

Article

Distributed Semantic Architecture for Smart Grids

Yoseba K. Peña ^{1,*}, Juan Carlos Nieves ², Angelina Espinoza ³, Cruz E. Borges ¹, Aitor Peña ¹ and Mariano Ortega ⁴

¹ Deusto Institute of Technology, University of Deusto, Avenida de las Universidades 24, Vizcaya, 48007 Bilbao, Spain; E-Mails: cruz.borges@deusto.es (C.E.B.); aitor.pena@deusto.es (A.P.)

² Department of Computing Sciences, Umeå University, Umeå SE-901 87, Sweden; E-Mail: jcnieves@cs.umu.se

³ Electrical Engineering Department, Metropolitan Autonomous University, Delegación Tlalpan, Mexico City 09340, Mexico; E-Mail: angelina.espinoza.limon@gmail.com

⁴ Indra Systems Inc., 35 Avenida De Bruselas, Alcobendas, Madrid 28108, Spain; E-Mail: mortega@indra.es

* Author to whom correspondence should be addressed; E-Mail: yoseba.pena@deusto.es; Tel.: +34-943-326-600 (ext. 5607).

Received: 9 October 2012; in revised form: 1 November 2012 / Accepted: 14 November 2012 /

Published: 20 November 2012

Abstract: The smart grid revolution demands a huge effort in redesigning and enhancing current power networks, as well as integrating emerging scenarios such as distributed generation, renewable energies or the electric vehicle. This novel situation will cause a huge flood of data that can only be handled, processed and exploited in real-time with the help of cutting-edge ICT (Information and Communication Technologies). We present here a new architecture that, contrary to the previous centralised and static model, distributes the intelligence all over the grid by means of individual intelligent nodes controlling a number of electric assets. The nodes own a profile of the standard smart grid ontology stored in the knowledge base with the inferred information about their environment in RDF triples. Since the system does not have a central registry or a service directory, the connectivity emerges from the view of the world semantically encoded by each individual intelligent node (*i.e.*, profile + inferred information). We have described a use-case both with and without real-time requirements to illustrate and validate this novel approach.

Keywords: smart grids; architecture; semantic; distributed control

1. Introduction

A hundred years from their initial development, deployment and spreading, power networks must face nowadays a new revolution. With the impending peak of petrol production, the continuous increase in the demand worldwide as well as the need to reduce greenhouse gases, efficiency at all levels becomes the key to the optimal and sustainable use of the resources. Yet energy efficiency is not the only goal to be achieved: a number of new models, methods and possibilities are already imposing different requirements that the classical power networks just cannot fulfil. The paradigm *smart grid* (term coined in contrast to the *old* allegedly non-intelligent ones) refers to the future electrical networks in which these upcoming scenarios and ICT will go hand in hand to achieve this excellence.

Historically, the electrical grid evolved to nation-wide scale controlled centrally by the government. The liberalization of electric markets broke this tight shape, opening the way to transnational energy business and corporations. However, the inheritance of the legacy infrastructure remained: diverse network specifications, models, protocols, *de facto* standards, architecture and markets... preventing, hence, the design and development of an unique smart grid approach [1].

As the supply is constantly struggling to meet the demand, there has been a notable research effort around smart grid as the keystone to the efficient energy management and consumption. The focus has been laid on certain aspects of the problem, usually either the logical *or* the physical part, such as demand response [2], voltage balancing [3], distributed management [4–6], integration of renewables [7,8], failure recovery [9], or diverse communication issues [10–13]. We present here the ENERGOS project [14], which fully addresses the optimal deployment and management of smart grids in both levels, *i.e.*, logical and physical. Though ENERGOS is not the first effort to employ the so-called *Semantic Web Technologies*, it is the primer to use them thoroughly in its architecture, fully exploiting the advantage that this model offers.

Since the smart grid concept involves a radical role change of the system-wide participants (e.g., passive members such as clients will generate *and* simultaneously consume information becoming in this way *prosumers*), there is a pressing issue concerning handling, storing, and extracting knowledge (*i.e.*, taking advantage) of the data created, all in real-time and flowing bidirectionally: consumption measures of millions of clients, price signals, grid events, transport network operation orders, alarms, and so on, including data of integrated brand-new technologies such as the electric vehicle (EV).

The semantic web was designed to dam, cope and finally exploit such data floods, but it is not yet mature enough and a number of open issues remain to be addressed. For instance, assuming that the majority of the data generators in the system were semantic nodes (*i.e.*, *information* generators), it is not yet clear which existing technologies may allow the best way to store and query that information (which may require real-time handling). There exist some initiatives [15] dealing with managing big amounts of semantic data (streams), but this framework has not been tested so far under the special premises of the electric domain and neither has it been combined with other middle-ware communication solutions as we do here. Moreover, having distributed information stores implies that a query must be divided, sent to the nodes that may have the pertinent information, and rejoined when the results arrive (*i.e.*, the so-called *query federation* problem [16]). In summary, there are a number of issues unaddressed concerning intelligence distribution and semantic information processing in the smart grid domain.

Against this background, we present here an advance in the state-of-the-art in two aspects. First, we describe a communication architecture to support the processing of semantic information in quasi-real-time. Second, we combine the multi-agent paradigm with semantic technologies to enable agents discovering logically or physically connected peers to collaborate with them, achieving in this way an effective intelligence distribution. Both the logical infrastructure to enable real-time management of the grid and the distribution of intelligence, operations and, therefore, management, are the crucial requirements for the successful transition from current power networks into smart grids. We illustrate these contributions with a use-case that describes how this collaboration takes place in a real-time scenario and in a non-real-time one. As we will see, the proposed architecture allows a much more dynamic handling of the network. This fact avoids complicated reconfiguration and update process (unlike in the SCADA systems nowadays where all communication and applications must be hard-coded, making updates and changes very difficult and costly). The remainder of the paper is divided as follows. Section 2 presents the smart grid scenario. Section 3 discusses related work. Section 4 details the architecture designed and the compounding parts. Section 5 introduces the use-cases worked out to evaluate our approach and, finally, Section 6 concludes and draws the avenues of future work.

2. Power Networks Nowadays

Electric grids have moved a long way from their initial status to the current precise systems we have. Still, the evolution stopped somewhere in the 1960s and, thereafter, there has not been a significant change in the technologies employed or the backbone working mode. Smart grids focus precisely on this gap, trying to combine ICT with power networks, in order to bring the latter ones into the 21st Century. The first step to this revolution was taken in the early 1990s when the UK Government privatised the electricity supply industry in England and Wales for the sake of competitiveness. This process was subsequently followed in many other countries. In most cases, the restructuring involves separating the electricity generation and retail from the natural monopoly functions of transport and distribution, breaking the old hierarchic vertical structure. The deregulation, in turn, leads to the establishment of a wholesale electricity market for electricity generation and a retail electricity market for electricity retailing.

In any case, the underlying layout remained the same and there are basically two separate physical networks: the high-voltage (HV) ones (the higher the voltage, the less energy transformed into heat and, therefore, lost), which transmits the electricity nation-wide from the power station to the regional distribution networks, and the low-voltage (LV) ones, which deliver the electricity regionally to the client. Further, there are a number of elements that control the energy flow to make sure it is done in a safe way. Finally, at the bottom of the system, meters record the energy consumption of each client. Depending on the size and type of these customers, we may have different kinds of meters (e.g., residential in LV, industrial in LV or HV, *etc.*) Currently, most of the LV meters worldwide have to be read *in situ* (remote metering the most important function to be fulfilled in *smart metering*). In the upper layer, the Substation Data Concentrators (SDC) gather the load data from a number of meters (though sometimes, as in the USA, this component is replaced by direct access to the meters or access through a gateway). Then, the SDCs and, eventually, some meters, hang from the *Secondary Substation* (SecST), which manages the

energy deliverance in a certain zone. Further up, the *Primary Substation* (PriST) receives HV energy and lowers it to the required distribution network voltage. Further, it is in charge of a number of SecSTs, SDCs, and, eventually, some industrial HV meters. Moreover, the protections, switches, and relays spread all over to prevent damages on the aforementioned elements and in the lines in case of failure or extreme conditions. Finally, in the uppermost level, human operators are in charge of managing current power network infrastructures. In this task, they are supported by diverse SCADA (Supervisory Control And Data Acquisition), EMS (energy management system), and DMS (demand-management system), conforming the so-called Central Controller Centre. As we may see, the architecture is framed in a very tight hierarchy.

This picture has been blurred lately by the apparition of new technologies and models with mixed roles. First, distributed generation has broken the unique direction of the energy and now it may flow either way. Second, smart grids will enable data interchange from the client to the utility, and vice versa (e.g., to transmit a price signal), and between all the members of the systems, enabling customer-side active policies and new client-provider relationships (e.g., Demand-Side Management, a voluntary re-scheduling of customer's demand to smooth consumption peaks [17]). Third, *micro-grids* and the EV, for instance, out-shape the classic rigidly-layered architecture by merging features of all the levels. Micro-grids, on one hand, combine low-scale energy generation and storage to provide electricity to their customers, and may be connected to the grid or work as an isolated isle; therefore, they contain generation, distribution, consumption, and optionally, utility and transport-related functions. EVs, on the other hand, may act as energy consumers and *generators* (using their batteries), in both roles introducing an additional obstacle: the ability of changing places (*i.e.*, not being attached to a certain distribution grid).

Moreover, another of the challenges the smart grids must face is the unification (when possible) and untangling of the related protocol mess. NIST (National Institute of Standards and Technology) originally identified 16 key smart grid standard protocols. This initial document was later modified according to the report sent by the EPRI (Electric Power Research Institute), increasing the standards list to 77 items [1]. The major domain authorities (EPRI and IEC, the International Electro-technical Commission) have backed the development of a central smart grid standard, the so-called CIM (Common Information Model, [18]) and, then, defined its OWL model, laying the foundations for many research projects that have extended the CIM to include related electric protocols [19], combined the CIM ontology with other protocols' ontologies to achieve interoperability [20], *etc.* The EPRI has issued a number of recommendations to carry out this harmonisation, which could be implemented through OWL language assertions, once both data models are converted to its semantic form, thus also allowing us to perform automatic reasoning and inference processes [21–23].

3. Related Work

So far there is an intensive research in order to archive smart grid vision. In the context of intelligence systems, some authors consider that even though the connection between intelligent systems and power system engineering is beneficial for both communities, there is little visibility of power and energy application at AI/Intelligent systems events or in journals [24]. In fact according to Ramos and

Liu [24], power system engineering and energy market problems are still requiring research in logic reasoning, heuristic search, perception and the ability for handle uncertainties. So far, neural networks, reinforcement learning, genetic algorithms, fuzzy systems, particle swarm optimization and multi-agent systems are the usual AI techniques used for confronting Smart Grid problems. For instance, Cartterson *et al.* [25] discuss the processing of electric grid data through a number of artificial intelligence techniques (e.g., Constraint Programming, rule-based Expert Systems considering fuzziness) applied to diverse application domains such as autonomous control of distribution networks, condition monitoring, post-fault analysis or voltage sag and swell monitoring. To this end, they also put forward a multi-agent architecture but do not go beyond stating that CIM and the IEC 61850 should be the key standard for syntactical interoperability (not semantic). Frederico *et al.* [26] follow a machine learning approach based on genetic algorithms and Fuzzy programming. Given that the flexibility of the machine learning methods depend on the training data sets, Frederico *et al.* approach is static with respect to events that are not in the scope of the training data sets. Paricio-Garcia *et al.* [27] address similar challenges to ours but their work is too focused on multi-agent system aspects (e.g., all agents interactions exclusively performed in FIPA ACL—Foundation for Intelligent Physical Agents' standard proposal for Agents Communication Language) and do not clarify whether they support real-time stream processing. Moreover, they introduce agents having a knowledge base expressed in rules but do not specify which. Ramchurn *et al.* [28] present an approach based on an FIPA-compliant multi-agent system to reduce household energy bills, but their focus is smaller than ours since we address the whole smart grid scenarios. On the other hand, Ramchurn *et al.*'s approach does not consider industrial standard such as CIM and ICE 61850.

In general terms, even though institutions such as NIST and EPRI have pointed out that semantic data models and semantic engineering processes are required for reaching the vision of the smart grid (indeed the EPRI has suggested a set of IEC standards for leading with semantic interoperability in smart grid), there is no proposal that includes semantic data models to support decision making. On the other hand, to the best of our knowledge there is no initiative addressing the intelligent software agent paradigm with the features applied to the smart grid, such as real-time decision making, intelligence distribution management and semantic interoperability.

4. System Architecture

The power network operation needs to be safe, reliable and, as long as possible, cost effective. The new scenario enabled by the massive adoption of distributed generation and storage and the intensive application of information technologies render the old centralised and static architecture infeasible. The main reason is that gathering the complete information to achieve optimal management, if possible at all, would entail a prohibitive cost in infrastructure and time.

Therefore, we have to find a trade-off between distributing the intelligence and the costs associated to this decision. In the aforementioned ENERGOS project [14], the anchor for addressing this challenge is an intelligent node called PGDIN (Power Grid Distributed Intelligence Node) [21]. A PGDIN encapsulates a variety of powerful technologies for event and data processing of the grid, in an intelligent and autonomous way. Figure 1 shows the PGDIN's components design. The ENERGOS architecture

stems the data and event processing based on a logical grid of PGDINs, which are widespread all over the physical assets of the power grid. Then, the intelligence that enables information processing is also distributed between the nodes as long as the old physical architecture is kept. As depicted in Figure 2, PGDINs are placed in the most important nodes of the power grid (*i.e.*, Primary Substations, Secondary Substations, Data Concentrators, *etc.*), as well as at the Utility in order to control those assets (in the first case) or carry out business processes involving data of the ENERGO network (in the latter).

Figure 1. PGDIN design.

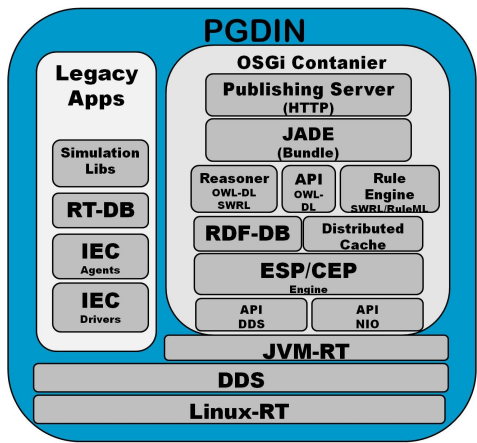
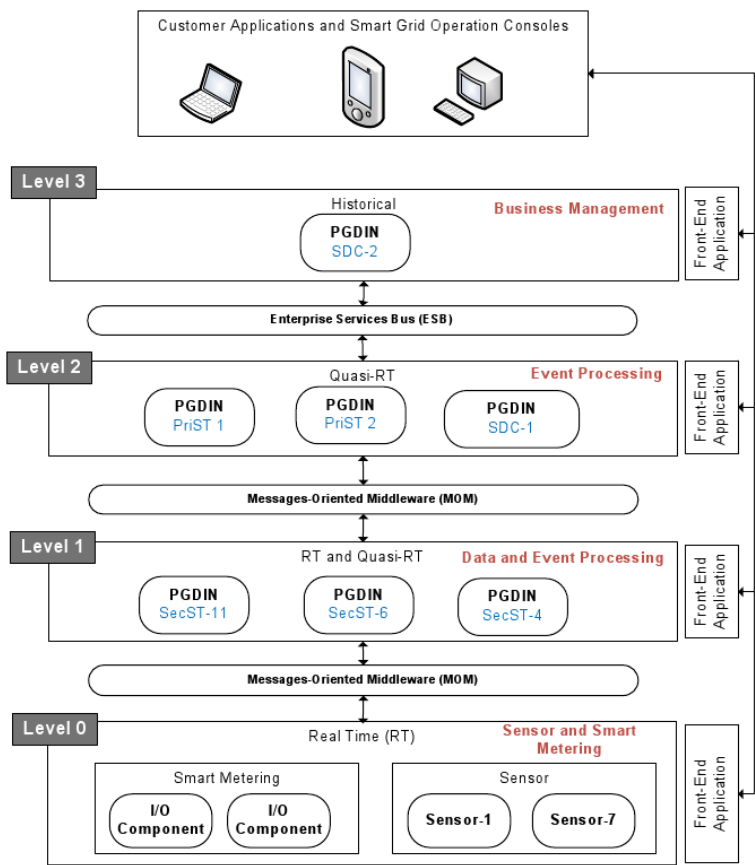


Figure 2. ENERGO architecture in four layers.



PGDINs behave as intelligent agents that may carry out diverse collaboration algorithms and business rules as well as download several event processing configuration sets and adaptive components. They can act either in an autonomous manner, using their local information, or cooperate with other nodes to resolve events and analyse specific grid situations. In any case, these operations present diverse abstraction levels (e.g., primary substations, metering or control centre) and time horizons (historical (H), quasi real-time (Q-RT) or real-time (RT)). PGDINS are spread in the network following two principles that assure that every single intelligent node is autonomous enough to manage all the assets under its concern.

1. *Location Abstraction Principle*: Every PGDIN must be able to deal with the information relevant to the set of assets it manages. Moreover, each PGDIN is responsible to store and manage that information from the distributed cache.
2. *Decision Abstraction Principle*: Every PGDIN must be able to evaluate any decision to be taken about the assets it manages.

Further, PGDINs has been designed to understand the most important existing standards in the electricity sector guiding the operational and business areas in which the intelligent nodes are deployed (e.g., IEC 61970, IEC 61850 or IEC 61968). These standards have been integrated in a semantic manner using the ontology specification language OWL-DL (*Ontology Web Language Description Logic*, a decidable subset of OWL), starting up from the UML versions of the standards (such as the CIM and SCL models) and, based on the OWL CIM [23]. The aforementioned OWL-DL ontology enables the PGDIN to deal with a unified domain that combines all concerning standards. In order to be able to cope with all the requirements discussed before, the PGDIN is composed of the following modules and technology interfaces (the rationale of choosing the technologies presented next has been discussed and justified in [16,21–23]), as shown in Figure 1.

4.1. PGDIN Components

As explained in the previous section, the PGDIN is composed of several technologies or components. The components' description are described in Table 1, as well as the reasons for selecting each of these components. Figure 1 shows the PGDIN's components design.

The PGDIN design has been currently implemented in the version 1.0, in which some design technologies have been included. In Table 1, the vendors/version of the implemented technologies for PGDIN 1.0 are also indicated.

The PGDIN includes the most powerful technologies for supporting the implementation of a Smart Grid. Some of these technologies are implemented in the PGDIN 1.0, encapsulating at least one of the PGDIN's components that are described in Table 1.

Table 1. Technology specification for the PGDIN implementation (Version 1.0).

Technology	Vendor (Version)
<i>Legacy Apps</i> Existing power grid applications.	Not implemented in this version
<i>OSGi</i> The Open Services Gateway initiative (OSGi) supports to encapsulate as a bundle an application/component.	Karaf (2.3.0) [29], Knopflerfish (R4 v4.2) [30]
<i>JADE</i> For implementing agent applications.	Not implemented in this version
<i>Publisher Server (HTTP)</i> Publishes on-line the power grid data and process.	Jetty (9.0) [31]
<i>Reasoner (OWL-DL, SWRL)</i> Reasoners are used for consistency checking and for increasing the intelligent capabilities of the power grid (inference process). Reasoning is performed over the languages OWL-DL and Semantic Web Rule Language (SWRL)	Pellet (2.3.0) [32]
<i>Rule Engine (SWRL, RuleML)</i> Supports the business rule execution that are specified on languages SWRL (for ontologies) and the RuleML (native language).	Jess (7.1) [33]
<i>API OWL-DL</i> The Application Programming Interface (API) for OWL-DL supports the connectivity between the semantic repository and reasoners.	Jena Ontology API (0.9.4) [34]
<i>RDF-DB</i> The Resource Description Framework (RDF) [35]—Data Base (DB) is the semantic repository to store the data in each PGDIN as OWL triplets, as well as the ENERGOS Ontology subset (OWL Profile).	Jena TDB (0.9.4) [36]
<i>Distributed Cache</i> The PGDIN maintains a real-time data repository supported by distributed object caches with persistence capabilities among the components.	Oracle Coherence (3.7.1) [37]
<i>ESP/CEP</i> The Event Stream Processing (ESP)/Complex Event Processing (CEP) supports to anticipate deviations and errors about the grid function.	Esper (4.3.0-for personal computers) [38], Oracle EP (11g-for servers computers) [39]
<i>API DDS</i> The API Data Distribution Service (DDS) supports the connection to the DDS bus.	Not implemented in this version
<i>JVM-RT</i> It is the Java Virtual Machine (JVM) for Real-Time.	Java RTS (2.2) [40]
<i>DDS</i> It is the middleware for publishing/subscribing to the services of the PGDIN.	RTi (4.5) [41]
<i>Linux-RT</i> Operating system over which all PGDIN technologies are installed, and then this is for real-time.	Core Real Time (2.6.31-11-rt) [42]

These technologies are explained as follows:

- *Data-distributed repositories*: Real-time repositories supported by distributed object caches with persistence capabilities between the components. Moreover, it includes a semantic database that allows storing event information either to be directly processed by the agent or for semantic inference. This technology is implemented by means of the components: Oracle Coherence and Jena TDB.
- *OWL-DL reasoning and SWRL inference*: OWL-DL enables the usage of reasoners to extract information on incomplete datasets and to evaluate their consistency. Further, it stores implicit information expressed in the ontology, custom power network operation rules (expressed in SWRL), and new assertions or knowledge (e.g., about certain problems, assets behaviour, *etc.*) that has been automatically generated at the inference process. This technology is implemented by means of the components: Pellet, Jess, and Jena Ontology API.
- *Agent Platforms*: The agents hosted in the platform present the ability to follow certain behaviours and to react upon specific situations. Moreover, they may communicate with other agents to collaboratively reach a certain goal (e.g., anticipate incorrect behaviour of the network, restoring the supply, issue a consumption forecast, coordinate distributed generation, *etc.*) There are several approaches and alternatives to implement agents following the FIPA specifications, and the PGDIN implements the JADE bundle (eventually, depending on the final hardware platform and its performance, some PGDINs may alternatively implement a light-weight agent platform such as LEAP). Further information on the PGDIN design can be found in [21]. This technology is implemented by means of the components: JADE and OSGi.
- An *OSGi* (Open Services Gateway initiative) *container*: It allows to apply life-cycle management (with functions such as install, stop, or start) through a set of well-known APIs, facilitates the detection, addition and removal of services, and allows them to react in consequence.
- *SOA* (Service-Oriented Architecture) *interface*: SOA helps to encapsulate legacy applications (e.g., SCADA, EMS or DMS applications) as services. This technology is not implemented in PGDIN 1.0.
- *Message-Oriented Middleware* (MOM): MOM enables bidirectional real-time reliable communication, which allows handling large amounts of information and can adapt to changes. More accurately, PGDINs implement the OMG DDS (Object Managing Group's Data Distribution Service standard) 1.2 standard. Similarly, DDS allows real-time data subscribing and publishing with parameterisable quality of service. This technology is implemented by means of the components: DDS and API DDS.
- *ESP* (Event Stream Processing) / *CEP* (Complex Event Processing): ESP/CEP enables processing events (e.g., route, aggregate, filter) and allows complex calculations and execution of rules at runtime. This technology is implemented by means of the components: Esper and Oracle EP.

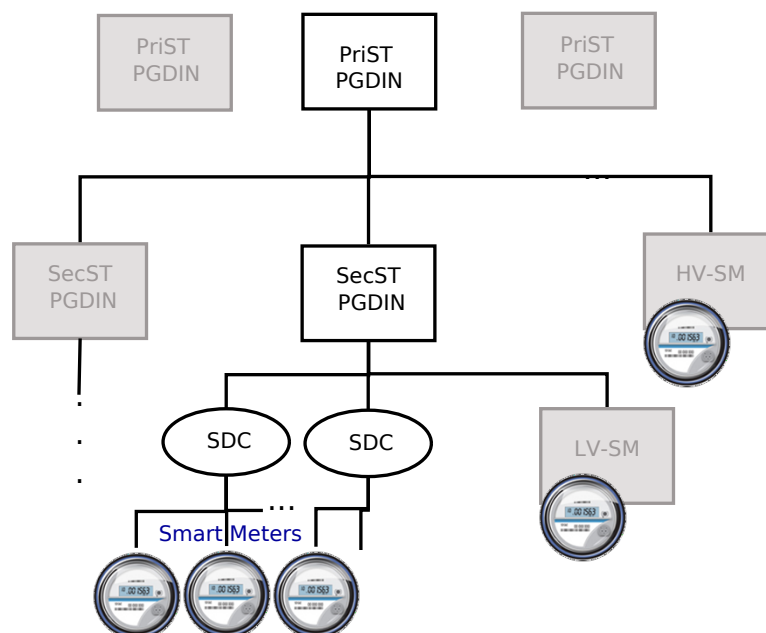
4.2. Communication and Data Exchange

Sharing the information in this architecture is achieved through the DDS bus and following a *publish/subscribe* communication paradigm. Therefore, PGDINs select the information susceptible to

be consulted by the rest of the network components and publish it. Interested peers only have to use the subscribe mechanism offered by DDS and they will receive promptly upcoming information as soon as it is generated. Eventually, one PGDIN may require another to publish certain information but, as a rule, depending on their physical location and task, they all share a predefined set of data-points. Still, there are some legacy applications and systems that are not compliant with this paradigm and offer their information on a request-response basis (for instance through a Service-Oriented Architecture interface). This possibility is not mandatory and is reserved for PGDINs with Internet connection (e.g., PGDINs at the utility, carrying out business process involving PGDINs-network-generated information and additional information from the aforementioned legacy systems). In any case, the shared information ranges from *usual* (e.g., without time constraints such as historical load data, active and reactive power consumption in different time horizons) to *urgent* (e.g., presenting strict real-time requirements such as alarms from the ESP/CEP engine, security sensor data from primary or secondary substations, *etc.*) DDS enables sharing information in different time frames through the same bus due to its widely-parameterisable QoS configuration.

The collaboration between PGDINs helps to resolve specific situations that, for instance, require comparing or combining data from several intelligent nodes. Furthermore, these interactions can also be understood from an interoperability perspective problem in which all the PGDINs own a common ontology. Figure 3 illustrates the proposed architecture, focusing on the deployment of the PGDINs and the communication technologies.

Figure 3. PGDINs involved in the energy imbalance billing use-case.



The flow of events orchestrated by the node occurs as follows: field events (alarms, measurements, *etc.*) potentially produced by the various devices are grouped and routed using protocols adapters and the DDS bus, then converted to Java objects, and finally injected into an engine for event processing. This engine is responsible for detecting patterns of unusual conditions such as sudden changes in measured quantities, cutting conditions or supply problems. Eventually, intelligent decisions about these

situations require the evaluation of rules and semantically rich query execution. Additionally, the event processing engine is responsible for feeding information repositories based on two technologies: first, a distributed cache of objects that represent the state of configuration and real-time information on devices (“Distributed Cache” in Figure 1), and second, a semantic database that stores information on events that must be addressed through collaboration with other agents or semantic inference (“RDF-DB”, or database of RDF triplets, in Figure 1).

4.3. Sensing the Environment

As already mentioned, all PGDINs are connected to the DDS bus, which is their standard way of communication. Each PDGIN has a knowledge base obtained from harmonising the semantic data model of CIM and IEC 61850. The characterisation of each knowledge base is an OWL profile (*i.e.*, a subset of the harmonised semantic data model between CIM and IEC 61850) and an electric model that is CIM/IEC 61850 compliant. Hence, a profile is the information that they own *a priori* (in a Kantian sense) of the world around. Since the knowledge in this profile also determines the valid vocabulary for each PDGIN, the knowledge base defines the information that may be exchanged in their communication acts. Further, when the PDGINs *discover* or *infer* new knowledge, it is stored as RDF entities in their knowledge base, allowing in this way sharing this newly *created* information easily.

As discussed above, the electric system is strictly hierarchical but PGDINs do not necessarily follow this hierarchy, since there are some operations that may request the collaboration of non-*sibling* PGDINs. In order to capture these requirements, we have defined a semantic data model characterising the logical layers of the architecture. As shown in Figure 4, we have extended the CIM ontology to represent the GridAgent. This novel concept allows us to identify groups of PGDINs in each layer depending on their function or geographical situation (*i.e.*, belonging to a certain area), farther from their topological relationship within the grid hierarchy (which is still crucial, as we will see). Thus, the PGDINs can infer from their knowledge base which PGDINs are *virtually* connected to it. Please note that the DDS bus already connects them all, but this fact does not imply that a PGDIN knows all existing peers due to scalability and performance problems that might appear in big power networks.

If a PGDIN wants to know which other intelligent agents are controlling assets connected to its asset, it can follow the semantic data model describing the architecture-layers and its electric model (the harmonised data model and, thus, CIM compliant [23]). CIM-compliant electric models allow identifying a particular topology that defines the connectivity between power systems resources in an electric network. This topology is basically based on three CIM classes:

- **Terminal**: An electrical connection point to a piece of conducting equipment. Terminals are connected at physical connection points called *connectivity nodes*.
- **ConnectivityNode**: Connectivity nodes are points where terminals of conducting equipment are connected together with zero impedance.
- **ConductingEquipment**: The parts of the power system that are designed to carry current or that are conductively connected therewith. `ConductingEquipment` is contained within an `EquipmentContainer`, which may be a primary substation, a voltage level or a bay within a primary substation.

Figure 4. Set of classes that characterise the relation between ENERGOS-Layers and GridAgent/PDGIN.

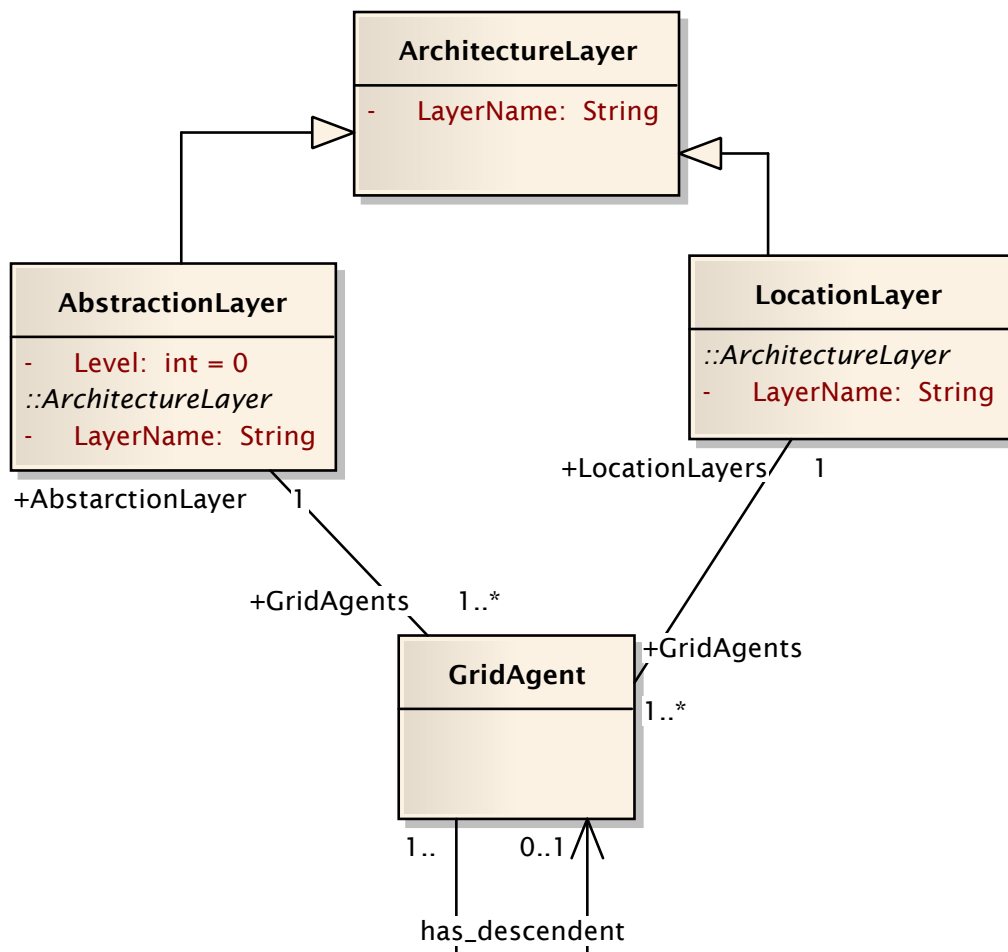
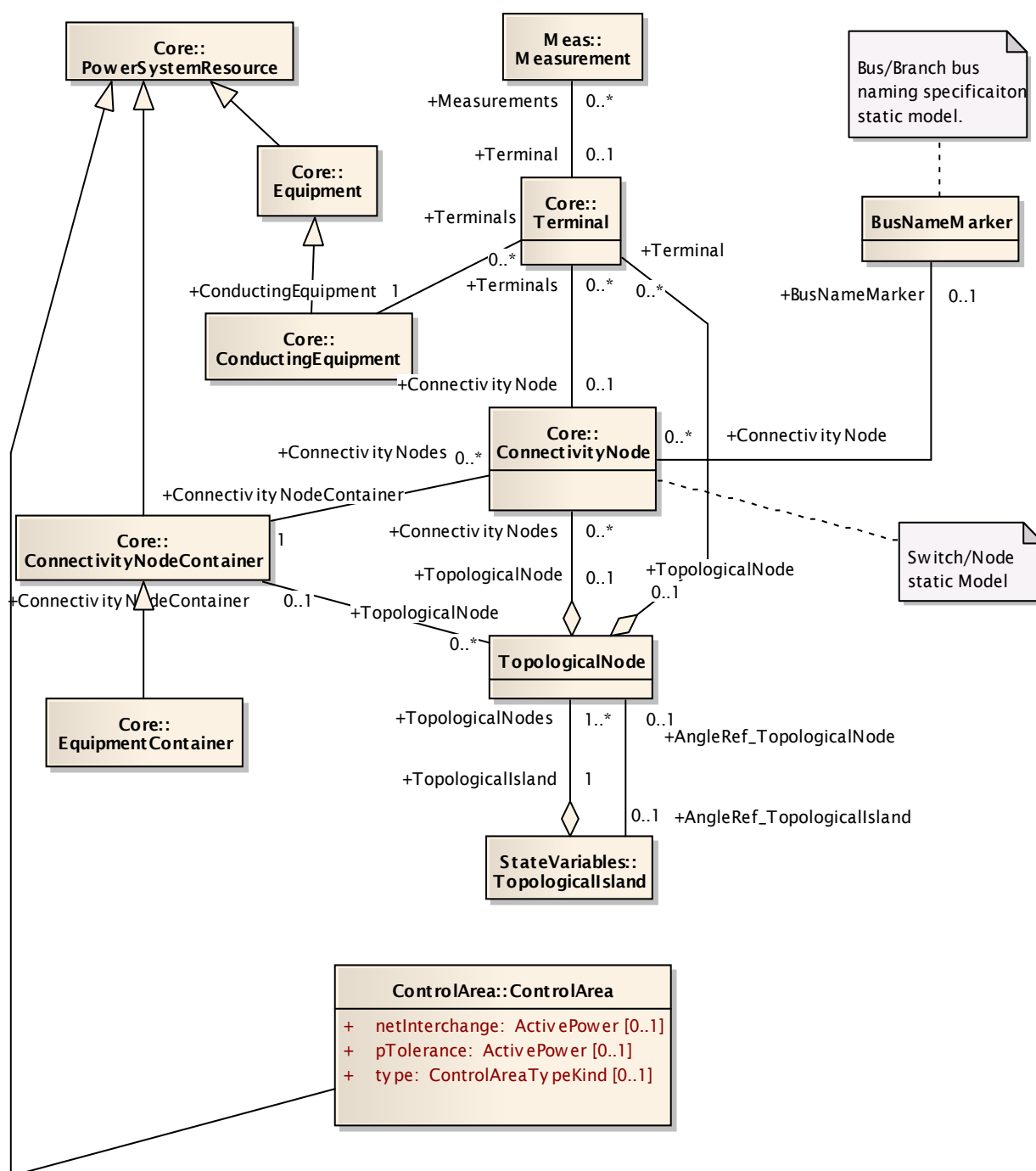


Figure 5 portrays the relationship between these classes. Observe that since the `terminal` class is related to the `measurement` class, it is also used to define points of connectivity (*i.e.*, related measurement in the electric network such as power flows, currents and voltages). In this way, the PGDIN has annotated in its knowledge base the information on the asset or assets it controls. For instance, it may find (by means of SPARQL queries upon its “RDF-DB” in Figure 1 [16]) to whom is that asset connected. Subsequently, having inferred the URI of the connecting asset, it may ask in the DDS bus who is the one in charge of that asset. The response will be stored similarly as an RDF rule so this operation does not need to be repeated in the future. In summary, this novel semantic description allows PGDINs to work independently of their location (Figure 4) and discover other peers they need to interact with (Figure 5).

Figure 5. UML diagram of the CIM ontology part dealing with the connection of power system resources.



5. Use-Case: Energy Imbalance Billing at the Primary Substation

The use-case presented herewith aims at illustrating how the ENERGOS architecture allows intelligence, responsibility, and action distribution. Current power networks present an automatic mechanism to couple with (non-severe) imbalances: in these cases (*i.e.*, supply and the demand do not match in a certain area), power flows into or out of the distribution system, and the alteration is absorbed by reserve power plants appointed to that end and placed at the transmission system. Moreover, in order

to be able to locate, invoice and charge occurred deviations, the Transmission System Operator (TSO) performs *ex-post* the so-called *Energy Imbalance Billing* (EIB) process. More accurately, EIB aims at calculating the imbalances occurred during a certain time period (usually a day) operation between the contracted consumption and the actual generation. The objective of EIB is detecting and charging accordingly deviations in one sense or another made by generators or customers.

Carried on from the PriST PGDIN perspective, load balancing consists in comparing the current value of its own meter with the aggregated consumptions registered by the secondary substation meters connected to it (and, eventually, SDCs or HV-meters directly connected to the PriST PGDIN). If the amounts do not match, the PriST PGDIN triggers an alarm and the corresponding PGDIN at the utility starts a process to clear the source of the anomaly (*i.e.*, determine whether there is a failure in a meter, a potential fraud, *etc.*) It causes subsequent reply processes in the SecSTs since they start a mirror validation to check that their own meter does equal the aggregated consumption of the SDCs and meters attached. Therefore, assuming the power network presents a family-tree architecture (Figure 3 illustrates the kinship in the use-case), this process involves coordinating an ancestor root-node (PriST) and all the descendants until the youngest members or leaves (the smart-meters).

5.1. ESP/CSP-Based Energy Imbalance Billing

The PriST PGDIN performs the energy imbalance billing check according to a pre-established frequency (daily or more often). In order to ensure that this process is carried out within at least quasi real-time constraints, in our architecture this task is performed by the ESP/CEP engine. DDS offers real-time communication and so do a number of CEP engines (see [43] for real-time performance tests of Exper, an open-source CEP). The third member of the triad needed to guarantee strict real-time response time, the PGDINs, depend on the hardware platform in which they will be actually deployed (this information will be disclosed when the project's industrial partners present the products designed within the project). So far, the knowledge processing algorithms (especially those dealing that deal with RDF triplets handling) are the main bottleneck that force to offer *only* quasi real-time.

First, the PriST PGDIN applies an event processing network (EPN) as follows (see in Listing 1 this EPN for an Oracle EP): first, it identifies the events (including those already cached) and issues a new composite event joining the simple ones. To this end, the PriST PGDIN has to discover the identity of the PGDINs susceptible to send information on this topic. To this end, as introduced in Section 4.3, it infers from the ontology profile the electric assets physically connected to the one(s) it controls. Then, it asks in the DDS bus who is the PGDIN managing that/those asset(s) (note that these tasks are carried out only once and thereafter, the information is retrieved directly from the knowledge database) and, finally, it selects a subscription of the event through the DDS bus. Second, the PriST applies an EQL (Event Query Language) query to validate the composite event (EQL is similar to SQL but is vendor-dependent, so each CEP/ESP engine presents a different specification depending on the vendor). The information of the single events is published by the corresponding SecST PGDINs encapsulated as `cim:Readings` into a Plain Old Java Object (POJO), and is injected into the ESP/CEP engine through the DDS connector. Thus, the PriST PGDIN, previously subscribed, will receive the information upon publishing. As one may think, there is neither order nor orchestration on publishing this measurement values. Therefore,

the PriST PGDIN actually receives a stream of events that must be either processed directly or stored in the cache to be analysed later. Moreover, there are a number of basic SQL (Structured Query Language, for managing data in Relational Database Management Systems) concepts (e.g., aggregation through grouping, filtering or correlation through joints) that can be carried out. Unfortunately, EQL is not flexible enough and cannot apply any semantic property as we will do in the next use case (note that in Listing 1 concepts such as `MeterReadings` are not expressed according to the unified data model described in Section 4.1 and, thus, no name-space is cited). Finally, there is a rule that validates that the aggregated and the retrieved own values match. Hence, if this constraint is not fulfilled, the PriST PGDIN as mentioned above will react immediately by injecting a new `alarm` event into the CEP/ESP bus, which will be then processed either by the same PGDIN or by its predecessors.

Listing 1. EQL query for energy imbalance billing and fraud detection in real time.

```
<?xml version="1.0" encoding="UTF-8"? >
<wlevs:config xmlns:wlevs="http://www.bea.com/ns/wlevs/config/application">
<processor>
<name>LoadBalancingAndFraud_Processor</name>
<rules>
<query id="LoadBalance"> select "LOADBALANCE", b.StationID, b.TimeStamp from
T4_T3_Meter_Readings as a, TimeStampTotalAggregation as b where
a.TimeStamp = b.TimeStamp and ( (a.T4_T3_Reading > (b.TotalAggregation +
LoadBalancingProcessor.threshold)) or (a.T4_T3_Reading <
(b.TotalAggregation - LoadBalancingProcessor.threshold)) )
</query>
</rules>
</processor>
</wlevs:config>
```

5.2. Historical Load Energy Imbalance Billing

Performing energy imbalance billing based on historical data (say last month's load balancing) requires using a different path than in the previous case. This is because the distributed object cache will most likely not be able to store measurement events for so long (definitely not if we open the window to, for instance, six months). Therefore, in this case the PriST PGDIN will not make use of the CEP/ESP: it will directly *contact* the corresponding underlying PGDINs. Contrary to the ESP/CSP-based approach of the previous case, this alternative involves much more components of the PGDIN.

According to the CIM protocol [18], the querying entity (*i.e.*, the PriST PGDIN) must create and deliver a `cim:MeterReadings` message, which includes 0 to infinite number of `cim:MeterReading` and a 0 to infinite amount of `cim:ReadingType` entities. This message structure may combine readings from one or more meters, each one with its own quality value. Now, the problem is how to send this message exactly to the PGDINs that can reply with the information required (*i.e.*, the PGDINs controlling the meters under this PriST). This challenge presents the same nature as the so-called *federation problem* in distributed systems [16]. That is, the receiving PGDINs

will eventually have to re-send the query only to a number of (again, underlying) PGDINs, before they are able to answer.

In the proposed system, the query federation problem is addressed semantically, as we have advanced in Section 4.3. Based on the description of the world it owns (according to the ontology profile kept in the knowledge base), the PriST PGDIN infers the assets connected to it, and asks for their respective managing SecST PGDINs. In this sense, the connectivity among PGDINs *emerges* from the semantic information inferred from the knowledge base and the expressiveness about the grid it has. Once the receivers of the query are clear, the PriST PGDIN subscribes to the corresponding data sources to retrieve the load curve of the required energy imbalance billing time window. Subsequently, the involved SecST PGDINs perform the internal SPARQL query detailed in Listing 2, to locally retrieve the load consumption for that window (local consumption is stored in the RDF repository).

Listing 2. SPARQL query to retrieve locally-stored load data.

```
SELECT ?intRead ?val ?rdnUnit ?rdnMulti ?quality WHERE {
  ?metRdns a mr:MeterReading; mr:MeterReading.IntervalBlocks ?intBlock;
  mr:MeterReading.MeterAsset ?metAsset; mr:MeterReading.valuesInterval ?valInt.
  ?valInt a mr:DateTimeInterval; mr:DateTimeInterval.beginDate ?beginDate;
  mr:DateTimeInterval.endDate ?endDate.
  ?metAsset mr:IdentifiedObject.mRID ?uri
  ?intBlock a mr:IntervalBlock; mr:IntervalBlock.IntervalReadings ?intRead;
  mr:IntervalBlock.ReadingType ?rdnType; mr:MeasurementValue.timestamp ?tms.
  ?rdnType mr:ReadingType.kind ?rdnKind; mr:ReadingType.multiplier ?rdnMulti;
  mr:ReadingType.unit ?rdnUnit.
  ?intRead rdf:type mr:IntervalReading; mr:IntervalReading.value ?val;
  mr:IntervalReading.ReadingQualities ?rq.
  ?rq mr:ReadingQuality.quality ?quality. FILTER regex(?rdnKind,"demand").
  FILTER regex(?uri,"M123342343"). FILTER (?beginDate >=
    "2010-01-01T00:00:00"^^xsd:dateTime && ?endDate <=
    "2010-12-31T23:59:59"^^xsd:dateTime ). }
ORDER BY ?tms
```

The SPARQL engine will process the query and issue an answer in which, unlike in the previous case, the information belongs to a period (say load curve) and, therefore, is represented as `cim:IntervalBlocks` [18].

In case another sort of *kinship* would be needed (say, 2nd degree, *i.e.*, assets connected to assets connected to mine), the reasoner would be required to apply in this case the *transitive* property to get all 2nd degree descendants. Parallel to the process started by the PriST PGDIN, the requested SecST PGDINs will start mirror sub-process to retrieve from their underlying electric assets the registered consumption for the required period (in a similar fashion). Upon arrival of the information, they will use the reasoner to apply the formula that will compare the received aggregated load with the registered one (*i.e.*, retrieved by means of the SPARQL query). In case they match, it will return the aggregated load curve to the PriST PGDIN. Finally, as in the case of the SecST, the PriST PGDIN checks its own meter's value (obtained by a local SPARQL query) and compares it to the retrieved one (by the WS). In

this use-case, the comparison is trivial, but the built-in reasoner and the rule engine allow for much more complicated operations.

6. Conclusions

The evolution of the classic power grid network has forced a shift in the dominating electric paradigm. Smart grids demand novel technologies to answer new urgent needs, such as dealing with bidirectional information flows, modelling the electric domain, or replacing the centralised static intelligence by a decentralised approach. Based on the CIM specification (a common language for describing all the assets and management terms of the electric grid) defined and maintained by the IEC, the UML model can be converted into a semantic OWL representation, opening the way to exploiting the use of semantic web technologies.

The ENERGOS project we present here is the first initiative that addresses the smart grid challenge as a whole, based on the combination of real-time technologies, the multi-agent paradigm, and semantic tools. Hence, proposed architecture devises a multi-layer system where intelligent nodes (PGDINs) control a number of electric assets. These intelligent nodes own a profile of the CIM containing the part of the ontology needed to carry on their task. Moreover, they have a reasoner and a knowledge base storing RDF rules of the knowledge they infer. Finally, they use the DDS bus that allows communication ranging from strict real-time to much more relaxed constraints. Under these premises, the PGDINs are able to detect the world around and collaborate to perform complex tasks, such as the one presented here: energy imbalance billing (we have illustrated both situations, quasi real-time and based on historical data time). We prove with this use-case the feasibility of the architecture, portraying the interaction and role of the diverse components mentioned above (e.g., the unified data model, the DDS real-time bus or the multi-agent system). The presented architecture breaks the classical tight vertical scheme that usual power-networks have; unlike in these typical scenarios, the architecture addressed hereby enables horizontal communication and enhances the integration of applications on top.

Fully exploiting the possibilities that this intelligence allows includes, for instance, advanced processing and reasoning on huge data streams (expressed in RDF triplets) as shown in [16,22]. Future works will aim to refine the distributed service discovery and determine the physical limits of the PGDIN's processing event streams.

Acknowledgment

This research work has been partially sponsored by the Spanish CDTI (ENERGOS CEN2009-1048), ITEA2 IMPONET Project (TSI-020400-2010-103) and ITEA2 NEMO & CODED (IDI-20110864).

Angelina Espinoza started developing part of the research presented in this paper while she was working for the SYST group at the Technical University of Madrid.

References

1. NIST (National Institute of Standards and Technology). *NIST Framework and Roadmap for Smart Grid Interoperability Standards*; NIST Special Publication 1108; NIST: Gaithersburg, MD, USA, 2012.
2. Palensky, P.; Dietrich, D. Demand side management: Demand response, intelligent energy systems, and smart loads. *IEEE Trans. Ind. Inform.* **2011**, *7*, 381–388.
3. Vaccaro, A.; Velotto, G.; Zobaa, A. A decentralized and cooperative architecture for optimal voltage regulation in smart grids. *IEEE Trans. Ind. Electron.* **2011**, *58*, 4593–4602.
4. Ramachandran, B.; Srivastava, S.; Edrington, C.; Cartes, D. An intelligent auction scheme for smart grid market using a hybrid immune algorithm. *IEEE Trans. Ind. Electron.* **2011**, *58*, 4603–4612.
5. Higgins, N.; Vyatkin, V.; Nair, N.K.; Schwarz, K. Distributed power system automation with IEC 61850, IEC 61499, and intelligent control. *IEEE Trans. Syst. Man Cybern. Part C Appl. Rev.* **2011**, *41*, 81–92.
6. McArthur, S.; Davidson, E.; Catterson, V.; Dimeas, A.; Hatziaargyriou, N.; Ponci, F.; Funabashi, T. Multi-agent systems for power engineering applications. Part II: Technologies, MANUALs, and tools for building multi-agent systems. *IEEE Trans. Power Syst.* **2007**, *22*, 1753–1759.
7. Saber, A.; Venayagamoorthy, G. Plug-in vehicles and renewable energy sources for cost and emission reductions. *IEEE Trans. Ind. Electron.* **2011**, *58*, 1229–1238.
8. Cecati, C.; Citro, C.; Piccolo, A.; Siano, P. Smart operation of wind turbines and diesel generators according to economic criteria. *IEEE Trans. Ind. Electron.* **2011**, *58*, 4514–4525.
9. Calderaro, V.; Hadjicostis, C.; Piccolo, A.; Siano, P. Failure identification in smart grids based on Petri net modeling. *IEEE Trans. Ind. Electron.* **2011**, *58*, 4613–4623.
10. Sauter, T.; Lobashov, M. End-to-end communication architecture for smart grids. *IEEE Trans. Ind. Electron.* **2011**, *58*, 1218–1228.
11. Gungor, V.; Lu, B.; Hancke, G. Opportunities and challenges of wireless sensor networks in smart grid. *IEEE Trans. Ind. Electron.* **2010**, *57*, 3557–3564.
12. Gungor, V.; Sahin, D.; Kocak, T.; Ergut, S.; Buccella, C.; Cecati, C.; Hancke, G. Smart grid technologies: Communication technologies and MANUALs. *IEEE Trans. Ind. Inform.* **2011**, *7*, 529–539.
13. Yang, Q.; Barria, J.; Green, T. Communication infrastructures for distributed control of power distribution networks. *IEEE Trans. Ind. Inform.* **2011**, *7*, 316–327.
14. Peña, Y.K.; Garbajosa, J.; Ortega, M.; González, E. ENERGOS: Integral smart grid management. In *Proceedings of the 9th International Conference on Industrial Informatics (INDIN)*, Lisbon, Portugal, 26–29 July 2011.
15. Fensel, D.; van Harmelen, F.; Andersson, B.; Brennan, P.; Cunningham, H.; Della Valle, E.; Fischer, F.; Huang, Z.; Kiryakov, A.; Lee, T.K.; School, L.; Tresp, V.; Wesner, S.; Witbrock, M.; Zhong, N. Towards LarKC: A platform for web-scale reasoning. In *Proceedings of the IEEE International Conference on Semantic Computing (ICSC 2008)*, Santa Clara, CA, USA, 4–7 August 2008.

16. Peña, A.; Peña, Y.K. Distributed semantic repositories in smart grids. In *Proceedings of the 9th International Conference on Industrial Informatics (INDIN)*, Lisbon, Portugal, 26–29 July 2011.
17. Peña, Y.; Jennings, N.R. Optimal combinatorial electricity markets. *Int. J. Web Intell. Agent Syst.* **2008**, *6*, 1–13.
18. IEC (International Electrotechnical Commission). *System Interfaces for Distribution Management. Part 11: Common Information Model (CIM)*; IEC 61968 Application Integration at Electric Utilities; IEC: Geneva, Switzerland, 2012.
19. Taylor, D.; Kmethy, G. *DLMS/COSEM/CIM Study Report: 61968-9 Message Profiles for DLMS/COSEM*; Technical Report for DLMS User Association: Zug, Switzerland, 2010.
20. Breedillet, P.; Lambert, E.; Schultz, E. CIM, 61850, COSEM standard used in a model driven integration approach to build the smart grid service oriented architecture. In *Proceedings of the IEEE 1st International Conference on Smart Grid Communications (SmartGridComm)*, Gaithersburg, MD, USA, 4–6 October 2010.
21. De Mues, M.O.; Alvarez, A.; Espinoza, A.; Garbajosa, J. Towards a distributed intelligent ICT architecture for the smart grid. In *Proceedings of the 9th International Conference on Industrial Informatics (INDIN)*, Lisbon, Portugal, 26–29 July 2011.
22. Espinoza, A.; de Mues, M.O.; Fernandez, C.; Garbajosa, J.; Alvarez, A. Software-intensive systems interoperability in smart grids: A semantic approach. In *Proceedings of the 9th International Conference on Industrial Informatics (INDIN)*, Lisbon, Portugal, 26–29 July 2011.
23. Nieves, J.C.; de Mues, M.O.; Espinoza, A.; Rodríguez-Álvarez, D. Harmonization of semantic data models of electric data standards. In *Proceedings of the 9th International Conference on Industrial Informatics (INDIN)*, Lisbon, Portugal, 26–29 July 2011.
24. Ramos, C.; Liu, C.C. AI in power systems and energy markets. *IEEE Intell. Syst.* **2011**, *26*, 5–8.
25. Catterson, V.M.; Davidson, E.M.; McArthur, S.D. Embedded intelligence for electrical network operation and control. *IEEE Intell. Syst.* **2011**, *26*, 38–45.
26. Frederico, C.; Almeida, M.; Kagan, N. Using genetic algorithms and fuzzy programming to monitor voltage sags and swells. *IEEE Intell. Syst.* **2011**, *26*, 46–53.
27. Paricio-Garcia, A.; Oliver, J.; Gosch, D. An intelligent agent-based distributed architecture for smart-grid integrated network management. Presented at the 1st IEEE Workshop on Smart Grid Networking Infrastructure, Denver, CO, USA, 14 October 2010.
28. Ramchurn, S.D.; Vytelingum, P.; Rogers, A.; Jennings, N.R. Agent-based homeostatic control for green energy in the smart grid. *ACM Trans. Intell. Syst. Technol.* **2011**, *2*, doi:10.1145/1989734.1989739.
29. *Apache Karaf*, Version 2.3.0; Apache Software Foundation: Wilmington, DE, USA, 2012. Available online: <http://karaf.apache.org/> (accessed on 14 November 2012).
30. Knopflerfish: Open Source OSGi Service Platform. Available online: <http://www.knopflerfish.org/> (accessed on 14 November 2012).
31. Jetty Web Server. Available online: <http://www.eclipse.org/jetty/about.php> (accessed on 14 November 2012).
32. *Pellet: OWL 2 Reasoner for Java*, Version 2.3.0; ClarkParsia: Washington, DC, USA, 2012. Available online: <http://clarkparsia.com/pellet> (accessed on 14 November 2012).

33. Java Expert System Shell (Jess). Available online: <http://www.jessrules.com/jess> (accessed on 14 November 2012).
34. Jena Ontology API. Available online: <http://jena.apache.org/documentation/ontology/index.html> (accessed on 14 November 2012).
35. Resource Description Framework (RDF) Schema Specification 1.0. Available online: <http://www.w3.org/RDF/> (accessed on 14 November 2012).
36. Jena A Semantic Web Framework for Java. Available online: <http://jena.sourceforge.net/> (accessed on 14 November 2012).
37. *Oracle Coherence*, Version 3.7.1; Oracle Corporation: Redwood Shores, CA, USA, 2012. Available online: <http://www.oracle.com/technetwork/middleware/coherence/downloads/index.html> (accessed on 14 November 2012).
38. Esper. Available online: <http://esper.codehaus.org/> (accessed on 14 November 2012).
39. Oracle Event Processing. Available online: <http://www.oracle.com/technetwork/middleware/complex-event-processing/overview/index.html> (accessed on 14 November 2012).
40. Java Real-Time System. Available online: <http://www.oracle.com/technetwork/java/javase/tech/index-jsp-139921.html> (accessed on 14 November 2012).
41. RTI Connex 4.5f. Available online: <http://community.rti.com/content/page/rti-connex-45f-documentation> (accessed on 14 November 2012).
42. Linux Real-Time. Available online: <http://packages.ubuntu.com/lucid-updates/linux-rt-headers-2.6.31-11> (accessed on 14 November 2012).
43. Esper and NEsper Wiki. Available online: <http://docs.codehaus.org/display/ESPER/Home> (accessed on 14 November 2012).

© 2012 by the authors; licensee MDPI, Basel, Switzerland. This article is an open access article distributed under the terms and conditions of the Creative Commons Attribution license (<http://creativecommons.org/licenses/by/3.0/>).