A Generic Holonic Control Architecture for Heterogeneous Multi-Scale and Multi-Objective Smart Micro-Grids

Sylvain Frey, Télécom ParisTech, CNRS LTCI, EDF R&D
Ada Diaconescu, Télécom ParisTech, CNRS LTCI
David Menga, EDF R&D
Isabelle Demeure, Télécom ParisTech, CNRS LTCI

Designing the control infrastructure of future "smart" power grids is a challenging task. Future grids will integrate a wide variety of heterogeneous producers and consumers that are unpredictable and operate at various scales. ICT solutions will have to control these in order to attain global objectives at the macro-level, while also considering private interests at the micro-level. This paper proposes a *generic holonic architecture to help the development of ICT control systems* that meet these requirements. We show how this architecture can integrate heterogeneous control designs, including state-of-the-art smart grid solutions. To illustrate the applicability and utility of this generic architecture we exemplify its use via a concrete proof-of-concept implementation for a holonic controller, which integrates two types of control solutions and manages a multi-scale multi-objective grid simulator in several scenarios. We believe that the proposed contribution is essential *for helping to understand, to reason about and to develop the "smart" side of future power grids*.

Categories and Subject Descriptors: Computer Systems Organisation [**C.0**]: General

General Terms: Design

Additional Key Words and Phrases: Autonomic Smart Micro-Grid, Integration of Multi-Scale and Multi-Objective Control Systems, Holonic Architecture, Integration Design Patterns, Proof-of-Concept Simulation

ACM Reference Format:
Sylvain Frey, Ada Diaconescu, David Menga and Isabelle Demeure, 2014. A Generic Holonic Control Architecture for Heterogeneous Multi-Scale and Multi-Objective Smart Micro-Grids. *ACM Trans. Autonom. Adapt. Syst.* ?, ?, Article ? (? 2014), 22 pages.
DOI:http://dx.doi.org/10.1145/0000000.0000000

1. INTRODUCTION

Over the next decades, the electric grid is expected to undergo massive changes both in its composition and control infrastructure [Callaway and Hiskens 2011] [Marnay and Venkataramanan 2006] [Negrete-Pincetic and Meyn 2011]. The increasing introduction of renewable energy sources brings unprecedented unpredictability in grid energy production. This, along with the rise in local storage technologies, can significantly disrupt the current distribution of energy flows. Hence, the grid's control infrastructure must also change to deal with the new energy fluctuation patterns. Information and Communication Technology (ICT) is a necessary extension for transforming power grids into "smart" grids that can answer such requirements. Most experts in industry and research communities agree that the smart grid's control should be more decentralised, yet no consensus exists so far on a common control solution [Callaway and Hiskens 2011] [Marnay and Venkataramanan 2006] [SGAM 2012].

DOI:http://dx.doi.org/10.1145/0000000.0000000

While essential to modern grids, ICT-based control also introduces new challenges. Since more and more energy production and storage facilities can be owned privately, the energy market is becoming progressively less centralised. Many actors, including private consumers, can define their power-management *objectives* (or *goals*), like business profits, bill reduction or environment preservation. The smart grid's ICT infrastructure is key for pursuing such goals, since it enables the independent control of various grid parts – from countries and areas to districts, houses and devices. Yet, such new facilities can further increase the grid's dynamism and unpredictability, hence making its control even more challenging [Negrete-Pincetic and Meyn 2011]. Furthermore, they introduce *additional requirements* on the smart grid's overall design.

Firstly, the diversity of interested parties, with different assets, business goals and application contexts, are likely to prevent consensus on a unique grid control solution [SGAM 2012]. Rather, *heterogeneous control solutions may have to be adopted in different grid sections*. Secondly, the pursuit of *private goals in various smart grid parts (micro-level) may often go against the global goals of the overall grid (macro-level)*. Finally, smart grids develop in parallel with other socio-technical systems, such as smart appliances and houses, with their own constraints and goals. *Integrating such smart technologies within the smart grid will raise further management conflicts*.

The ad-hoc integration of *heterogeneous multi-goal grid control solutions*, entangled with other smart systems, can jeopardise the very benefits that motivated the transition towards distributed power systems. Hence, great care must be taken when designing the smart grid's general *integration infrastructure*, to prevent global disasters from emerging out of local self-interested decisions. Composition of disparate, local control solutions into coherent organisations, ensuring stability, safety, and, when possible, fair balance between micro and macro goals, is a key step of the design process.

In this paper, we capitalise on our Software Engineering (SE) experience in building integration solutions for scalable, heterogeneous, multi-objective self-* systems. Previous work [Frey et al. 2013b] proposed a *generic architecture and methodology* for designing adaptive controllers for such systems. It was complemented by a catalogue of *integration design patterns* [Frey et al. 2012]. In this paper we propose a holonic architecture to enrich this approach and refine it for the smart micro-grid domain.

The main contributions include:

— identifying **requirements** on grid controller design that highlight the need for an **integration-driven approach** (section 2);
— proposing a **generic holonic architecture** that enables the recursive integration of control solutions of various types in smart micro-grids; this focuses on the generic specification of structural, functional and interface-related elements (sections 3 and 4), most notably:
— providing an **abstract formalisation of control goals** and showing how this is a key step for identifying and interfacing controllers via goal operations, such as translation, splitting, composition, conflict detection and resolution (subsection 3.2);
— showing how state-of-the-art grid control applications mostly follow **three generic integration patterns** presented in [Frey et al. 2012] (with variants) and specifying these as **reusable organisations** (subsections 4.3, 4.4 and 4.5), which can be integrated via the holonic meta-pattern (subsection 4.6).

We **exemplify** the proposed holonic architecture via a **concrete smart grid design and implementation** (section 6). We show how two control organisations, based on different patterns, can be integrated into a multi-scale multi-goal micro-grid. **Experimental results** from several power management scenarios, run on a distributed grid simulator, illustrate how such integrated controller can function and indicate the applicability and usefulness of the approach. While the concrete example is greatly

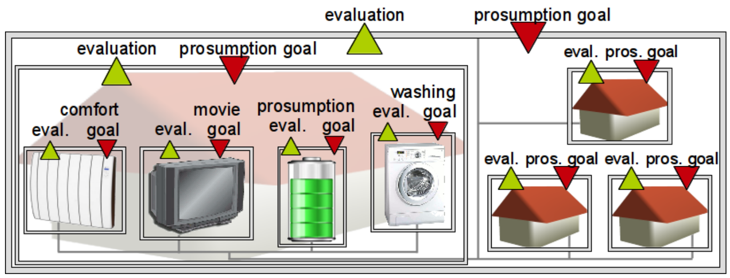

Fig. 1. Simplified micro-grid with district-level, house-level and appliance-level goals.

simplified with respect to a real smart grid, its purpose is to highlight how solutions based on the holonic architecture can deal with multi-authority, heterogeneity, multi-goal and multi-scale concerns. We believe that this contribution lays a good foundation for developing grid control systems that can address the important issues identified. We also hope that it is sufficiently generic for becoming relevant to other application domains, such as pervasive computing, or socio-technical systems in general.

The content of this paper is as follows. Section 2 introduces a simplified grid model, insisting on control goals and integration requirements. Section 3 provides a goal-oriented view and generic formalisation of the control architecture. Section 4 describes the integration-oriented architectural view and the identified patterns via concrete examples from literature. Section 5 focuses on related work in holonic systems and generic smart grid architectures. The proof-of-concept implementation and results are discussed in Section 6, followed by conclusions and future work proposals.

2. REQUIREMENTS FOR SMART GRID MANAGEMENT

2.1. A simplified smart grid model

This paper focuses on low-voltage small-scale electrical networks, or *micro-grids*. A classic example is that of a district grid regrouping a few dozens houses, each featuring variable power consumption and production profiles. Such micro-grid is seen here as a hierarchichal system integrating power producers and consumers via a distribution network [Fig. 1]. Leaves represent end-user producers – like solar panels; consumers – like city lights or washing machines; or both – like batteries. The term *prosumer* designates grid elements that can both produce and consume, at different times [Toffler 1980]. A *prosumption* is measured in Watt. We consider that a positive prosumption is a consumption and a negative one a production, which simplifies the definition of prosumption objectives (cf. subsection 2.2). With regard to the hierarchy, internal nodes delimit hierarchical sub-networks: a house electricity meter defines the boundary of the house's local network, which is in turn nested within a district network.

Traditional power grids consisted of a majority of consumers supplied by a few large producers via one-way power transport and distribution networks. Load control used to be quite centralised, and top-down, adjusting production to consumption estimates at a regional or national level. With the progressive introduction of renewable energy sources and batteries at any grid level this situation may change. Production may surpass consumption at any grid level, causing inversions of the energy flow through the grid [Callaway and Hiskens 2011]. Hence, any internal grid node can also be modeled as a prosumer. From the point of view of its parent, a sub-network can be seen as a single prosumer, with a total prosumption that equals the sum of its child prosumptions. For instance, a house's prosumption is the sum of prosumptions of all its electric devices; the same applies to a district's prosumption with respect to its houses.

The purpose of this model is to reflect those core features of smart grids, including multi-scale and multi-goal, which the proposed solution addresses. This should ensure that **our generic contribution still applies when grid models are refined**.

2.2. Control goals and conflicts in smart grids and smart houses

The main grid control goal is to *balance power productions and consumptions*, avoiding peaks and flicker, at different grid scales [Kok et al. 2005] [Wedde et al. 2008]. Over short periods, like 0.5s, this goal avoids blackouts [Wedde et al. 2008] by managing either consumption or production. Non-null prosumptions must be compensated for with power from local sources or the parent network. At larger time scales, like hours, days, or months, this goal helps to plan prosumptions based on predictions [Kok et al. 2005; Vinyals et al. 2014; Allerding and Schmeck 2011]. Another important goal is *cost*-related, with consumers aiming to minimise bills and producers to maximise profits. Hence, many research approaches use cost incentives to address the power balancing goal, by adopting micro-economy concepts [Vinyals et al. 2014],[Kok et al. 2005],[Wedde et al. 2008]. This paper assumes no specific pricing or market regulation. It provides a generic solution that can be adapted to various norms. Other goals, including *ecological* ones, can be included later within the same general framework.

We suppose that one administrative authority (human or autonomic) is in charge of each sub-grid (e.g. house and district grid). Their prosumption goal for their grid is a viability interval $[p^{min}, p^{max}]$ over a time period. In the context of future grids with significant production capabilities, excessive production is as much a threat as excessive consumption, which justifies the use of target intervals instead of maximum production limits only. Power intervals are chosen arbitrarily in the paper to ensure the flexibility necessary for embracing a wide range of realistic goals later on. The generic model supports goals defined over longer intervals (cf. section 3) yet for simplicity we only exemplify instant goals in our experiments.

Additional goals are related to home owners' *comfort* and translated into *smart house services* – e.g. electric appliances provide heating or entertainment. Home owners are also interested in power management goals (as above) [Fig. 1]. Since these may interfere with comfort goals – e.g. thermostats consume electricity – the grid controller must cater for such conflicts, even when working under a single authority. Additional conflicts may occur between the power goals of different authorities, either at different grid scales – e.g. home consumptions cause district-level peaks; or different time intervals – e.g. incompatible day-ahead plans and current prosumptions.

2.3. Requirements for smart micro-grid controllers

To clearly delineate the relevance of our contribution, we focus on the following subset of smart grid requirements that demand the use of integration-oriented architectures (detailed in [Frey et al. 2013b]):

— **Multi-authority and multi-scale:** Several administrative authorities operate the grid, each one defining their goals over a grid part (appliance, house, district);
— **Multi-objective:** The authorities controlling the grid may define additional goals, such as comfort or security, which may interfere with power management;
— **Heterogeneity:** Prosumers are widely diverse in their capabilities, design settings, usage context, preferences, life-cycles, and so on;
— **Scalability:** Control architectures and algorithms must scale with the number and dynamism of prosumers;
— **Incremental change:** The transition from traditional to smart grids must be progressive, for technical, economical and/or political reasons.

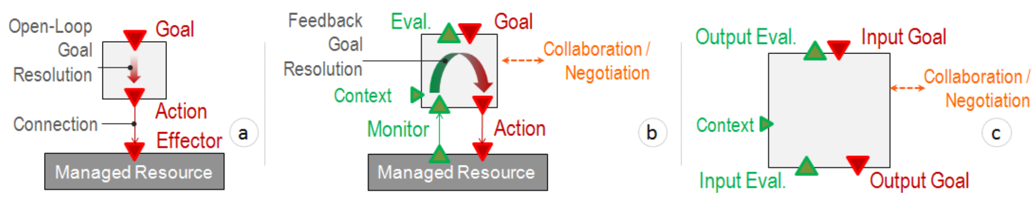

Fig. 2. Goal resolution: a) Open-Loop; b) Closed-Loop (Feedback); c) Black-Box view

Based on these considerations we argue that no single control design can address all requirements across the entire smart-grid [Schmeck and Karg 2010]. A **mixed solution integrating several heterogeneous controllers** is needed, requiring:

— **Standard taxonomies and protocols:** To enable the (dynamic) integration of heterogeneous, third-party appliances and controllers in various environments;
— **Flexible micro-macro integration:** To ensure an adaptive balance between individual (micro) and collective (macro) goals, based on (dynamic) user specifications and grid regulations [Kok et al. 2005];
— **Meta-management feedback:** To evaluate management decisions (human or autonomic) and enable adaptive top-down regulation based on bottom-up evaluation.

3. GOAL-ORIENTED CONTROL ARCHITECTURE

3.1. Goal-Oriented Controller Overview

The ultimate purpose of a control system is to achieve one or several *goals* by executing one or several *actions* on system resources. Goals can be defined by either external entities, like human administrators or other controllers, or the controller itself. A *goal resolution* process determines the actions to perform for achieving a goal. A full classification of goals and resolution processes is beyond this paper's scope; we only focus on aspects relevant to the generic architecture presented. We adopt two well-known control approaches for the resolution process [Nolle et al. 2002], and later show how more complex controllers can be based on combinations of these. Firstly, *open-loop resolution* [Fig. 2-a] is triggered when the goal is defined (detailed below), and determines the actions to take based on a static model of resources and the environment. Secondly, *feedback-based resolution* [Fig. 2-b] determines and adapts the actions cyclically, based on the system state and the environment.

[Fig. 2-c] provides a black-box perspective of both controller types. It is compatible with the Autonomic Management [Kephart and Chess 2003] and Organic Computing [Allerding et al. 2011] reference architectures. *Input goal* defines a goal that is given to the controller to attain. *Output evaluation* provides feedback on the goal's achievement – e.g. an autonomous controller may refuse the goal, or only achieve it partially. *Output goal* defines the actions performed to achieve the input goal. We consider output actions to be conceptually equivalent to output goals sent to managed resources, since these can be actual system resources or other controllers (discussed below). *Input evaluation* provides the controller with feedback on the achievement of its output goals (only for feedback controllers). This allows it to replan its output goals or even reconsider its input goals. Finally, controllers may also feature non-goal related interfaces, such as for negotiation and collaboration purposes. It is important to note that these represent conceptual interfaces and can be defined optionally and differently in each concrete architecture. This controller representation helps designers to understand and reason about control systems with multiple authorities and goals.

3.2. Generic Formalisation of Goals, Goal Operations and Conflicts

In short, Goals are defined as a triplet (V, S, T), where:

— V is a *viability* constraint defining the system's desired state space or behaviour;
— S is a *scope* defining where, or over which managed resources V is to be evaluated;
— T is a *time* constraint defining when V is to be applied and evaluated.

So far we have focused on viability and scope parameters; we only exemplify time constraints in the paper to illustrate how the proposed model matches existing smart grid solutions. For instance, a power goal in a house can be expressed as $G_P^H([P_{Hmin} = 1kW, P_{Hmax} = 2kW], House_H, Period_{0.1})$, imposing a viable power interval (V), for a house (S) and over each 0.1 seconds (T). Similarly, a power goal can be defined over an entire district as $G_P^D([P_{Dmin} = 100kW, P_{Dmax} = 150kW], District_D, Period_{1.0})$.

Conflicts can occur between goals with incompatible viability constraints, when their scopes and times intersect [Frey et al. 2013b] – e.g. the goals exemplified above may become in conflict with a temperature goal $G_T^H([T_{min}, T_{max}], House_H, Period_{day})$. Goals may also be defined in a domain-specific manner more suitable for home owners: $G_C^H(`comfort', House_H, Period_{day})$. Finally, goals may be defined for appliances, like washing machines, with flexible times: $G_{on}^{wm}(`cycle'_x, wm, [9am, 5pm])$. Incompatible goal viabilities can be determined via goal *operations*. Controllers may transform input goals into output goals via two operations: *translation* and *splitting*. Input evaluations follow the inverse process: inverse-translation and composition. Both operations can be applied on any goal parameters (V,S,T), via open-loop or feedback-based processes.

Goal Translation transforms an input goal into an output action or output goal – e.g. mapping a 'comfort' goal G_C^H into a 'temperature' goal G_T^H and then into a 'power' goal G_P^H. This process may be implemented via more or less complicated procedures. Conceptually, goal translation serves either to reduce a goal's abstraction level – e.g. progressively translating business goals into technical actions –; or to integrate heterogeneous controllers with diverse goal representations and resolution protocols. Inverse-translation operations can also be defined to increase a goal's abstraction bottom-up (we have not exemplified this option so far).

Goal Splitting transforms an input goal into several output goals, where at least one of the input goal's parameters (V,S,T) are mapped into a set of new parameters. For instance, the viability of the power goal G_P^H (above) can be split into smaller power intervals (V_i), for smaller scopes – individual appliances (S_j), and/or smaller time intervals (T_k). Similarly, G_P^D can be split into several $G_{P_j}^{H_j}$, one for each house H_j. For the temperature goal G_T^H it only makes sense to split the scope and time parameters. Conceptually, goal splitting serves to reduce larger, more complex, goals into smaller, simpler goals. However, this does *not* necessarily imply a reductionist approach – i.e. splitting a parameter may not result in a $k{-}partition$ of sub-parameters. This accounts for cases where emergent phenomena enables a goal to be attained via a sub-division of goals that is not a partition of the initial goal. For instance, attaining the temperature goal in a house (G_T^H) may be achieved by splitting its scope into sub-scopes that only represent a subset of a house's rooms (e.g. $\bigcup S_k \neq S$); the goal may still be achieved due to heat transfer through doors and walls. As above, an inverse of this operation – composition – is also possible; we have so far equated it with conflict resolution.

Controllers can perform both operations on any of their input goals to determine their output goals. For instance, a room controller translates and splits a 'comfort' input goal into a 'temperature' and a 'luminosity' output goals. These become inputs for controllers that split their scopes among thermostats and lamps, respectively. If a light controller detects a new lamp it re-splits the 'luminosity' goal to include it. Thermostat controllers transform their 'temperature' goal into sequences of 'power' goals, triggered

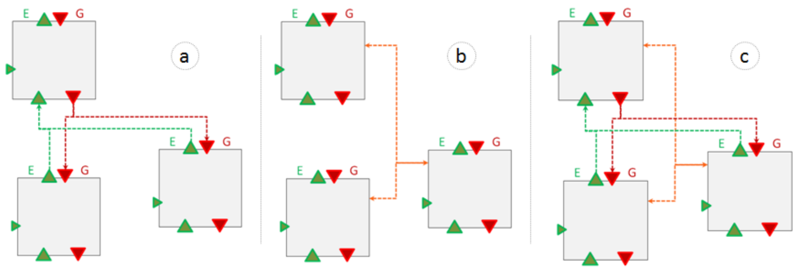

Fig. 3. Controller integration: a) via goal connections; b) via collaboration/negotiation; c) mixed.

in parallel or in sequence. These examples show how controllers can interconnect their input and output goal (and evaluation) interfaces to form more complicated functions [Fig. 3-a]. Alternatively, they can interconnect via their negotiation interfaces [Fig. 3-b] – e.g. agent pre-planning and price negotiation enable them to set their power goals [Kok et al. 2005] [Wedde et al. 2008]. [Fig. 3-c] depicts a mixed solution.

Projecting business goals on system resources via the above operations helps identify potentially conflicting goals and the necessity for ***Conflict-Resolution*** logic [Frey et al. 2012]. When *one* controller receives conflicting goals it may resolve them internally, in an application-specific way (e.g. Pareto optimisation). Here, conflict resolution can be viewed as a third operation that composes input goals (conflicting) into one output goal (coherent). When *different* controllers receive conflicting goals a specific *integration pattern* is needed [Frey et al. 2012] (subsections 4.3–4.5).

So far, our examples illustrated the top-down direction of goal analysis and resolution. However, the generic architecture also supports bottom-up approaches: firstly, by connecting the output goals of 'lower-level' controllers to the input goals of 'higher-level' controllers; and secondly, by allowing 'higher-level' controllers to redefine their output goals and their connections to 'lower-level' controllers, based on feedback from these via evaluation interfaces. The latter process can propagate upwards as controllers exhaust their resolution options and must change their input goals or provide negative evaluations upwards. This bi-directional adaptation process – or Yo-Yo design [Tomforde and Müller-Schloer 2014] – will be the subject of future work.

4. INTEGRATION-ORIENTED ARCHITECTURE

4.1. Integration Architecture Overview

Composing controllers into more complex control systems raises several integration issues, including: conflict resolution – who has priority?; time-related coordination – who executes when?; and communication – what information and exchange protocols?. To help address these issues from an architectural perspective we propose three *integration patterns*, based on concrete examples from the smart grid literature: ***hierarchy, stigmergy*** and ***collaboration***. We also propose ***a holonic meta-pattern*** for recursively integrating various instances of these patterns, at different grid scales. This paper only focuses on discussing the conflict resolution concern, even if some generic aspects of coordination and communication are also included implicitly in the pattern descriptions. Future publications will focus on these latter concerns in more detail.

Patterns are expressed here in terms of *organisations* of controllers. We have borrowed the *organisation* concept from the multi-agent systems (MAS) domain, where it is widely employed to deal with system dynamism and openness [Weyns et al. 2010] [Cabri et al. 2004]. An organisation specifies abstract *roles* – in our case implemented by concrete controllers; and *messages* – exchanged among role players.

Individually, each *power management organisation* captures invariant structures of state-of-the-art grid control systems in the domain. When combined, the resulting organisations are meant to be: sufficiently generic to allow standardisation and to favour re-usability (cf. "heterogeneity" and "incremental change" requirements in subsection 2.3); and, sufficiently expressive to help design controllers that answer specific smart grid constraints (cf. "multi-scale", "multi-objective" and "scalability"). This contribution offers designers a reusable base for experimenting with widely diverse smart grid solutions, where controllers based on different patterns can be combined. We believe this process is essential for addressing all the smart-grid requirements identified above.

4.2. Generic context of integration patterns

From now on, we focus on how the identified patterns address conflict-resolution. Future studies will analyse other integration concerns, particularly time-related. All patterns must address a common set of conflicts. We discuss these here and the pattern-specific requirements in corresponding subsections. As discussed in 3.2, conflicts may arise between goals that define incompatible viability constraints (or actions) on shared resources and over overlapping time frames. The same applies when resources and time frames influence each other indirectly [Frey et al. 2013b]. Conflicts can be inherent in the administrative goals or occur after translation and splitting. Such goals may or may not belong to the same authority and may or may not define the same type of viability parameters, like temperature or power.

In a smart micro-grid, we consider the following conflict situation, involving several smart houses in a district. The patterns should still apply when this simplified context is refined. A district power manager defines a 'prosumption' goal G_P^D (all sample goals were defined in 3.2). This goal is split dynamically into 'prosumption' goals for each house $G_{P_j}^{H_j}$. Each home owner defines 'prosumption' and 'comfort' goals for their house H_j – ${G'}_{P_j}^{H_j}$ and ${G'}_{C_j}^{H_j}$, respectively. These two goals may be in conflict with the district manager's goal G_P^D, which after being split intersects the house scope H_j. Further goal translation and splitting over rooms and appliances reveals further potential conflicts; especially since home owners may add appliance-specific goals, like G_{on}^{wm}. Maintaining $25°C$ in all rooms and running a washing cycle, while consuming less than $10kW$, may be problematic. Finally, conflicts may arise between goals defined over different time scales, like a house's day-ahead 'prosumption' goal based on prediction and its real-time 'prosumption' goals over a period of 0.5s.

These examples illustrate conflicts between goals of the same type (e.g. 'prosumption' from different authorities at different scales) and of different types (e.g. 'prosumption' and 'comfort' in a house). To resolve these conflicts some goals must take precedence over others (e.g. favouring energy savings over comfort), or a compromise must be found (e.g. Pareto optimum). Such preferences may be customised across equipments and services (e.g. accept dimmed lights but not low temperatures). Concerning time-based conflicts, short-term power balancing goals should always take precedence over prediction-based goals to avoid blackouts. This paper does *not* aim to contribute a particular algorithm for conflict-resolution, price negotiation or multi-criteria optimisation, but rather to indicate *where* such algorithms should be distributed within the general architecture: within a single controller if its input goals are conflicting; across a conflict-resolution pattern if conflicts affect several controllers; or within a holonic controller's membrane (cf. subsection 4.6) if resolving micro-macro conflicts.

4.3. Hierarchy pattern

Context: Conflicting controllers are limited in number; some of their properties and behaviours can be known in advance and/or obtained dynamically; and they can take

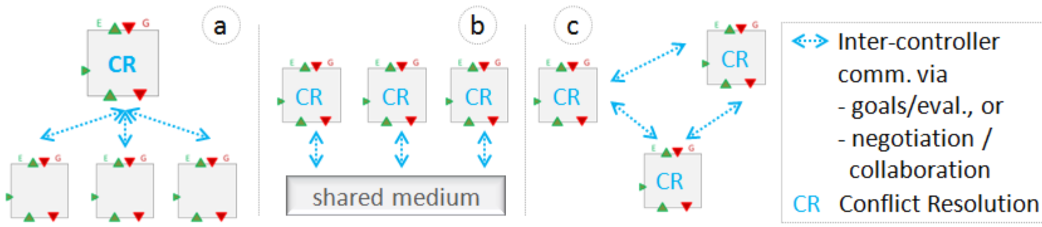

Fig. 4. Integration patterns: a) Hierarchy; b) Stigmergy; c) Collaboration

external directives into account. Also, one or both of the following requirements apply: i) global model: decisions based on larger system views are desirable, such as for overall optimisations; and/or ii) higher authority: a single actor has the power and interest to direct the resolution process towards a desired result.

Description: The Hierarchy pattern relies on a control orchestrator, or *Hierarch*, that implements the resolution logic for several conflicting controllers, or *Prosumers* [Fig. 4-a]. The Hierarch monitors the Prosumers' states, compiles a non-conflicting solution and distributes directives accordingly. This pattern can be repeated recursively at multiple levels. In general, hierarchies are organised into tree-like structures, which can be defined with respect to: *abstraction* – where higher-level nodes have a more global view of underlying hierarchies; and/or, *authority* – where higher-level nodes dictate over lower ones. Hierarchies can be formed by connecting controllers' goal/evaluation interfaces, negotiation/collaboration interfaces, or both.

Similar patterns: The Hierarchy resembles the Holarchy pattern (subsection 4.6) in its recursive tree-like structure with increasing abstractions. Yet, the Hierarchy differs from the Holarchy in its structural uniformity and lack of encapsulation, that is, of explicit isolation and access regulation to sub-trees from external entities. Also, Hierarchies are more often defined for imposing increasing authority. The negotiation-oriented Hierarchy variant (below) can be equated conceptually to a Collaboration (in case of direct agent negotiation) or Stigmergy (in case of mediated negotiation).

Concrete examples: Power management has often been defined as a Resource Constrained Scheduling Problem that can be solved via a centralised scheduler, which plays the Hierarch role. The scheduler collects prosumer states or schedules, computes a global schedule that meets global constraints and issues corresponding directives to prosumers (e.g. start, stop, shift or scale prosumption). When prosumer flexibilities are insufficient to attain the hierarchy's goals, a higher level is notified.

The Organic Smart Home [Allerding and Schmeck 2011] uses a two-layer hierarchy for managing its appliances and for its integration into a smart micro-grid. The hierarchy consists of several instances of the Observer Controller (OC) architecture: a Global OC Unit manages the smart house; and, multiple Local OC Units manage individual appliances. Local OC Units provide filtered models of monitored devices to the Global OC Unit (with increased abstraction). The Global OC Unit uses this information together with external grid signals, history and predictions to compute optimised schedules (both day-ahead and real-time) and to issue commands for the Local OC Units (using its higher authority). A working prototype implementation of Organic Smart Home has shown the validity of the approach.

[NiceGrid 2013] proposes an open-loop Hierarch to control residential storages and heat pumps in a district, allowing it to go into "island mode" daily. [Becker et al 2010] and [Pipattanasomporn et al. 2012] propose *Hierarchy*-based solutions where appliances provide profiles and user preferences to a house *Hierarch*, which computes an optimal prosumption schedule and executes it. [Schiendorfer et al. 2014] propose a recursive hierarchy with increasing abstraction where virtual agents compute schedules for lower-level prosumers. The above examples instantiate the Hierarchy pattern with both increasing abstraction and authority, and interconnect controllers via goal interfaces. Section 6 also illustrates a sample Hierarchy implementation using a simple scheduling algorithm (details can be found in [Frey et al. 2013a]).

An alternative instantiation uses micro-economy concepts to connect controllers via their negotiation/collaboration interfaces. PowerMatcher [Kok et al. 2005] uses a tree-like structure of aggregators for collecting power bids from prosumer agents. A top agent uses aggregated bids (global abstraction) to compute a price (regulation authority), which is redistributed to prosumer agents. Each agent sets its prosumption (goal) based on the difference between its bid and the price. DEZENT [Wedde et al. 2008] proposes a three-layer hierarchy (for the three grid voltage levels) for power balancing based on price negotiations over 0.5s intervals. Customer agents (prosumers) try to buy and sell power packets at various prices via a hierarchy of Balancing Group Manager (BGM) agents. The BGM at each level matches buyers and sellers so as to maximise local balancing; unmatched packets are aggregated and forwarded to the upward level, where prices become less interesting to deter lengthy negotiations.

Evaluation: The Hierarchy pattern is a “classic” organisation that allows fine global control and optimisation. Precision of control is only limited by the amount of information available to the Hierarch and by the effectiveness of its logic. This pattern also features well-known issues, like the scalability of each node with respect to the number of its child nodes. Also, the amount and detail of collected information must be limited to avoid complexifying and slowing down the Hierarch [Schiendorfer et al. 2014]. These issues are typically addressed by adding intermediate hierarchical levels (e.g. dynamically in [Steghöfer et al. 2013]). The Hierarch also introduces a single point of failure and so requires self-repair capabilities. From a business perspective, this pattern is best suited to cases involving a single authority (e.g. scheduler or price regulator).

4.4. Stigmergy pattern

Context: Conflicting controllers can neither be controlled externally nor communicate with each other directly – e.g. controllers are too numerous, join and leave the system too often, cannot trust external orders and/or cannot be trusted at individual level (only on average). There may also be a need to distribute the control logic across system resources (e.g. prosumers) rather than concentrate it into a few entities (e.g. Hierarch) – e.g. system components have limited resources and none can perform complicated control algorithms or store extensive knowledge.

Description: The Stigmergy pattern relies on control decentralisation and on choreography based on global information formed and accessible via a *shared medium* [Fig. 4-b]. All participants fulfil the same role (e.g. prosumers) and do not communicate with each other directly. Conflict resolution emerges as all participants adapt to and influence the state of the shared medium, in parallel, and typically in the same way. This can generate global overshoot and / or oscillations, unless the organisation implements some form of synchronisation between participants [Anders et al. 2012]. Stigmergy variants use different medium types, like the physical environment, a broadcast channel, or a shared information repository (Blackboard).

Similar patterns: The Stigmergy variant using a centralised Blackboard resembles a Hierarchy, yet its Blackboard contains *no* decision logic; all control belongs to participants. Hence, we considered PowerMatcher and Dezent (subsection 4.3) as Hierarchies since their negotiation mediums implement price regulation and broker decisions. Of course, the exact line between decisions and environment ‘laws’ is debatable.

Concrete examples: Observing frequency degradations to determine grid loads has been common practice for decades in supply-side power regulation [Jaleeli et al.

1992]. In this cases frequency acts as a natural shared medium. More recently, smart grid proposals based on the *Stigmergy* pattern aim to provide ultra-large scale control of individual appliances. Here, power balancing is achieved by having electric devices monitor and adapt their prosumption to the grid's frequency, which is the physical medium's indicator of the grid's state – e.g. underloaded, overloaded, or balanced. Probabilistic methods are employed to prevent prosumers from reacting in synchrony to frequency variations and cause oscillations. [Beal et al. 2012] relies on users setting flexibility preferences for their appliances via a color code, which influences their reactions to the grid state. [Mathieu and Callaway 2012] relies on the ability of thermal storages (e.g. heaters or fridges) to switch on or off rapidly at certain points in their thermodynamic cycle without comfort degradation. [Li et al. 2010] exemplifies the Blackboard-based Stigmergy variant by employing a shared "*stigspace*" as the sole means of coordination among resource and broker agents. We also illustrate this variant to control house appliances in our prototype (section 6).

Evaluation: The Stigmergy pattern is a decentralised organisation that can ensure massive scalability, device heterogeneity and runtime variability. Devices only need to be able to read and react to globally available information. It requires, in turn, an observable shared medium that can scale with the number and frequency of alterations. Also, when relying heavily on probabilistic behaviours, the system may actually *require* large-scale participation, hence excluding cases where single houses operate in "island mode" or where smart grids are developed progressively (cf. "incremental change" requirement). The additional de-synchronisation mechanisms [Beal et al. 2012] [Mathieu and Callaway 2012] may also increase overheads.

Stigmergy avoids the single point of failure of the Hierarchy (except for the centralised Blackboard variant). Since heavily relying on the cooperation of most participants, this pattern is easiest applied when participants can share compatible goals and behaviours (e.g. an ant hill may fail to emerge if each ant follows its own rules).

4.5. Collaboration pattern

Context: Centralised conflict resolution is unfeasible (like in the Stigmergy pattern), yet controllers *are* able to communicate with each other directly.

Description: The Collaboration pattern also relies on decentralised control, delegating conflict resolution to participants (e.g. prosumers) [Fig. 4-c]. These may belong to different authorities and implement decentralised negotiation protocols and decision algorithms to meet their respective goals. Coalitions can sometimes be formed.

Similar patterns: Collaborative and Stigmergic approaches are similar as both use decentralised resolution logic. The main difference is that conflicting controllers communicate directly or indirectly, respectively. In some multi-agent approaches agents collaborate by first electing a leader (higher authority) to coordinate their actions. Such organisation corresponds to a flexible self-forming Hierarchy.

Concrete examples: Collaboration is common in the multi-agent system (MAS) and game theory literatures [Osborne 1994]. It is applied to smart grids for peer-to-peer scheduling, multi-agent negotiations and various types of games. Well-known multi-agent algorithms include Contract Net Protocol [Kim et al. 2011], auction markets [Cheng 2012] or agent games [Mohsenian-Rad et al 2010]. Standardisation efforts for agent controllers in the grid, such as [McArthur et al. 2007], were based on the FIPA standard [FIPA 2014]. Recently, cooperative game theory was applied to form dynamic Customer Coalitions that can negotiate better prices for members with similar prosumption profiles [Vinyals et al. 2014], over month, day-ahead and short intervals.

Evaluation: The Collaboration pattern is a decentralised organisation that should also feature good scalability, heterogeneity, adaptability and robustness. The exact properties will highly depend on the concrete design and implementation. Complicated agent strategies requiring 'intelligence' can complexify prosumers and require more resources. In the absence of a higher-level controller (cf. Hierarchy), or of an expected common behaviour (cf. Stigmergy), each prosumer can more readily pursue its own goals. Hence, Collaboration is suitable for cases that involve several competing authorities. A minimum number of participants is generally not required.

4.6. Holonic integration meta-pattern

Context: The control system must integrate several heterogeneous organisations, which must operate simultaneously at different scales – i.e. scopes and time. The system includes a large number of resources and controllers based various abstractions, so that none of them can know and communicate with all the others. They must be able to adapt to local changes while limiting impact on the overall system. Multiple authorities impose various restrictions on the visibility and access to system controllers and resources – e.g. home owners may only trade electricity with a limited set of district electricity managers; this restricts access to home controllers to those district managers only. Similarly, devices belonging to home owners should be inaccessible to district managers (unless a specific contract is signed beforehand, such as for fridges).

Description: *Holonic* systems, or *holarchies*, can address these concerns in grid control systems [Fig. 5] [Negeri et al. 2013]. A *holon* is a semi-autonomous entity: a *whole* with respect to its parts, which it controls to achieve its goals; and a mere *part* in an enclosing supra-structure, whose directives it follows to contribute to higher goals [Koestler 1967]. This enables the construction of 'hierarchical' organisations [Simon 1969] that are readily applicable to smart grids. Here, each grid level (e.g. a house grid) represents both a self-reliant *whole* for the hierarchical level below (e.g. house appliances), and a dependent *part* for the hierarchical level above (e.g. district grid) [Frey et al. 2013b]. Each Holon encapsulates and controls access to its contents for privacy and heterogeneity reasons. Internally, its *organisation* follows one of the patterns presented above. Externally, it represents a mere *role* in a higher-level organisation.

Since the patterns of a holon's supra- and intra-organisations have been presented already, the only new architectural element needed is one that provides the *in-between logic* connecting the holon's external and internal organisations – i.e. implementing the holon's double nature. We define this as a *goal management membrane*, encapsulating the holon's internal organisation and connecting it to external entities [Fig. 5]. It features the same architecture as any generic controller [Fig. 2-c], except for its encapsulating nature, which means that the generic goal-oriented design is repeated twice: once for the holon's goal inputs (input membrane); and once for the holon's goal outputs (output membrane). Conceptually, the controller goal-oriented interfaces and resolution process [Fig. 2] map onto the (input) membrane as follows [Fig. 5]:

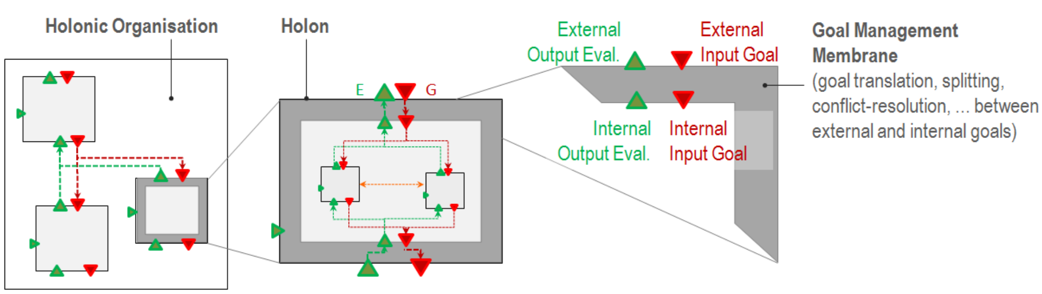

Fig. 5. Holonic control integration meta-pattern

— Input goals map to the holon's *external input goals*, which represent directives or goals from its supra-organisation and/or from the holon itself;
— The holon's *Internal input goals* are coherent non-conflictual goals that are set for its internal organisation controllers, via connections to their input goals; (the organisation's resolution process transforms these into *internal output goals*, which the output membrane transforms into *external output goals* – not shown);
— The input membrane's *inward goal resolution* process transforms the holon's external input goals into its Internal Input Goals, via application-specific goal operations: translation, splitting and conflict-resolution;
— Output evaluation maps to the holon's *external output evaluation*, which provides feedback on goal achievements to its supra-organisation;
— The holon's *internal output evaluation* collects evaluations from its internal organisation controllers, by being connected to their output evaluation interfaces;
— The input membrane's *outward evaluation process* transforms internal output evaluations into an external output evaluation (not discussed here);
— The membrane also features a negotiation/collaboration interface (not shown).

The goal management membrane is a logical component: it can be designed as an actual layer "around" the holon; distributed across the holon's parts; or both. The concrete solution is application-specific; section 6 provides such example.

Similar patterns: Numerous smart grid solutions implement hierarchical designs that resemble holarchies [Kok et al. 2005; Wedde et al. 2008]. Yet they lack the encapsulation aspect and typically use self-similar organisations across the grid.

Concrete Examples: Several contributions have started embracing the holonic paradigm to design decentralised controllers for smart-grids e.g., [Negeri et al. 2013; Ounnar et al. 2013; Thomas and Devanathan 2011; Lässig et al. 2011]. [Negeri et al. 2013] propose a holonic control architecture focusing on a holon's internal modules and overlooking its integration at recursive levels. [Ounnar et al. 2013] concentrate on a multi-criteria decision aid method, which represents the application-specific implementation of optimisation decisions. [Thomas and Devanathan 2011] support the idea of holonic smart grids by proposing a multi-cluster design, with three self-contained layers: individual power producers, intermediary mini-grids and a traditional top-level grid. Layers feature self-similar designs based on the traditional grid control architecture for voltage and frequency regulation. This is one (non-heterogeneous) design instance of a holonic smart grid and only considers electricity management. [Lässig et al. 2011] propose a solution for price negotiations in the smart grid based on 'hierarchical' (holonic) markets. Each market contains a set of rules and agents, which can be organised internally as other markets, with their own rules and agents. Only agents belonging to a market can negotiate and regulate the price of that market. A special-purpose 'head' agent ensures the link between supra-markets and internal markets, by playing double negotiation roles. This agent represents the goal management membrane in our generic architecture. This approach provides an example of holonic design where controllers (agents) interconnect via their negotiation interfaces. The generic architecture proposed here can help integrate such approaches with complementary solutions managing other administrative concerns. The Organic Smart Home (cf. subsection 4.3) can also be integrated as a holon into a smart micro-grid, since it can react to external signals (as illustrated in the sample implementation in section 6). The Global OC Unit performs the holon's membrane functions in this case.

Evaluation: The Holonic meta-pattern enables organisations of various types to recursively encapsulate each other, hence addressing the "heterogeneity" and "multi-scale" requirements. It helps system "scalability" and complexity management by exposing to each controller only those controllers and resources that belong to the same organisation, while abstracting away their internal details. An application-specific goal management membrane ensures the translation of protocols and abstraction levels between supra- and intra-organisations. It can also also enforce access-control as imposed by business-related restrictions among authorities. The holonic structure separates the grid into loosely-coupled sub-grids hence favouring reliability and robustness – e.g. failures can be isolated from the whole and handled locally [Ramchurn et al. 2012]; or, micro-grids can switch to island mode should the main grid fail [Thomas and Devanathan 2011]. Reactivity can also improve as changes can be addressed locally and contained when possible to minimise impact on the overall system.

A holonic structure may also help with self-protection and privacy concerns. Here, the state information a holon provides to its supra-organisation can depend on the business context (e.g. collaboration or competition) and may even change during runtime (e.g. threat detection). For instance, if privacy is an important concern (e.g. between house and district managers) the holon may simply respond to orders (e.g. to reduce or increment consumption) by accepting or declining them, with no further justification that may give away its internal state. In a more trusted environment, a holon may provide minimal state information, in an aggregate form (like exemplified in section 6). Finally, in a trusted context, like within a household, devices may freely share information with the house manager, via secured protocols. While highly relevant, these aspects are beyond the paper's scope.

5. RELATED WORK

5.1. Holonic designs for complex systems

Since [Simon 1962] has identified *hierarchical architectures* as key enablers of complex systems and [Koestler 1967] coined the terms 'holon' and 'holarchy', holonic approaches have received increasing interest from numerous research communities. In the engineering field, holonic solutions have also been shown to be essential for self-adapting and self-organising systems [Valckenaers et al. 2008] such as traffic control [Fischer 1999], manufacturing [Ulieru et al. 2002] and smart grids (subsection 4.6).

As a common basis for many application domains, *holonic multi-agent systems (HMAS)* have been proposed to manage complexity and bridge the micro-macro gap in MAS theories [Fischer 1999; Rodriguez 2005; Cossentino et al. 2010]. Closest to our research is work by S. Rodriguez et al [Rodriguez 2005; Rodriguez et al. 2006], including: a holonic framework for structuring open agent-based applications; and agent *capacities* – representing agent know-how – for enabling agent organisations to achieve goals. Their framework proposes a special-purpose agent *organisation* for allowing a holon's integration into its super-holon. Integration is achieved via a dedicated agent fulfilling a *Head* role (similar to [Lässig et al. 2011]), which is one possible implementation of to the goal management membrane we propose. Yet, we do not impose that this layer be an autonomous agent with its own goals. It can also be a procedural component performing translation and conflict-resolution operations on incoming goals. Also, we equate an agent's 'capacities' with the goals it can pursue and hence expose them explicitly in the holon's interfaces. Overall, we employ reusable organisation patterns [Zambonelli et al. 2003] to help design the domain-specific system part (i.e. smart micro-grid) and rely on more "classic" SE constructs, such as component membranes and pattern composition [Garlan and Shaw 1994], to conceive the reusable holarchy-related infrastructure. This approach also fulfils the *closure* requirement identified in the *operator hierarchy* theory for complex systems [op Akkerhuis 2010].

Hierarchical planning [Nolle et al. 2002] is another useful application of the holonic approach, enabling designers (or planners) to concentrate on major decisions first (higher-level holons) and be able to elaborate the details of each one separately, and later on (lower-level holons). Finally, *Systems of Systems (SOS)* and *Federations of Systems (FOS)* [Sage and Cuppan 2001] feature certain similarities with Holonic designs, including control decentralisation and support for multi-authority, multi-goal, heterogeneity, scalability and local adaptability. As the main difference, in SOS and FOS, the overall behaviour is not generally controlled directly, but rather emerges from the behaviours of composing systems. *No* higher-level component contains and controls these sub-systems. Holonic systems can adopt this approach, but can also exert higher-level control over sub-systems, via hierarchical organisations, if permission were granted.

5.2. Generic smart grid architectures and platforms

Considerable effort has been invested in developing standard smart grid architectures and platforms, via various organisations in Europe and the United States, e.g. the National Institute of Standards and Technology (NIST), OASIS Energy Market Information Exchange (EMIX) and Energy Interoperation technical committees, the Open Gateway Energy Management Alliance (OGEMA) and the CEN-CENELEC-ETSI Smart Grid Coordination Group [SGAM 2012].

Notably, the *Smart Grid Architecture Model* [SGAM 2012] aims to provide a standardised model that is sufficiently abstract to represent all major stakeholders and match their existing solutions; supports a variety of solutions that conform to different stakeholder requirements – from fully centralised to decentralised models – and allows their integration; and provides a methodology for helping users apply the generic architecture to various use cases. The SGAM framework is organised along three dimensions. The first two define a Power Plan with *domains* representing stages in the energy conversion chain – i.e. bulk generation, transmission, distribution, distributed energy resources (DER) and customer premises – and *zones* defining hierarchical levels of power management – i.e. from physical equipment, relays and plant supervision, to energy management systems, enterprise-related processes and market operations. The third dimension identifies key *Interoperability Layers* representing various stakeholders – i.e. Component, Communication, Information, Function and Business layers. A number of cross-cutting concerns are also identified, including security and privacy, discovery and configuration, performance, reliability and scalability.

The generic architecture we propose is compatible with the SGAM framework. It focuses on a control viewpoint and offers a goal-oriented pattern-based approach for facilitating controller design and interoperability. It can be positioned in the Power Plan as covering the micro-grid zones and domains (customer, DER and distribution). It can provide useful generic elements to most Interoperability Layers, starting with the definition of objectives (goals) in the Business Layer and supporting their progressive transformation into more technical goals in the Function Layer. It also addresses aspects of the Information and Communication Layers with respect to information exchanges via goals, evaluations and negotiations via generic patterns. Finally, the holonic meta-pattern provides conceptual and architectural elements for integrating stakeholder-specific solutions, which SGAM aims to support.

The *Open Gateway Energy Management Alliance (OGEMA)*[1] aims to offer an open-source standard and reference implementation for a software platform that allows energy management applications from various sources to execute on a common operating system, or gateway, in order to operate and connect customer facilities to the grid. In the Organic Smart Grid [Allerding and Schmeck 2011] the Hardware Abstraction Layer provides a similar gateway. Also, the iCASA[2] project offers an OSGi-based gateway, which can be extended for power management. OGEMA software is also based on

[1] Open Gateway Energy Management Alliance (OGEMA) – http://www.ogema.org
[2] iCASA Digital Home Execution Platform and Simulation Tool – http://adeleresearchgroup.github.io/iCasa

OSGi, which we use as well in the micro-grid simulator and control prototype (section 6). Such technology-independent layers can be equated in the proposed architecture with a controller performing translation operations.

6. PROOF-OF-CONCEPT DESIGN AND IMPLEMENTATION

6.1. Simulator

We developed a proof-of-concept smart micro-grid simulator – MisTiGriD[3]– for two purposes. Firstly, it illustrates the proposed holonic architecture via a concrete implementation that integrates two organisation types. It supports experiments in several scenarios showcasing the integration capabilities of the approach (details in [Frey et al. 2013b] and the project's web page). Secondly, it offers a reusable platform for exploring various combinations of control organisations and algorithms.

At the ***House*** level, power management implements the *Blackboard variant of the Stigmergy pattern*. The Blackboard is a discrete schedule shared among controllers. A House Controller (or Power Manager) specifies its prosumption goal G_P^H (cf. subsection 3.2). Device Controllers (or Prosumers) publish their prosumption profiles – i.e. power value and duration –, and preferences – *Flexible* or *Non-Flexible*. For instance, ***Smart Lamps*** can consume at full or reduced power – 100W or 30W, respectively. When *Flexible* – e.g. the user accepts reduced light intensity for meeting a power goal – a smart lamp can lower its consumption in reaction to high loads observed on the Blackboard. When a *Non-Flexible* lamp is switched on it prioritises quality of service and always consumes at full power. Similarly, *Flexible* ***Smart Heaters*** can reduce their target temperature during load peaks. A simple de-synchronisation mechanism prevents *Flexible* appliances from reacting simultaneously to load variations, in order to avoid overshoots and oscillations as discussed in subsection 4.4.

At the ***District*** level, power management implements the *Hierarchy pattern*. Here, House-level Controllers represent District-level Prosumers. They aggregate the content of their local Blackboard into a single house prosumption profile and flexibility, and publish the result to a district schedule. A District Power Manager (Hierarch) monitors this schedule to determine the district state and sends direct orders to house controllers (e.g. 'reduce' prosumption). Controllers that accept such orders (flexible) reduce their prosumption goal (e.g. P_{Hmax} from 800W to 200W). This, in turn, affects the house's internal power management.

This sample implementation integrates a Stigmergy organisation (house holon) within a Hierarchy (district holon). The goal management membrane functions are included in the House Controller. They ensure the bi-directional transition between the house as a *whole* – with its internal Stigmergy organisation – and as a *part* – playing a Prosumer role in the district Hierarchy. The House Controller aggregates device information to provide an abstracted prosumer state to its supra-organisation (district). In the other direction, the House Controller receives orders from its supra-organisation (as if it were a single prosumer), resolves conflicts between these and the house's 'comfort' goal (depending on user priorities and overall device flexibilities), and translates the result into a power goal for its internal organisation. Alternative pattern combinations are also possible (e.g. house hierarchy and district collaboration).

6.2. Scenarios and Results

The purpose of the presented experiments is to *illustrate* how control solutions can be recursively integrated by having an entire organisation (whole) represent one part in

[3]MisTiGriD project: http://perso.telecom-paristech.fr/∼sfrey/

a higher-level organisation. Results indicate the feasibility of this (holonic) integration approach. They are *not* meant to validate any of the control solutions presented – either individually or composed – *nor* to evaluate their performance. Nor are they meant to validate the proposed architecture, since this cannot be achieved by showing that one implementation is possible. This would require numerous implementations and long-term testing in various contexts. When such extensive resources are unavailable, a design proposal can only be defended via solid argumentation, and concrete examples from literature, which we hope to have achieved in the previous sections.

The House Grid Scenario. In this scenario, we regard a house composed of 6 rooms, containing 6 heaters and 5 lamps (simulated as described above). Fig. 6 left shows the House Controller's goal, the sum of lamp consumptions, and the total house consumption (including all heaters and lamps, averaged over a short sliding window for better visibility). This scenario consists of three consecutive phases.

Phase 1: The House Controller pursues a constant maximum power consumption goal of 800W. Heaters pursue constant temperature goals in their rooms. In the middle of Phase 1 (about 100s), residents switch on several *non-flexible* lamps. The flexible heaters compensate for the load increase so the house consumption goal is still met.

Phase 2: In this phase, only heaters are consuming. The House Controller lowers the maximum power goal from 800W to 200W in reaction to a district signal (cf. next scenario). The 6 heaters consuming 200W each manage to schedule themselves stigmergically so that their total consumption meets the goal with limited error (averaged). All returns to normal when the house goal is reset to 800W.

Phase 3: This phase features an *inflexible* lamp consumption, as in Phase 1. Also, a simulated cold wave causes more frequent heating cycles. Since the user does not allow heaters to reduce their temperature below a minimum threshold the total house consumption increases abruptly. This is consistent with the user's preferences.

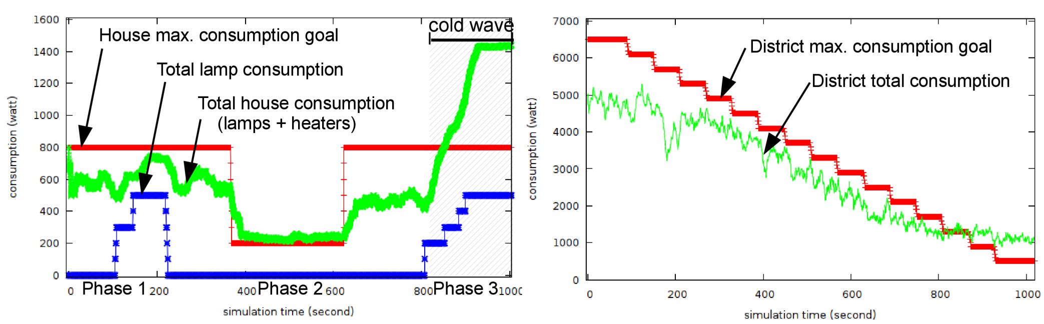

Fig. 6. House (left) and district (right) scenarios.

The District Grid Scenario. This scenario could be similar to the one run at the house level, due to the micro-grid's holonic architecture. In this case, flexible houses would compensate for high-consuming neighbours (phase 1 above), global loads would adapt to changes in the district's power goal (phase 2), and excessive consumption would occur when a majority of houses would go into comfort mode (phase 3). To diversify the examples, Fig. 6 right shows another type of scenario, where the district administrator tests the system's adaptation range. Starting from a district goal with high maximum consumption, the administrator decreases this value step-by-step. Results show how houses can adapt their consumptions in response to such district goal reductions, until the limit becomes too small for these to follow – i.e. from 800s onwards. The simulation was run with 8 houses similar to the one described above.

A prompt adaptation of the district grid is a good result for a district administrator. It indicates that the district may feature specific constants, such as minimum and maximum prosumption plateaux within which the district's load can adapt. This will help implement power management policies with respect to the higher grid levels, like region and country. It also means that house power management, although not under the direct control of the district manager, can contribute to the district's goals. This was one of the main purposes of the integration architecture proposed.

7. CONCLUSIONS & FUTURE WORK

This paper presented **a generic holonic architecture for smart-grid control, enabling the recursive integration of heterogeneous control patterns implemented at different grid scales**. This contribution aimed to address a number of key smart grid requirements, like multiple authorities, conflicting objectives, design heterogeneity and scalability. We identified three main integration patterns that seem to characterise state-of-the-art control solutions in the domain; and showed how different instances of these can be integrated via a holonic architecture, which enables exchanges of heterogeneous state information and goals, as well as conflict resolution among different holons. We experimented with a proof-of-concept implementation of two of the patterns, integrated and run on a distributed smart grid simulator. Results illustrated the ability of the generic architecture to support control solutions that:

— can achieve multiple micro- and macro-objectives, by integrating heterogeneous controllers into organisations operating at two grid scales – house and district;
— enable administrators to dynamically tune the priorities of their objectives, in order to re-balance results at the micro and macro scales;
— enable multiple administrators to pursue conflicting objectives, at various scales, while preserving macro-level coherence where key economic interests lie.

This contribution aims to help smart grid designers to understand, reason about, and architect viable solutions for control systems that must meet challenging requirements – i.e. multi-authority, multi-scale, multi-goal, heterogeneity and scalability – while not disrupting existing systems and being able to integrate new ones – i.e. allowing for incremental change. Such conceptual and architectural support is increasingly valuable and even necessary in a context where numerous actors aim to introduce ICT-controlled cyber-physical systems into living environments, including smart grids but also smart houses and traffic control, intertwined into smart cities and larger-scale socio-technical systems that must preserve their coherence.

This work opens wide perspectives for future research. In the short term, we will study the behaviour of the approach via the simulator in more control scenarios, at larger scales and with new prosumer types (e.g. local producers and storage). This will include experiments with alternative pattern combinations and algorithms. Over the longer term, we intend to address additional controller integration concerns, particularly time-related. Further effort will refine and categorise the generic controller's goal and evaluation interfaces and its control operations. We aim to identify generic mechanisms for enabling holonic organisations to self-adapt to both top-down goal changes and context events (as illustrated in the experiments) and also to bottom-up evaluation feedback, by reconsidering control goals, hot-swapping between internal controllers or patterns, and retuning their configurations. Finally, we will study the applicability and impact of the proposed holonic architecture on other domains.

REFERENCES

F. Allerding, B. Becker, and H. Schmeck. 2011. Decentralised Energy Management for Smart Homes. In *Organic Computing A Paradigm Shift for Complex Systems*, C. Müller-Schloer, H. Schmeck, and T. Ungerer (Eds.). Autonomic Systems, Vol. 1. Springer Basel, 605–607.

F. Allerding and H. Schmeck. 2011. Organic Smart Home: Architecture for Energy Management in Intelligent Buildings. In *Proceedings of the 2011 Workshop on Organic Computing (OC '11)*. 67–76.

G. Anders, C. Hinrichs, F. Siefert, P. Behrmann, W. Reif, and M. Sonnenschein. 2012. On the Influence of Inter-Agent Variation on Multi-Agent Algorithms Solving a Dynamic Task Allocation Problem under Uncertainty. In *Self-Adaptive and Self-Organizing Systems (IEEE SASO)*. 29–38.

J. Beal, J. Berliner, and K. Hunter. 2012. Fast Precise Distributed Control for Energy Demand Management. In *Self-Adaptive and Self-Organizing Systems (IEEE SASO)*.

B. Becker et al. 2010. Decentralized Energy-Management to Control Smart-Home Architectures. In *ARCS (Lecture Notes in Computer Science)*, Vol. 5974. 150–161.

G. Cabri, L. Ferrari, and F. Zambonelli. 2004. Role-Based Approaches for Engineering Interactions in Large-Scale Multi-agent Systems. In *Software Engineering for Multi-Agent Systems II*, C. Lucena, A. Garcia, A. Romanovsky, J. Castro, and P. Alencar (Eds.). LNCS, Vol. 2940. Springer Berlin Heidelberg, 243–263.

D. Callaway and I. Hiskens. 2011. Achieving Controllability of Electric Loads. *Proc. of the IEEE* 99, 1 (2011).

Y. Cheng. 2012. Architecture and Principles of Smart Grids for Distributed Power Generation and Demand Side Management. In *Int. Conf. on Smart Cities and Green ICT Systems (SMARTGREENS)*.

M. Cossentino, S. Galland, N. Gaud, V. Hilaire, and A. Koukam. 2010. An organisational approach to engineer emergence within holarchies. *Int. Journ. of Agent Oriented Software Engineering* 4 (2010).

FIPA. 2014. http://www.fipa.org/. (2014). [online, accessed July-28th-2014].

K. Fischer. 1999. Holonic Multiagent Systems Theory and Applications . In *Progress in Artificial Intelligence*, Pedro Barahona and JosJ. Alferes (Eds.). LNCS, Vol. 1695. Springer Berlin Heidelberg, 34–48.

S. Frey, A. Diaconescu, and I. Demeure. 2012. Architectural Integration Patterns for Autonomic Management Systems. *IEEE Int. Conf. on the Engineering of Autonomic & Autonomous Systems (EASe)* (2012).

S. Frey, A. Diaconescu, D. Menga, and I. Demeure. 2013a. A Holonic Control Architecture for a Heterogeneous Multi-Objective Micro Smart Grid. *Self-Adaptive and Self-Organizing Systems (SASO)* (2013).

S. Frey, A. Diaconescu, D. Menga, and I. Demeure. 2013b. Towards a reference model for multi-goal, highly-distributed and dynamic autonomic systems. *Int. Conf. on Autonomic Computing (ICAC)* (2013).

D. Garlan and M. Shaw. 1994. *An Introduction to Software Architecture*. Technical Report. Pittsburgh, USA.

N. Jaleeli, L. S. VanSlyck, D.N. Ewart, L.H. Fink, and AG. Hoffmann. 1992. Understanding automatic generation control. *Power Systems, IEEE Transactions on* 7, 3 (Aug 1992), 1106–1122.

J. O. Kephart and D. M. Chess. 2003. The Vision of Autonomic Computing. *Computer* 36 (Jan 2003), 41–50.

H.-M. Kim, W. Wei, and T. Kinoshita. 2011. A New Modified CNP for Autonomous Microgrid Operation Based on Multiagent System. *Journal of Electrical Engineering and Technology 6-1* (2011).

A. Koestler. 1967. The Ghost in the Machine. (1967).

J. K. Kok, C. J. Warmer, and I. G. Kamphuis. 2005. PowerMatcher: Multiagent Control in the Electricity Infrastructure. In *Proc. of the 4th Int. Conf. on Autonomous Agents and Multiagent Systems (AAMAS'05)*.

J. Lässig, B. Satzger, and O. Kramer. 2011. Hierarchically Structured Energy Markets as Novel Smart Grid Control Approach. In *KI 2011: Advances in Artificial Intelligence*. LNCS, Vol. 7006.

J. Li, G. Poulton, and G. James. 2010. Coordination of Distributed Energy Resource Agents. *Applied Artificial Intelligence* 24, 5 (2010), 351–380. DOI:http://dx.doi.org/10.1080/08839514.2010.481483

C. Marnay and G. Venkataramanan. 2006. Microgrids in the evolving electricity generation and delivery infrastructure. In *Power Engineering Society General Meeting, IEEE*.

J. L. Mathieu and D. S. Callaway. 2012. State Estimation and Control of Heterogeneous Thermostatically Controlled Loads for Load Following. *2014 47th Int. Conf. on System Sciences* 0 (2012), 2002–2011.

S.D.J. McArthur, E.M. Davidson, V.M. Catterson, A.L. Dimeas, N.D. Hatziargyriou, F. Ponci, and T. Funabashi. 2007. Multi-Agent Systems for Power Engineering Applications 2014;Part I: Concepts, Approaches, and Technical Challenges. *Power Systems, IEEE Transactions on* 22, 4 (2007), 1743–1752.

A.-H. Mohsenian-Rad et al. 2010. Autonomous Demand-Side Management Based on Game-Theoretic Energy Consumption Scheduling for the Future Smart Grid. *IEEE Tr. Smart Grid* 1, 3 (2010), 320–331.

E. Negeri, N. Baken, and M. Popov. 2013. Holonic Architecture of the Smart Grid. *Scientific Research online journal, special issue on 'Smart Grid and Renewable Energy'* (2013).

M. Negrete-Pincetic and S. Meyn. 2011. Intelligence by design for the entropic grid. In *Power and Energy Society General Meeting, IEEE*.

NiceGrid. 2013. http://www.nicegrid.fr/. (2013). [online, accessed July-28th-2014].

L. Nolle, K.C.P. Wong, and A.A. Hopgood. 2002. DARBS: A Distributed Blackboard System. In *Research and Development in Intelligent Systems XVIII*, Max Bramer, Frans Coenen, and Alun Preece (Eds.). Springer London, 161–170. DOI: http://dx.doi.org/10.1007/978-1-4471-0119-2_13

G. Jagers op Akkerhuis. 2010. *The operator hierarchy, a chain of closures linking matter, life and artificial intelligence*. Ph.D. Dissertation.

M. J. Osborne. 1994. *A Course in Game Theory*. MIT Press, USA.

F. Ounnar, A. Naamane, P. Pujo, and N. M'Sirdi. 2013. Intelligent Control of Renewable Holonic Energy Systems. *Energy Procedia* 42, 0 (2013), 465 – 472.

M. Pipattanasomporn, M. Kuzlu, and S. Rahman. 2012. An Algorithm for Intelligent Home Energy Management and Demand Response Analysis. *IEEE Tr. Smart Grid 3-4* (2012).

S. D. Ramchurn, P. Vytelingum, A. Rogers, and N. R. Jennings. 2012. Putting the 'smarts' into the smart grid: a grand challenge for artificial intelligence. *Comm. ACM 55-4* (2012).

S. Rodriguez, N. Gaud, V. Hilaire, S. Galland, and A. Koukam. 2006. An Analysis and Design Concept for Self-organization in Holonic Multi-agent Systems. In *Engineering Self-Organising Systems, 4th International Workshop - Revised and Invited Papers (LNCS)*, Vol. 4335. Springer, 15–27.

S. A. Rodriguez. 2005. *From analysis to design of holonic multi-agent systems: A framework, methodological guidelines and applications*. Ph.D. Dissertation. Université de Technologie de Belfort-Montbéliard and Université de Franche-Compté.

A. Sage and C. Cuppan. 2001. On the Systems Engineering and Management of Systems of Systems and Federations of Systems. *Information-Knowledge-Systems Management Journal. 2(4)* (2001).

A. Schiendorfer, J-P. Steghöfer, and W. Reif. 2014. Synthesis and Abstraction of Constraint Models for Hierarchical Resource Allocation Problems. *Proc. of the 6th International Conference on Agents and Artificial Intelligence (ICAART)* 2 (2014).

H. Schmeck and L. Karg. 2010. E-Energy - Paving the Way for an Internet of Energy (Auf dem Weg zum Internet der Energie). *it - Information Technology* 52, 2 (2010), 55–57.

CEN-CENELEC-ETSI Smart Grid Coordination Group SGAM. 2012. Smart Grid Reference Architecture. http://ec.europa.eu/energy/gas_electricity/smartgrids/doc/xpert_group1_reference_architecture.pdf. (2012). [online, accessed July-28th-2014].

H. A. Simon. 1962. The architecture of complexity. In *Proc. of the American Philosophical Society*.

H. A. Simon. 1969. The Sciences Of The Artificial. MIT Press, Cambridge, Mass, 1st edition. (1969).

J-P. Steghöfer, P. Behrmann, G. Anders, F. Siefert, and W. Reif. 2013. HiSPADA: Self-Organising Hierarchies for Large-Scale Multi-Agent Systems. *Proc. Int. Cnf. on Autonomic and Autonomous Systems* (2013).

A.K. Thomas and R. Devanathan. 2011. Energy efficient dynamic multi-level hierarchical clustering technique for network discovery in wireless sensor networks. In *Ultra Modern Telecommunications and Control Systems and Workshops (ICUMT), 2011 3rd International Congress on*. 1–5.

A. Toffler. 1980. *The Third Wave*. Black Rose Books.

S. Tomforde and C. Müller-Schloer. 2014. Incremental Design of Adaptive Systems. *J. Ambient Intell. Smart Environ.* 6, 2 (March 2014), 179–198. http://dl.acm.org/citation.cfm?id=2593416.2593421

M. Ulieru, R.W. Brennan, and S. Walker. 2002. The holonic enterprise: a model for Internet-enabled global manufacturing supply chain and workflow management. (2002). Integrated Manufacturing Systems, Vol. 13 Iss: 8, pp.538 - 550.

P. Valckenaers, H. Van Brussel, and T. Holvoet. 2008. Fundamentals of Holonic Systems and Their Implications for Self-Adaptive and Self-Organizing Systems. In *SASO Workshops*. IEEE CS, 168–173.

M. Vinyals, V. Robu, A. Rogers, and N.R. Jennings. 2014. Prediction-of-use games: a cooperative game theory approach to sustainable energy tariffs. In *Proc. 13th Int. Conf. on Autonomous Agents and Multi-Agent Systems*. 829–836.

H.F. Wedde, S. Lehnhoff, C. Rehtanz, and O. Krause. 2008. Bottom-Up Self-Organization of Unpredictable Demand and Supply under Decentralized Power Management. In *Self-Adaptive and Self-Organizing Systems, 2008. SASO '08. Second IEEE International Conference on*. 74–83.

D. Weyns, R. Haesevoets, A. Helleboogh, T. Holvoet, and W. Joosen. 2010. The MACODO middleware for context-driven dynamic agent organizations. *ACM Trans. on Autonomous & Adaptive Systems* 5, Article 3 (February 2010), 28 pages. Issue 1.

F. Zambonelli, N.R. Jennings, and M. Wooldridge. 2003. Developing Multiagent Systems: The Gaia Methodology. *ACM Trans. on Software Engineering & Methodology* 12, 3 (July 2003), 317–370.

Received February 2014; revised August 2014 and October 2014; accepted December 2014