Abstract: Given the challenges posed by the smart grid paradigm, we propose hereby a distributed intelligent ICT architecture, which is built upon a basic brick named PGDIN (Power Grid Distributed Intelligence Node) integrating a DDS middleware, an ESP/CEP engine, a distributed cache, and an OWL unified model for power standards. A case study dealing with substation monitor and control is also introduced.

keywords: {information technology;knowledge representation languages;middleware;power engineering computing;smart power grids;substation automation;DDS middleware;ESP-CEP engine;OWL unified model;PGDIN;distributed cache;distributed intelligent ICT architecture;power grid distributed intelligence node;power standard;smart grid paradigm;substation control;substation monitor;Engines;IEC standards;OWL;Ontologies;Real time systems;Semantics;Smart grids},

URL: http://ieeexplore.ieee.org/stamp/stamp.jsp?tp=&arnumber=6034985&isnumber=6034824
Towards a Distributed Intelligent ICT Architecture for the Smart Grid

Mariano Ortega de Mues (1), Alejandro Alvarez (1), Angelina Espinoza (2), Juan Garbajosa (2)
(1) Indra Sistemas, Energy Market Department
Madrid, Spain
{mortega,aacero}@indra.es
(2) Technical University of Madrid (UPM)
Madrid, Spain
{a.espinoza,jgs}@eui.upm.es

Abstract: Given the challenges posed by the smart grid paradigm, we propose hereby a distributed intelligent ICT architecture, which is built upon a basic brick named PG DIN (Power Grid Distributed Intelligence Node) integrating a DDS middleware, an ESP/CEP engine, a distributed cache, and an OWL unified model for power standards. A case study dealing with substation monitor and control is also introduced.

I. INTRODUCTION

First used during the 90’s, the Power Smart Grid concept made its official appearance in 2005[1] and since then, it is mobilizing one of the largest known technological effort.

Compared to the current electricity networks, designed to operate according to an unidirectional flow (generation - transport - distribution - consumption), future smart grids will enable the integration of distributed bi-directional flow power generation, enabled by the wide application of information technology devices. This radical change in the way of generating, transporting, distributing, and consuming energy arises many challenges that affect the entire business model around electricity. One of these challenges is the transformation of the current operation schema from a Central Control Center (where the intelligence required to manage network infrastructures is provided by human operators and great SCADA, EMS, and DMS systems1) into a management architecture based on distributed intelligent devices capable enough to perform complex event processings, simulations, and to execute high-level technical and business rules. This challenge will transform the network operation process from a top-down approach to a distributed control perspective [2].

II. MOTIVATION AND BACKGROUND

In this context, current ICT system architectures like SOA2, EDA3, cloud, real-time, etc. and associated technologies are a good starting point to investigate how it may be possible to distribute in a computational grid both massive real-time data about assets (e.g. status, configuration, etc.) and the intelligence needed to deal with power network operations and their events at various abstraction levels (metering, substations, etc.) in different time horizons (real-time, quasi-real-time, etc.).

The research presented herewith aims at introducing a distributed intelligent ICT architecture to deal with the aforementioned challenges by fixing requirements with a SOA/EDA platform, which deploys an uniform domain model, services for real-time messaging, distributed object cache, complex event processing, and semantic-oriented decision support.

This paper is organized as follows: Section III summarizes the technologies identified as relevant to address this challenge. Section IV introduces the proposed architecture by establishing its main features and capabilities, as well as defining the core component in its deployment, the Power Grid Distributed Intelligence Node. Section V details the case study developed and the results obtained with the current version of the components. Finally, Section VI covers the general conclusions and draws the line of future works.

III. SIX ICT TECHNOLOGIES FOR THE SMART GRID

The following 6 ICT technologies were identified as a key to face the aforementioned challenges:

1. - **Uniform Domain**: Given the large number of existing standards in the electricity sector that meet the diverse needs of representation of different managed areas (IEC 61970, IEC 61850, IEC 61968), etc., we have integrated them semantically using OWL4 profiles [3] over commonly-used UML models (CIM, SCL, etc.) [4][5][6].

2. - **Message-oriented Middleware**: We have analyzed diverse messaging systems available to connect the

---

1 SCADA (Supervisory Control And Data Acquisition), EMS (Energy Management System), DMS (Distribution Management System).
2 SOA (Service-Oriented Architecture).
3 EDA (Event-Driven Architecture).
4 OWL (Ontology Web Language).
components of the architecture taking into account the need for a reliable communication mechanism, real-time, bidirectional, adaptable to changes and capable of handling large amounts of information. Finally, the OMG DDS\(^5\) 1.2 standard was selected [7] with extensions for supporting WS\(^6\) and JMS\(^7\) [8].

3.- **Data-distributed grid**: The smart grid provides very diverse databases scenarios: real-time storage and information retrieval from sensors, large volumes of historical information, etc. We have studied the features and integration capabilities of different technologies available for the presentation of information in repositories (relational, object oriented, and semantic ones). Finally, we decided to articulate the platform around a grid of real-time data supported by distributed object caches with persistence capabilities among the components.

4.- **ESP/CEP\(^8\)**: Both models have emerged as useful tools in identifying adaptive patterns that allow real-time management of complex alarms, probability of incidents, fraud in the use of resources, strategic infrastructure attacks, and other critical situations. In the smart grid paradigm, all the situations previously mentioned present an increasing need for real-time computing and the ability to correlate large amounts of information together. ESP/CEP can be essential in the processing of events that will occur along the smart grid with their capability to filter, aggregate, route, and process large amounts of information according to complex calculations and rules modifiable at runtime.

5.- **OWL-DL reasoning and SWRL inference**: The OWL-DL\(^9\) ontology as one of the possible data models allows the use of reasoners to extract information on incomplete data sets and evaluate their consistency. Additionally, the information contained in the ontology can be completed with SWRL [9] rule sets in order to infer high-level knowledge about power grid operation, assets behavior or problem solving.

6.- **Agent Frameworks**: Many of the problems we face in the context of the intelligent network, could be better solved with the use of collaborative agents. These agents have the ability to execute a behavior and respond to specific situations and to dialogue with other agents to find the best way to reach a goal (e.g. restoring the supply, anticipate incorrect behavior of the network, estimate consumption, coordinate distributed generation, etc.). There are different approaches to implementing agents, one of the most widely used model is the BDI (Beliefs, Desires and Intentions) [10]. Beliefs represent knowledge that the agent has of its environment (in our case a significant portion of the assets of the grid), desires represent the objectives that the agent must reach (electricity supply) and intentions represent the actions that agent has decided to implement and should be shared with other agents.

IV. **A DISTRIBUTED INTELLIGENCE ARCHITECTURE FOR THE SMART GRID**

The proposed architecture is built on a grid of PGDIN nodes. These nodes act as agents that may download different configuration sets about event processing, adaptive components, business rules and collaboration algorithms. The nodes can be deployed at various technical levels of electricity networks (primary substation, secondary substation or central control) taking decisions autonomously and actively collaborating with other nodes in order to resolve events and analyze specific situations. The aforementioned OWL ontology allows the PGDIN work with the unified domain that integrates the standards in Fig. 1.

![Fig.1. Standards proposed for the Unified Meta-Model (OWL-DL)](image)

The PGDIN follows two main principles:

1st.- **Location Abstraction Principle**: Each and every one of the nodes in the grid should be able to take responsibility for the information relevant to the operation of any set of assets in the electricity grid. Each node is responsible for adding or removing from the distributed cache all the information on the assets it controls.

2nd.- **Decision Abstraction Principle**: Each and every one of the nodes in the grid should be able to deploy and evaluate any decision to be taken (business rules, calculation or inference) on the assets under it responsibility.

These basic principles suggest that any PDGIN is autonomous enough for managing all the asset under its concern. 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 a real-time messaging middleware (DDS), 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. Often, intelligent decision about these

---

\(^5\) DDS (Data Distribution Service).

\(^6\) WS (Web Services).

\(^7\) JMS (Java Messaging Service).

\(^8\) ESP (Event Stream Processing), CEP (Complex Event Processing).

\(^9\) OWL-DL (Ontology Web Language Description Logic), a decidable subset of OWL.
situations requires the evaluation of rules and semantically rich query execution. Additionally, the event processing engine is responsible for feeding information repositories based on two technologies, a distributed cache of objects that represent the state of configuration and real-time information on devices, and a semantic database that stores information on events that must be addressed through collaboration with other agents or semantic inference.

Regarding software components, the current PGDIN version will be running on a Linux Ubuntu 10.X (with a modified core for real time), will integrate a predictive JVM\(^1\) (1.6 compliant) and an OSGi\(^2\) container to manage components and frameworks deployed (Fig. 2). Several bundles will be also included:

1. A DDS 1.2 stack to manage messaging and quality-of-service agreements between components, extensions for WS, and JMS.
2. CEP/ESP java engine with high-performance capabilities and a wide set of connectors like JMS, WS, streams, etc.
3. Distributed object cache with real-time performance capability like EH-Cache.
4. ANN (Artificial Neural Network) framework for adaptive components like Encog.
6. A lightweight java servlet container like Jetty 7.0 for node administration using REST and JSON.
7. Cloud Computing capabilities with GridGain 3.0 or Java GAT 2.1.

\(^1\) JVM (Java Virtual Machine), \(^2\) OSGi (Open Services Gateway initiative)

V. CASE STUDY

Several PGDIN prototypes were assembled using components from multiple vendors (Oracle, IBM, TIBCO, RTi, PrismTech, etc) and open-source (Drools Fussion, Esper). In the current configuration, Oracle provided JVM for Real Time (JRockit) and the CEP processor (CEP Processor 11g), and Real Time Innovations (RTI), the DDS implementation.

In a first case study, this prototype was used to execute the following data valiations for two GNF (Gas Natural Fenosa, a Spanish Power Utility) power substations:

1. Check if Power Module exceeds fixed limits, with P (active power) and Q (reactive power):
   \[
   if(G = (\sqrt{P^2 + Q^2})) > 20
   \]

2. Check consistency of P, Q, Int (current), T (voltage) where
   \[
   I = (\sqrt{3} * T * Int/1000)
   \]
   \[
   if(|G - I| = 0, 0, |G - I|/|G|) < 0, 2
   \]

These controls are performed nowadays with a powerful SCADA system collecting field signals and running routines built in C verifying the module relation between analog values of P, Q, Int, and T for each substation.

Using the PGDINs, the signals were modeled as “cim: analogValues” \([6]\) instances, encapsulated into Plain-Old-Java-Objects (POJOs) and injected into the ESP/CEP engine through a DDS connector. This component filters all the events with the same time-stamp and performs data aggregation; then evaluates several mathematical models on the magnitudes and detects events with abnormal thresholds (Fig. 3.).

In a second case study, collected measures were introduced within an instance of the ontology discussed above with a SWRL rule-set holding the same validation algorithm over
P, Q, I, and V. After aggregation of relevant information by the ESP/CEP engine, all information (in RDF) was loaded into a Jess engine to evaluate the consistency and running a similar magnitude validation.

We have also developed a Web RIA\textsuperscript{12} console (with JQuery and Highchars) to simulate the SCADA console with a throughput of 1 sg (Fig. 4).

VI. CONCLUSIONS AND FUTURE WORK

Modern SOA and EDA architectures around a real-time data grid are interesting departure points in order to address the intelligence distribution challenge in the context of the smart grid.

ESP and CEP technologies provide useful elements for data segmentation, filtering, and aggregation of events. Moreover, semantic models can be used to achieve a successful standards integration, business knowledge representation and agent communication.

Finally, though ESP and CEP are real-time-oriented technologies, current semantic processing components are not designed to manage information in this time-frame. Recent investigations\textsuperscript{13} on semantic streams [11] and incremental reasoning [12] suggest the importance that will have in the near future the possibility of using both technologies together [13].

The next steps in our research include deepening in real-time integration among semantic elements in the node, modeling patterns for supply issues that could be detected, and a collaboration algorithm among agents.

ACKNOWLEDGMENT

This research work has been partially sponsored by the Spanish MEC (ENERGOS TIN2009-00000). We also want to thank the various manufacturers who have provided us with their different platforms and components necessary for making prototypes: Oracle, IBM, Tibco, RTi and PrismTech.

\textsuperscript{12} RIA (Rich Internet Application).

\textsuperscript{13} A special interest project on this subject may be found in the EU FP 7 Large-Scale Integrating Project LarKC (Large Knowledge Collider): http://www.larkc.eu/

\begin{center}
\[\text{Fig. 4. Case study RIA SCADA console.}\]
\end{center}

References


