-A presents the architecture for a microgrid of a household; and Section IV-B presents the coordination architecture for the communication and negotiation of the network of microgrids. Section V describes the prototype implementation and Section VI describes the evaluation. Finally, Section VII concludes and identifies further relevant work.

II. Related work

Incentives to promote self-consumption of PV generation has become increasingly common. Castillo-Cagigal *et al.* modeled and experimented with combinations of storage and load shifting to increase self-consumption using a control system that was tested in a prototype of a self-sufficient solar powered house [1], [2]. Van der Kam and van Sark investigated the potential of combining smart grid technology with electric vehicles (EV) as storage for increasing self-consumption in the residential sector [3]: A model of a microgrid was developed and three smart grid control algorithms evaluated with simulation using one year data of PV-power and electricity demand from the Netherlands. The result showed that the control algorithms could significantly increase self-consumption and reduce peaks in electricity demand from the main public grid.

MAS technologies are designed for applications with distributed, dynamic, scalable and modular characteristics [7], and have therefore been applied to implement various applications in Smart Grid [6]. In particular, they have been widely studied for implementing distributed energy resources (DER) and microgrid energy management systems (EMS) [7], [8]. Kantamneni *et al.* provide a survey of MASs for microgrid control, where they review MAS concepts and architectures, develop platforms and processes, and provide example applications in power systems, mainly for market operations, microgrid protection and service restoration [9].

Several researchers have implemented a MAS to control the operation of microgrid that comprises PV generators, batteries and controllable loads [10], [11], [12], [13]. Dimeas and Hatziargyriou proposed a distributed control approach for microgids based on a MAS with a hierarchical architecture where three control levels are distinguished and implemented using MAS [11], [12]: Local Controllers (LC) for micro sources or loads locally; Distribution Network Operator (DNO) and Market Operator (MO) at the level of the Medium Voltage; and Microgrid Central Controller (MGCC) as the main interface between the public grid (DNO and MO) and the microgrid. The agents in the MGCC collaborate to maximize the gain of selling energy to the public grid, and compete to reduce their individual operating cost. In a centralized implementation, MGCC is responsible for the optimization of the microgird operation, while in a decentralized approach, MGCC only coordinates the LCs that assume the main responsibility for optimization. The microgrids participate in the market and interact with MO through the MGCC. This architecture has been adopted in several other approaches, such as [7], which describes the general architecture of a microgrid EMS based on MAS technologies and a secondary control system based on this architecture. Dimeas and Hatziargyriou further expanded their architecture with multi layered learning used to increase the intelligence and the efficiency of the microgrid [14].

Such an hierarchical and centralized architecture is rooted in the traditional top-down control. The CoSSMic architecture is a bottom-up approach and has no central entity for coordination like MGCC. Moreover, most of the above mentioned work focuses on the integration of microgrids to the public grid, and deals with the market-based control and the control from the public grid. The focus is therefore on the economic gains of individual units and the optimization is based on price. In contrast, CoSSMic considers the self-consumption of the whole neighborhood and thus optimizes based on the amount of PV energy used. Hence, the goal is to minimize the amount of electricity the neighborhood takes from the public grid. Secondary to this is the problem of allocating the economic benefit to the members of the neighborhood rewarding their individual contributions to this saving [15].

Current electricity distribution system treats home and office buildings as consisting of isolated and passive individual energy consuming units and do not include active collaboration from such buildings. In contrast, CoSSMic considers the negotiations and collaboration between buildings where energy can be intelligently managed. Similarly, the SmartHouse/SmartGrid project deals with the interactions of smart houses and smart grids, specially in the areas of energy efficiency and sustainability. Kok *et al.* first proposed a smart grid ICT architecture based on smart houses interacting with smart grids [16], and later added a "PowerMatcher", *i.e.* a MAS solution for coordination in the emerging sustainable electricity system with clusters of DERs [17].

Most of the microgrid control systems do not consider the automatic control of devices and the involvement of end users. The CoSSMic system targets automatic device control in the buildings governed by user preferences where users can plan the tasks and set preferences. In this aspect, the vision is similar to Building Automation Systems (BAS) and smart house technologies [18]. Yu *et al.* addresses the deployment and reconfiguration of the BAS [19]: The system can adapt the configuration to consider multiple users' preferences, based on the user model that incorporates preferences, rules, and user feedback to provide the final global automation policy. A main difference from the above work is that CoSSMic addresses the coordination of a *neighborhood* of microgrids and buildings instead of a single building.

III. A HYBRID CONTROL MECHANISM

There are two possible optimization mechanisms which can be used in a microgrid: Reactive and predictive. A reactive system always decides on the basis of current production and demand and can thus react instantaneously to fluctuations, whereas a predictive system knows the expected production and demand and can schedule within a prediction horizon. A reactive system will be sub-optimal if the duration of a demand exceeds the duration of a production, *e.g.* a long running washing machine programme is launched during a small cloud gap. The quality of the prediction-based mechanism depends on the accuracy of the predictions for PV production, device consumption, and prices. Currently energy tariffs are quite predictable as they are statically defined in the contract. However, this may change in the future when dynamic price models are available. For the other types of prediction it is important to remember that a predictive system will be sub-optimal if the prediction is inaccurate.

In CoSSMic, a hybrid approach is adopted. The prediction-based optimization uses *profiles* to represent the knowledge of energy production and consumption in the planning horizon. A profile describes the predicted evolution of a given parameter over a given time period. Profiles are represented as time series, i.e. series of time value pairs, and are used for various purposes in the system. Each device has one or more profiles describing its energy consumption, production, or charging characteristics, depending on the kind of device. There are also production profiles for PV, weather profiles for predicted weather data, and price profiles for predicted price signals. The profiles can be parameterized to reflect influencing factors, such as ambient temperature and inlet water temperature for a heating or cooling device, and other system parameters like the inclination and orientation for a PV installation.

Statistical learning is used to provide more accurate profiles as a new profile measurement becomes available once a particular mode of a device has completed a run [20]: For instance, a washing machine has several modes, one for each washing programme it has. For each mode there is a corresponding load profile, *i.e.* a time series showing the cumulative energy consumption over the duration of the programme. However, the energy consumption of the washing machine running a particular programme depends on the temperature of the inlet water and the amount of clothes filled into the machine. Measuring the energy consumption of each run using a smart plug allows us to learn statistically, over time, the expected variation in the energy consumption of this mode on this particular device. Hence, this approach provides increasingly better load profiles to use with the predictive load scheduling as more and more runs are made.

The accuracy of the predictions of PV energy production depends on the quality of the weather forecast data used to calculate the output of solar panels, yet it will always be subject to rapid and random variations. Figure 1 shows the PV production over time for a clear day and a cloudy day. A profile for predicted PV production will typically be computed based on the parameters of the PV installation and the weather forecast data a few hours ahead. It typically represents a curve within the envelope for the long term trend of the PV production, with the shape similar to the one for the clear day as depicted in Figure 1. Such prediction is the basis for predictive load scheduling.

However, it is difficult to predict the rapid variations as shown for the cloudy day. To cope with the less predictability of the rapid fluctuation caused by clouds, a reactive feedback loop is used as a corrective mechanism in addition to the long-term prediction based on profiles to ensure better result.

The profile for a background consumption in a household is similar to the PV production profile.

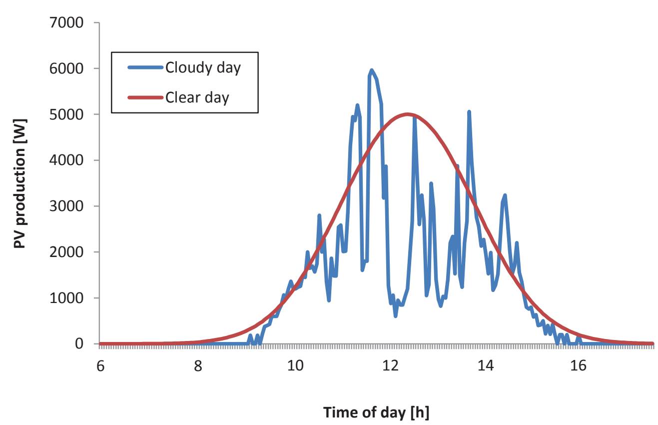

Fig. 1: Example PV production for a clear day and a cloudy day.

The system needs to maintain the list of energy consuming *tasks* defined by the user and not yet completed. Associated with each task are one or more *loads* representing the estimated energy consumption of planned tasks. The optimization part of the system works with loads, and decides the exact scheduling of the loads. Predictive load scheduling for a single PV panel means matching the time continuous profiles of the loads ready to run with the predicted time continuous production profile of the PV panel resulting in an assigned start time for each load. This *time continuous scheduling* is a novel branch of scheduling for which the first results are now becoming available [21].

The unpredicted variation in the production means to add an adjustment using short-term feedback, hence the approach constitutes a hybrid control mechanism. The short-term rapid variations can be absorbed using batteries or using intermittent loads, *e.g.* thermal storages like air conditioners and water heaters. Typically, the profile of an intermittent load consists of alternating on- and off-periods. The shape of these periods can be changed according to offer and demand, *e.g.* by adjusting the starting time of an on-perid, the maximum energy consumed, and the duration of the on-period. Most devices that can be used as intermittent loads have built-in controllers reacting to the deviation from the devices' setpoints, *e.g.* a fridge will extend its on-cycle if its door is left open. Ideally, the system can directly interact with the built-in controllers to obtain status information from the device, set configuration parameters or setpoints, and send control commands. Appliances offering these capabilities are usually sold as "smartgrid ready", but are still rare, and use proprietary protocols. The CoSSMic system therefore uses smart plugs to simply switch on and off the devices, and, if available, a sensor monitoring its status, *e.g.* the temperature.

IV. P2P-BASED TECHNICAL ARCHITECTURE

An ICT system realizing the hybrid control mechanism has been implemented as a MAS based on a distributed P2P model. The multi-agent programming paradigm is well suited to solve complex problems exploiting distributed computing resources and is thus chosen to realize the distributed load scheduling for optimizing energy utilization in the neighborhood. The P2P technology is leveraged to enable communication within the neighborhood without a central coordinator and to allow microgrids to dynamically join and leave a neighborhood. The design uses collaborating autonomous agents, which carry out simpler tasks, and allows for reducing both the performance requirements at each computing node and the complexity of formulation. The global solution will be provided by the emergent behavior of the multi-agent system, whose individuals communicate over the P2P overlay. As agents will address their own sub-problems, exploiting limited knowledge, the global solution will suffer the problem related to local optima. This is the trade-off accepted in CoSSMic to achieve scalability, avoidance of single point of failures, and utilization of limited computing resources.

Two architectures are defined for the system: Firstly, the basic unit in the system is the MAS-based microgrid with distributed intelligence. Section IV-A presents this microgrid architecture. Secondly, a coordination architecture based on a P2P overlay is defined for communication and negotiation to optimize the energy usage in the neighborhood, where each microgrid can dynamically join the network of microgrids. Section IV-B presents this coordination architecture.

A. Microgrid Architecture of A Household

The overall architecture for a microgrid is depicted in Figure 2.

The *Prediction* component computes forecasts for the PV production and the prices for electric power exchange with the public grid, based on third party services and knowledge about the house and its PV installations. For example, yr.no in Norway and DWD service in Germany provides weather services that give forecasts about cloud coverage, air temperature, moisture and wind, which can be used to predict irradiation on the solar panels and the heat exchange of the house with the environment. Furthermore pricing services can be used to utilize prices and price forecasts made available by the retailers and possibly also by the Distributed System Operators (DSOs) if grid use is priced separately. The pricing services may also include green mix of the public grid power and forecasts for that.

The *Task manager* serves as the master agent of the multi-agent system negotiating the load scheduling and has several responsibilities:

- It creates producer and consumer agents to represent the producer and consumer devices of the household and take care of the negotiation.
- It manages the list of planned tasks for the household, and provides the Task planning interface used by the *Graphical User Interface* (GUI) to allow users to create, plan and replan tasks with associated preferences and constraints and also to define more general policies. The task planning involves computing the load profile for the tasks based on the profile of the device assigned to execute it, and in the case of heating or cooling tasks, it also needs to take into account the prediction for heat exchange of the house with the environment. It abstracts the tasks into more or less shiftable loads, and hands them over to the appropriate consumer agents (see the description below).
- It receives prediction updates from the Prediction component and forwards them to the producer agents.

The responsibility of the *Load controller* is to execute the loads according to their schedule such that the total consumption and production of the household stays as close as possible to the schedule. It may interrupt and resume interruptible tasks, for example heating/cooling tasks and EV battery charging tasks. Thus it enables the reactive control loop driven by the short-term variations in PV production and a background demand.

The *Scheduler* implements distributed load scheduling and is responsible for negotiating load scheduling in the whole neighborhood. It consists of Consumer and Producer agents. The *Consumer agents* negotiate power delivery agreements with the *Producer agents* of the whole neighborhood and scheduling of the loads with the other households, and returns assigned start times for the loads. The goal of the negotiation is to maximize the self-consumption of locally produced PV power in the neighborhood. (Re)negotiation is triggered whenever it receives a new load, a change to an already received load, or a prediction change. There is one consumer agent per consuming device and one producer agent per PV system. A battery is associated with both one consumer agent and one producer agent, while an EV is associated with only one consumer agent due to the current situation that discharging of EV as energy source has been restricted in many countries. It is also assumed that the public grid is always available and can provide infinite energy as needed. Therefore, the public grid is represented with a producer agent with infinite capacities

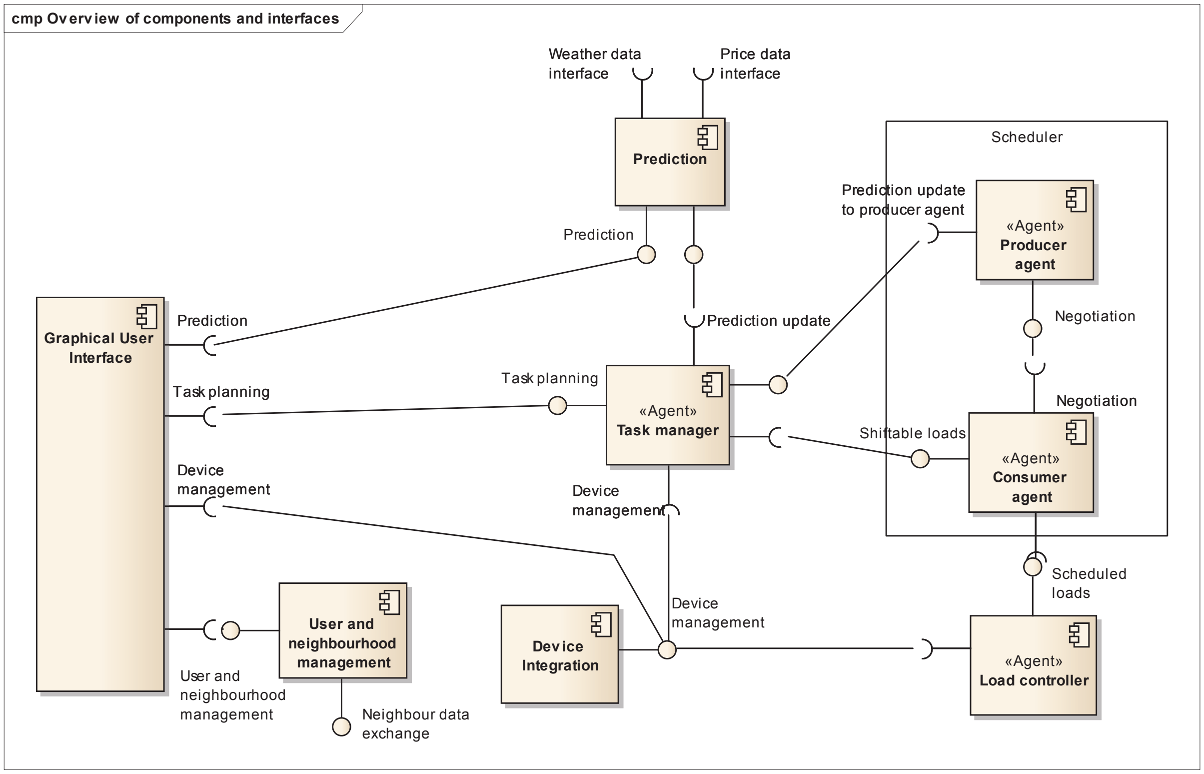

Fig. 2: Overview of components and interfaces

where a microgrid can buy energy when locally generated PV power is insufficient.

The *Device integration* provides a Device management interface for accessing the device status data, device measurement data, and other important information for optimization, e.g., negotiation results. In addition, the interface is also used to control individual device, e.g., switching on or off the device, and setting configuration parameters or selecting programmes if allowed.

The *User and neighborhood management* component is in charge of keeping track of the users of the system and their roles and privileges, and the other households that are members of the neighborhood. It also provides interfaces for the GUI to support the definition and inspection of these aspects.

B. Coordination Architecture of Collaborating Microgrids

Figure 3 shows a tree view of the collaborating microgrids. Each microgrid has a home gateway, which has capacity to firstly, act as a gateway for communication, both with devices within the household and with other neighbors; and secondly, execute the intelligence based on distributed computing. Interconnected gateways represent collaborating microgrids that form neighborhoods. Each microgrid is an autonomous subsystem until it joins a neighborhood. The leaves of a neighborhood sub-tree make up the multi-agent system (MAS). The agents in a neighborhood communicate with each other directly, *e.g.*, a consumer agent can negotiate with any producer agent in the whole neighborhood.

Agents compose a population of consumer and producer agents whose *perceptions* from the environment are the monitored information about the current energy consumption and production of devices, weather forecast, planned utilization of user's devices, and predicted background consumption. Perceptions are used to update agents' *believes*, which define energy requirements and availability of the handled devices. In particular believes are defined as cumulative energy profiles, which describe the time distribution of energy consumed by appliances and produced by PV panels. Moreover other believes describe the utilization plan of user's device. Agent's *desire* is the maximization of green energy utilization. Consumer agents aim at consuming energy produced by PV panels. Producer agents aim at distributing all their energy to other agents belonging to the neighborhood. The global objective is to minimize the grid energy used by the neighborhood. Agents' *intentions* are executed for learning profiles of consuming appliances, for predicting production profiles of PV panels and for negotiation. The goal is that the emergent behavior of the MAS will be close to an optimal schedule of devices utilization that maximizes the neighborhood self-consumption according to the users' preferences and constraints.

The MAS forms a P2P overlay to support communication and negotiation among agents as well as neighborhood management. More details about the implementation of the MAS will be presented in Section V.

V. Prototype Implementation

The described system has been implemented and runs on a selected technology baseline of software and hardware. The

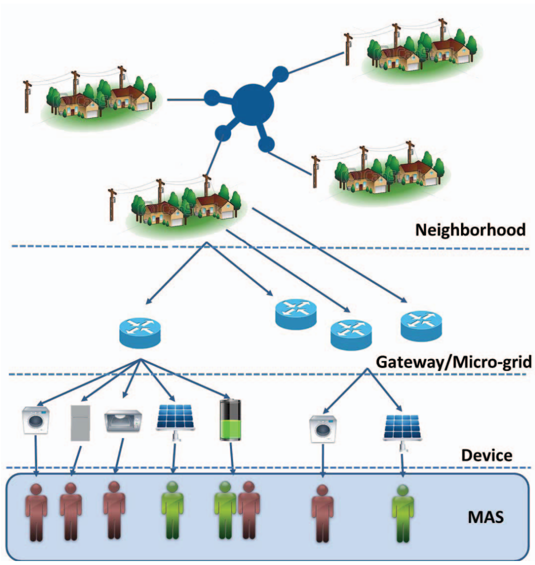

Fig. 3: A tree view of the collaborating microgrids

technologies have been selected with the goal to facilitate the realization of a system that can be used in the planned trial as well as capable to adapt to the expected evolution, and to enable the low threshold and widely applicable technology profile, meaning that the system must run on cheap hardware and be able to integrate appliances and solar systems in common use today.

Based on the above concerns, the *Raspberry PI*[1] has been chosen as the home gateway for the trial, as it is a widely used microcomputer, and is cheap and easy to use. It is possible to deploy the system on other microcomputers or existing home gateways that have the right hardware and software environment.

The GUI and Device integration components are implemented by extending *Emoncms*[2], a powerful open source web application for processing, logging and visualizing energy, temperature and other environmental data. By adopting Emoncms as a basis for the implementation, the reimplementation of the offered functionality is to a large extent avoided and efforts have been concentrated on the key contribution of the CoSSMic system, namely the coordination of the energy use in neighborhoods and the hybrid control mechanism.

Emoncms is a configurable framework with many modules that can be combined in different ways. It includes five core modules: input, feed, visualizations, dashboard, and user. The input module preprocesses meter input before it is inserted in the database, *e.g.* for creating histogram data. The feed module provides functionality for inserting, storing and retrieving time stamped data in the database. The visualizations module can analyze large datasets and generate graphics in different formats. The dashboard module includes the dashboard builder and viewer. The user module handles user actions and data, including authentication and sessions. A *Device Module* have been developed for managing and controlling devices. The device module consists of a device manager and a set of device instances in a microgrid. A device instance holds information about the device status, device type, different programmes and their load profiles. Nodes, i.e., meters, sensors and actuators, are used to get and set the status of the associated device, *e.g.* a washing machine or a fridge. Communication between the device manager and the nodes is done via drivers. Device drivers are used to enable integration of heterogeneous sources. In the current implementation, there are drivers for IEC62056[3], ModBus[4], Meter-Bus[5] and Ultra High Frequency (UHF) smart plugs available. In addition, a driver for eCar is available and can obtain information about the State of Charge (SOC). For testing reasons, also virtual devices can be created. The device manager offers a simple REST[6] Application Programming Interface (API) to support any kind of driver in the future. The extended Emoncms provides uniform access to monitored information to the agents based distributed scheduler and to the Graphical User Interface (GUI).

Although there are appliances on the market with electronic control interfaces, the market penetration is low and most appliances in use lacks such interface. Previous work has used a domotic system to integrate specific appliances for remote monitoring and control [1], [2]. To remedy the lack of appliance interfaces and to accommodate the ”dumb” devices allowing flexible installation and control of any devices, smart plugs are used to switch on and off connected devices. The user has to do the device setup and the task planning by a combination of the system user interface, and the manual controls and settings on the device, such as selecting a programme for a washing machine. For smarter devices in the future this will be much simplified as the Device Module can get information from the device and control the device directly.

The MAS is implemented by a mix of SPADE [22] and Theron[7] agents, communicating over the Extensible Messaging and Presence Protocol (XMPP[8]). SPADE is an open source agent platform written in Python. SPADE provides a library[9] that contains a collection of classes, functions and tools for creating agents that can work with the SPADE agent platform. Theron is a lightweight C++ based agent framework. The XMPP Protocol is a standard protocol implemented by both open source and commercial solutions. It provides P2P infrastructure to route messages, but it also defines presence management for implementing social relationships between agents and chatroom mechanisms to detect agents joining or leaving a neighborhood. The latter (*i.e.*, chatroom) is used to represent neighborhoods.

SPADE uses XMPP natively as transport protocol. An XMPP server provides various types of services: user account registration, authentication, channel encryption, prevention of address spoofing, message relaying, *etc.* Multiple servers can be deployed across the network to route and relay messages

[1]https://www.raspberrypi.org/
[2]http://www.emoncms.org

[3]International Electrotechnical Commission (IEC) standard for electricity metering data exchange
[4]http://www.modbus.org/
[5]European standard EN 13757-2 for the physical and link layer, and EN 13757-3 for the application layer
[6]Representational State Transfer
[7]http://www.theron-library.com
[8]http://xmpp.org/
[9]SPADE Agent Library

for workload balancing purpose.

The software platform hosted in the home gateway includes a light XMPP server that enables the communication between agents within the microgrid. It also allows for communication across different gateways setting up server-to-server connections with other XMPP servers.

Messages are routed within the neighborhood in a way that is transparent to the consumer and producer agents. Consumers and producers of the same neighborhood interact with each other according to a one-to-one negotiation protocol with a flat organization. However within the household there is a social organization of agents as shown in Figure 4 depending on the roles and tasks that agents have to carry out. For each microgrid a *task-manager* agent is responsible for abstracting tasks to loads and handing the loads to the corresponding consumer agents. It also gets weather prediction updates, predicts energy production from PV panels, and notifies prediction of production to the *actor-manager* agent. The actor-manager agent is responsible for creating producer and consumer agents. It also starts concurrent negotiations where local consumer agents negotiate their load requests with producer agents of the neighborhood, and returns negotiation results to the task-manager. The task-manager agent and the actor-manager agent realize the *Task manager* component in Figure 2. A *controller* agent is responsible for the execution of the schedule. The controller agent realizes the *Load controller* component in Figure 2.

The social organization is implemented using the mechanisms of the XMPP protocol. In order to join the P2P overlay agents use an XMPP account, which is identified by contact ID (*username@domainname*) and password. As the same user can connect at the same time by different devices (smartphone, PC, tablet, etc.), in the same way different agents can connect sharing the same XMPP account. In fact different connections with the same account are distinguished by the resource element in the XMPP address:
username@domainname/resource.

For each XMPP account it is possible to define the list of known contacts. The subscription of an agent to the contact lists of another defines their relationship and provides information about the mutual presence. A contact list of an XMPP user is called its roster. The XMPP server is aware about the rosters and forwards the presence information of subscribed contacts to each user.

In the CoSSMic system an XMPP user represents an agent role. Therefore, for each microgrid three types of XMPP accounts are used by agents: The task-manager account is subscribed to the actor-manager and to the controller accounts. In this way the task manager is informed about the presence of all the agents belonging to his own microgrid.

Agents responsible for negotiation are referred to as *actors*, therefore, the *actor* account is used by the *actor-manager* agent and by all the consumer and producer agents of the same microgrid. The actor agents connect to the XMPP overlay with different *resources*, such as manager, consumer1 and producer1 in Figure 4. The actor-manager joins a multi-user chatroom to know about the new microgrids that join the neighborhood. It subscribes to all *actor* accounts of the neighborhood and will thus get an updated view of the neighborhood. In this way all consumer agents will be aware of the presences of all producer agents in the neighborhood when they start a negotiation.

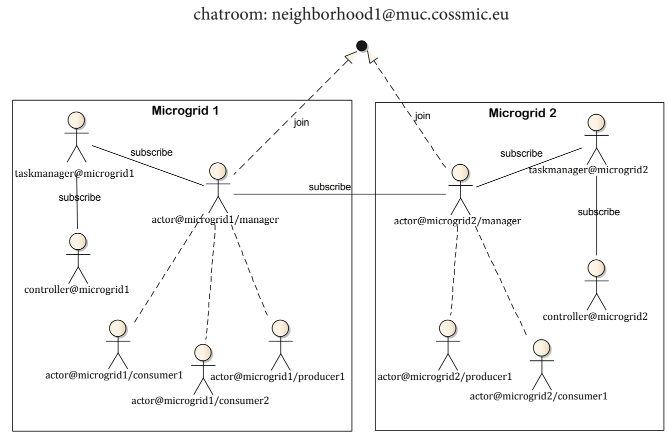

Fig. 4: Social organization of agents

A distributed negotiation protocol is designed. The software agents execute the protocol autonomously for the exchange of proposals and of the Service Level Agreement (SLA) described using machine readable templates, which describe different properties of the energy requirements including user's preferences and constraints.

The negotiation protocol is shown in Figure 5 and 6. There are two kinds of transactions. In Figure 5 negotiation is triggered by a new load to be allocated. The consumer agent uses a ranking of producers to choose the first one to ask for required energy. The producer matches the request with its energy availability and refuses the transaction or returns a start time for that load, which potentially updates the previous agreements. The consumer updates the ranking and asks the next producer or notifies the new schedule to the task manager, which alerts the controller.

In Figure 6 the transaction is triggered by a prediction update or by a planned load that has been canceled. The first message is a prediction update sent to the producer. The producer checks if its schedule is still valid. The second message is sent by the producer, which cannot satisfy an agreement with the updated prediction. Then the task-manager submits the load again to the consumer in order to look for a new agreement that is closed with producer1 of microgrid2. The last two messages of Figure 6 show the case where an agreement is terminated because the user of microgrid1 changes his plan and deletes a task. The task-manager of microgrid1 notifies the agreement termination to the task-manager of microgrid2, which alerts his producer to free the allocated energy.

For prediction of PV production, an agent monitors the weather forecast updates from the DWD weather forecast service (currently updated every six hours) and alerts the neighborhood about the updates using chatroom. The respective Prediction module then calculates the predicted PV production based on the weather forecast updates and the parameters of the local PV installation, and alerts the producer.

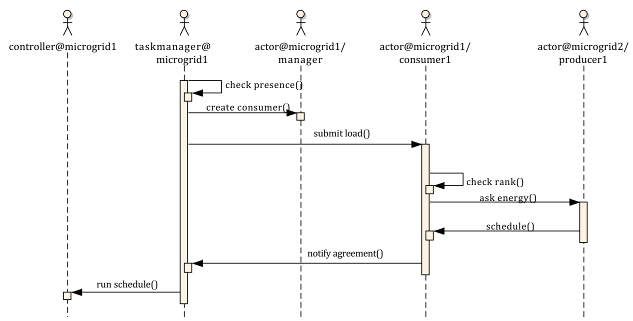

Fig. 5: Scheduling of new load

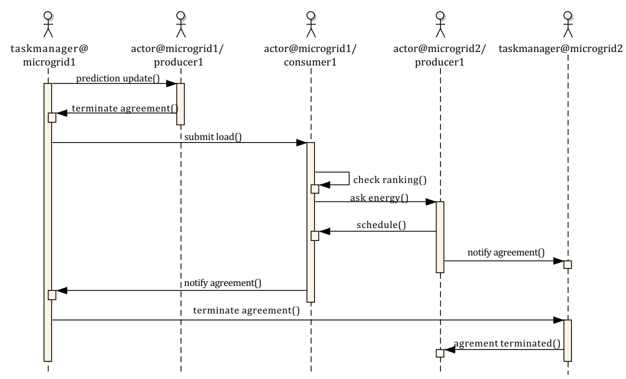

Fig. 6: Termination of agreement

VI. Evaluation

The approach described in this paper will be evaluated based on two types of evaluation:

1) Early evaluation based on user-centered design principles [23] and user involvement: User requirements have been collected from interviews and workshops at the two trial sites in Germany and Italy with potential trial users. Furthermore, the GUI has been designed following user-centered approach, and the concepts have been co-developed and validated with stakeholders and users in user centered design workshops [24], [25].

2) Final evaluation based on field trials: Trials with real users are now being carried out in two trial neighborhoods in the City of Konstanz, Germany and the Province of Caserta, Italy. The trial neighborhoods include private households, industrial buildings and schools, thus ensuring varied usage patterns and needs. Measurements of users' production and consumption have already been recorded since early 2015. The collected data include:

- The trace of the PV production prediction updates during the trials tagged with household, device and timestamp
- The trace of all grid price prediction updates tagged with timestamp
- The trace of all task planning events, also tagged with household, device and timestamp
- The trace of the output from all the producer devices
- The trace of the consumption per device
- The trace of the total energy consumption and production per house

The trials will last one full year to cover the seasonal variation in PV output and energy demand. Based on the data collected in the trials, the CoSSMic system will be evaluated against the following Key Performance Indicators (KPIs): self-consumption of the whole neighborhood, stacked peak avoidance, optimization of storage cycles, prediction accuracy, scalability, effect of collective PV systems on total power fluctuation, usability of software, energy cost reduction and return of investment cost.

As the trial only demonstrates two examples (*i.e.* neighborhoods) and it is not practical to substantially increase the number of users due to resource constraints, this limits what conclusions can be drawn only from observing them. Simulation can thus be used to evaluate the objectives with the goal to increase the value of the trial and produce more interesting results. With simulation user behaviors recorded during the trials can be replayed with varying neighborhood configurations, *e.g.* adding a new consuming device or a battery, changing the price models assuming futuristic dynamic pricing signals, and increasing artificially the prediction accuracy. In addition, with simulation, it is possible to increase the size of neighborhood by replicating users. The simulation software will reuse most of the software components implemented for the trials, but will execute them in a simulated environment.

VII. Conclusion

In this paper an architecture and a hybrid control mechanism have been proposed to coordinate the consumption and storage in a neighborhood of microgrids with local renewable energy generation, aiming at maximizing the self-consumption of locally produced energy. The mechanism combines distributed planning and scheduling based on predictions with a reactive feedback loop to compensate the inability to predict accurately the rapid fluctuations in PV output due to passing clouds. The design and implementation of a distributed multi-agent based system that realizes this mechanism has been described as well. Each microgrid is implemented as a Multi Agent System (MAS) with distributed intelligence, and the neighborhood has a coordination architecture based on peer-to-peer (P2P) overlay to facilitate the plug-in of a new microgrid into the neighborhood. The P2P solution allows the system to operate without centralized authority, favoring distribution and local autonomy, and providing tolerance to failures of each individual microgrid.

Technologies that use or comply with open protocols and standards have been chosen to achieve low cost, and allow integration with other smart home systems. Smart plugs are used as a flexible and low threshold technology to monitor energy consumptions and switch on or off any connected devices instead of using a dedicated domotic system for specific appliances. Smart appliances with electronic control interfaces will become more common in the future, and such interfaces can be used to access device status and control the operation automatically.

The system has been implemented with the possibility to collect data, control devices, shift/schedule loads, and remotely update the software. It is currently under a one-year trial with real users. Early evaluation based on user-centered design workshops has resulted in a working GUI validated by potential users. Final evaluation will be carried out using experiments and simulation based on the data collected during the trial. With simulation it is expected to produce more interesting results, and increase the value of the trial by varying neighborhood configurations, pricing models, and prediction accuracy.

ACKNOWLEDGMENT

This work has been partially funded by EU CoSSMic project under FP7 Grant agreement no. 608806.

REFERENCES

[1] M. Castillo-Cagigal, E. Caamao-Martn, E. Matallanas, D. Masa-Bote, A. Gutirrez, F. Monasterio-Huelin, and J. Jimnez-Leube, "Pv self-consumption optimization with storage and active dsm for the residential sector," *Solar Energy*, vol. 85, no. 9, pp. 2338 – 2348, 2011.

[2] M. Castillo-Cagigal, A. Gutirrez, F. Monasterio-Huelin, E. Caamao-Martn, D. Masa, and J. Jimenz-Leube, "A semi-distributed electric demand-side management system with pv generation for self-consumption enhancement," *Energy Conversion and Management*, vol. 52, no. 7, pp. 2659–2666, 2011.

[3] M. van der Kam and W. van Sark, "Increasing self-consumption of photovoltaic electricity by storing energy in electric vehicle using smart grid technology in the residential sector - a model for simulating different smart grid programs," in *3rd International Conference on Smart Grids and Green IT Systems (SMARTGREENS 2014)*, 2014, pp. 14–20.

[4] Linas Gelazanskas and Kelum A. A. Gamage, "Demand side management in smart grid: A review and proposals for future direction," *Sustainable Cities and Society*, vol. 11, pp. 22–30, Feb. 2014.

[5] B. Schafer, M. Matthiae, M. Timme, and D. Witthaut, "Decentral smart grid control," *New Journal of Physics*, vol. 17, pp. 22–30, Jan. 2015.

[6] M. S. Narkhede, S. Chatterji, and S. Ghosh, "Multi-agent systems (mas) controlled smart grid - a review," *IJCA Proceedings on International Conference on Recent Trends in Engineering and Technology 2013*, vol. ICRTET, no. 4, pp. 12–17, May 2013.

[7] J. Jimeno, J. Anduaga, J. Oyarzabal, and A. Gil de Muro, "Architecture of a microgrid energy management system," *Euro. Trans. Electr. Power*, vol. 21, pp. 1142–1158, 2011.

[8] C. Yilmaz, S. Albayrak, and M. Lutzenberger, "Smart grid architectures and the multi-agent system paradigm," in *Proceedings of The Fourth International Conference on Smart Grids, Green Communications and IT Energy-aware Technologies, ENERGY 2014*, 2014, pp. 90–95.

[9] A. Kantamneni, L. E. Brown, G. Parker, and W. W. Weaver, "Survey of multi-agent systems for microgrid control," *Engineering Applications of Artificial Intelligence*, vol. 45, pp. 192–203, 2015.

[10] D. Adolf, "Agent based architecture for smart grids," Semester thesis, ETH Zurich, 2010.

[11] A. Dimeas and N. Hatziargyrious, "A mas architecture for microgrids control," in *Proceedings of the 13th International Conference on Intelligent Systems Application to Power Systems*, 2005, pp. 402–406.

[12] A. L. Dimeas and N. D. Hatziargyriou, "Operation of a multiagent system for microgrid control," *IEEE Transactions on Power Systems*, vol. 20, no. 3, pp. 1447–1455, August 2005.

[13] M. Pipattanasomporn, H. Feroze, and S. Rahman, "Multi-agent systems in a distributed smart grid: Design and implementation," in *Power Systems Conference and Exposition*. IEEE, 2009, pp. 1–8.

[14] A. Dimeas and N. Hatziargyrious, "Agent based control for microgrids," in *Power Engineering Society General Meeting*. IEEE, 2007, pp. 1–5.

[15] Jens Leth Hougaard, *An Introduction to Allocation Rules*. Springer Berlin Heidelberg, 2009. [Online]. Available: 10.1007/978-3-642-01828-2

[16] K. e. a. Kok, "Smart houses for a smart grid," in *Proc. of 20th Int. Conf. on Electricity Distribution*, 2009, pp. 1–4.

[17] K. Kok, "Multi-agent coordination in the electricity grid, from concept towards market introduction," in *Proc. of 9th Int. Conf. on Autonomous Agents and Multiagent Systems*, 2010, pp. 1681–1688.

[18] A. Dounis and C. Caraiscos, "Advanced control systems engineering for energy and comfort management in a building environment-a review," *Renewable and Sustainable Energy Reviews*, vol. 13, pp. 1246–1261, 2009.

[19] D.-Y. Yu, E. Ferranti, and H. Hadeli, "An intelligent building that listens to your needs," in *Proceedings of the 28th Annual ACM Symposium on Applied Computing*. ACM, 2013, pp. 58–63.

[20] Geir Horn, Salvatore Venticinque, and Alba Amato, "Inferring appliance load profiles from measurements," in *Proceedings of the 8th International Conference on Internet and Distributed Computing Systems (IDCS 2015)*, ser. Lecture Notes in Computer Science, Giuseppe Di Fatta, Giancarlo Fortino, Wenfeng Li, Mukaddim Pathan, Frederic Stahl, and Antonio Guerrieri, Eds., vol. 9258. Conference Location: Windsor, UK: Springer International Publishing, Sep. 2015, pp. 118–130.

[21] Geir Horn, "Scheduling time variant jobs on a time variant resource," in *The 7th Multidisciplinary International Conference on Scheduling : Theory and Applications (MISTA 2015)*, Zdenek Hanzálek, Graham Kendall, Barry McCollum, and Premysl Šůcha, Eds., Conference location: Prague, Czech Republic, Aug. 2015, pp. 914–917.

[22] M. E. Gregori, J. P. Cámara, and G. A. Bada, "A jabber-based multi-agent system platform," in *Proceedings of the Fifth International Joint Conference on Autonomous Agents and Multiagent Systems*, ser. AAMAS '06. New York, NY, USA: ACM, 2006, pp. 1282–1284.

[23] J. Kubie, R. Melkus, L.A. Johnson, and G. Flanagan, Eds., *IS Management Handbook: 7th Edition*. CRC Press, 2000, ch. User-Centered Design, pp. 463–480.

[24] L. Wienhofen, C. Lindkvist, and M. Noebels, "User-centered design for smart solar-powered micro-grid communities," in *Proceedings of the 14th International Conference on Innovations for Community Services (I4CS), 2014*. IEEE, 2014, pp. 39–46.

[25] J. Glatz-Reichenbach, T. Vilharino, G. Cretella, C. Lindkvist, A. Minde, and L. W. M. Wienhofen, "End user centred interactive software architecture and design: The creation of communities for a smart energy use," in *Proceedings of the 14th International Conference on Innovations for Community Services (I4CS), 2015*. IEEE, 2015, pp. 1–8.