Experimental Validation of BDI Agents for Distributed Control of Electric Power Grids

Issicaba, Diego; Rosa, Mauro A.; Prostejovsky, Alexander Maria; Bindner, Henrik W.

Published in:
Proceedings of 2017 IEEE PES Innovative Smart Grid Technologies Conference Europe

Publication date:
2017

Document Version
Peer reviewed version

Citation (APA):
Experimental Validation of BDI Agents for Distributed Control of Electric Power Grids

Diego Issicaba and Mauro A. Rosa
Labplan/INESC P&D Brasil
Department of Electrical and Electronic Engineering
Federal University of Santa Catarina
Florianopolis-SC, Brazil 88040–900
Email: diego.issicaba@inescbrasil.org.br

Alexander M. Prostejovsky and Henrik W. Bindner
Center for Electric Power and Energy Systems
Department of Electrical Engineering
Technical University of Denmark
Roskilde, Denmark 4000
Email: alepros@elektro.dtu.dk

Abstract—This paper presents initial laboratory experiments designed to test belief-desire-intention agent reasoning in a web-of-cell context. The work introduces the application of the bridge between JASON and the Common Artifact Infrastructure for Agents Open Environments to agent and environment modeling, respectively. Belief-desire-intention reasoning is achieved through JASON’s engine while artifacts monitor and control grid devices using dedicated JAVA objects. Experiments have been conducted in SYSLAB, a testbed for distributed power system control and distributed solutions, located at the Risø campus of the Technical University of Denmark. Experimental results show the feasibility of applying belief-desire-intention reasoning to WoC control using a test case where tie-line power flow setpoints must be followed.

Index Terms—Power distribution, Multi-agent systems, Smart grid.

I. INTRODUCTION

Innovative concepts are the main catalysts to rapid advancements in science and technology. The power industry has been evolving by absorbing innovative concepts in order to build more effective processes, technologies, products, services and business models. Innovations have been motivated by initiatives, which foment generation through renewable energy, interconnection of distributed energy resources, adoption of decentralised management solutions, customer participation and power system modernisation as a whole.

The main driver behind these initiatives is the changing landscape of electric power systems. The integration of small-scale renewable energy sources and the active involvement of power consumers challenge the established (de)centralized monitoring and control systems. The Web of Cell (WoC) approach proposes a distributed solution, where cells, as opposed to traditional control areas, possess a high degree of automation and can be located throughout all voltage levels [1]. This concept is promoted by the European Liaison on Electricity Committed Towards long-term Research Activity (ELECTRA IRP) [2], which targets aligning national research efforts towards developing new approaches for distributed control.

The WoC concept can be facilitated using the agent paradigm, which provides a suitable approach to design and test distributed solutions. Multi-agent systems have been already applied in areas of power engineering including monitoring [3], [4], [5], diagnoses [6], fault location, isolation and power restoration [7], [8], [9]. However, most of multi-agent applications to power engineering relies on the middleware Java Agent Development Framework (JADE) [10], ignoring aspects related to computational modeling of the environment. Also, most applications lack actual experimentation in laboratory environments and in the field.

This paper presents initial laboratory experiments designed to test belief-desire-intention (BDI) agent reasoning to WoC control. Goal-directed behaviors interrelated with agent planning are then emulated using JASON [11], a Java-based interpreter for an extended version of AgentSpeak, which in turn is based on the BDI architecture. The application of Common ArTifact Infrastructure for Agents Open Environments (CArTAgO) [12] is proposed, allowing both a bridge to JASON and environment artifacts. BDI reasoning is achieved through JASON’s engine while CArTAgO’s artifacts are designed to act over JAVA objects, which in turn can read and act over monitoring and controlling devices through a laboratory communication infrastructure. The experiments have been conducted in SYSLAB, a smart grid laboratory facility provided with smart components and flexible configuration. Experimental results show the feasibility of applying BDI reasoning to cell control, more specifically in a test case.
where tie-line power flow setpoints must be followed.

The paper is organised as follows. Section II introduces the agent-artifact framework [13] designed to the experiments. Section III presents results for initial laboratory experiments on BDI reasoning to WoC control. Section IV outlines conclusions and final remarks.

II. AGENT-ARTIFACT FRAMEWORK TO BDI CELL CONTROL

The key principle behind the WoC is to solve local problems locally by splitting the grid within cells, as illustrated in Fig. 1. Cells must communicate with each other to enable distributed collaboration and information exchange, in similarity with the block-oriented approach proposed in [13]. A cell operator is then responsible for establishing and maintaining autonomous control mechanisms contributing to an adequate and secure operation. Among the modeling possibilities for the cell operator, the BDI agent architecture stands out as a successful mechanism to model human-like reasoning, planning and decision-making, with the additional benefit of allowing analysis via model checking.

Fig. 1. Overview of the WoC concept that proposes a flat hierarchy across all voltage levels. [14].

The BDI model is utilised in AgentSpeak, a logic-based agent-oriented language whose recent versions can be interpreted using JASON. If well-utilised, JASON allows high-level representations of agent reasoning through AgentSpeak and, at the same time, a sophisticated use of legacy code and object-oriented models implemented in Java. Differently from JADEX, JASON belongs to the group of theoretically-rooted agent-oriented programming languages, outlining a strong emphasis on rigorous formal semantics [15]. Furthermore, similarly to JADEX, JASON can use the JADE infrastructure. JASON is the standard instance utilized in CArTAgO documentation, code examples and literature. All these features made JASON the natural choice towards modelling and testing BDI-based cell control processes.

A. Agent Modeling and Reasoning

Following the WoC paradigm, a cellagent has been abstracted as cell operator. Each cellagent is provided with a belief base defined by a set of literals, where beliefs are modeled as predicates in symbolic forms [16], [17] such as

\[
on(cmp-X09-ABB) [\text{source (percept)}]
\]

which denotes that “the component named cmp-X09-ABB is on”. The instruction \( [\text{source (percept)}] \) denotes an annotation meaning that a sensor placed in the environment is the source of the belief. If \( on(cmp-X09-ABB)[\text{source (percept)}] \) is placed in an agent’s belief base, it means that currently the agent believes that the predicate is true. Using theoretical reasoning, inconsistencies can be found through rules, such as

\[
\text{alarm(state,Comp)} :\text{-} \text{off}(\text{Comp}) [\text{source (dms)}] & \text{on}(\text{Comp}) [\text{source (percept)}]
\]

which assigns an alarm if contradictory information has been delivered by distinct sources regarding component status.

States of affair of interest are modeled as achievement goals using the mark “!”. In order to verify whether the agent believes a set of literals, test goals are marked with “?”. For instance, the literal

\[
?\text{voltage_value}(cmp-Y09-ABB,Vmag)
\]

unifies Vmag with the voltage magnitude the agent believes is associated to component cmp-Y09-ABB, while the literal

\[
!\text{open}(\text{neigh_switches})
\]

denotes that the agent has the goal of achieving a state of affair in which it believes \( \text{open}(\text{neigh_switches}) \) is a true statement. The marks “+”/“-” symbolize addition/deletion, either of a belief or a goal.

For instance, let us consider the agent meta plan abstracted to isolate a cell shown in Fig. 2, as usually defined in agent design methodologies. Simplified versions of plans derived from this meta plan are shown below.

Fig. 2: Example of meta plan: cell isolation.
In JASON, these plans model the following reasoning: to handle the addition of the achievement goal corresponding to isolate the cell, if the cell is believed to be de-energized, then pursue the goal of opening neighboring switches.

Similarly to this example, several abstracted meta plans can have a large set of plans covering from simple inferences to complex reasonings. Each agent has then a threaded reasoning cycle with the steps: (i) perceive the environment, (ii) update belief base with perceptions; (iii) receive communications; (iv) select socially acceptable messages; (v) select an event to be handled; (vi) retrieve and determine applicable plans; (vii) select one applicable plan; (viii) select an intention (course of action) and execute one step of the intention.

JASON interpreter includes several inner complex functionalities such as libraries for internal actions, customization options, plan patterns, and so forth. These functionalities have been exploited to model the agent plans of actions.

B. Environment Modeling and Implementation

JASON is provided with an environment infrastructure centralized in a unique object, a strategy which somehow seems not perfectly linked (in modeling terms) with the idea of decentralizing services and resources to support collective and individual activities. Furthermore, in computational modeling of power systems, several objects are not directly affected by agent actions or have attributes to be perceived by agents. This reveals the need for an abstraction beyond the concept of an object in order to model environments that allow testing and validating agent-based solutions in power engineering.

The infrastructure mentioned above has been led by the classical notion of environment used to identify the external world that is perceived and acted upon by agents. There exists, however, a modern view of environment as a first-class abstraction of agent system engineering where to encapsulate services and resources to aid agent activities. This modern view is employed in the agents and artifacts (A&A) meta model of CArTAgO, where artifacts are conceived as general resources and tools to be shared and exploited by agents. From the point of view of designing, artifacts are basic modules to structure and organize environments, providing a general-purpose model to shape functionalities available to agents. On the other hand, from the agent point of view, artifacts are first-class entities structuring a world that the agents can affect in runtime. Artifacts are utilized according to the modeling layers illustrated in Fig. 3.

In the figure, one might observe the explicit separation of agent, artifact and object-oriented modeling layers. Summarily, the agent modeling layer aggregates agents, whilst the object-oriented modeling layer is comprised of computational objects for components (e.g. switch, node, line). In the artifact modeling layer, there are the artifacts which interact directly with the agents and may encapsulate several data in the form of computational objects. Moreover, they provide sets of operations and observable properties to the agents. These operations are computational processes which may be triggered by agents or other artifacts, whilst the observable properties are the variables whose value can be perceived by agents which are observing the artifact.

Artifacts have been defined for the cell components directly involved on agent capabilities utilised in the experiments. Physical components can be acted upon and monitored through objects attributed to the artifacts. The bridge between the modeled artifacts and agents is shown in Fig. 4, where concept mappings are marked through fitted squared end connections. In this bridge, one can verify the roles of agent and environment simulation of JASON and CArTAgO, respectively. Using this bridge, agents can be allowed to consult a manual with the description of artifact properties, create/dispose/link artifacts, use operations, perceive observable events, observe/perceive properties and join/quit workspaces. Observable events/properties are provided as literals (e.g. on(cmp-X09-ABB) [source(percept)]) to belief bases and ex-
ternal actions are performed through artifact operations.

III. LABORATORY EXPERIMENTS

In order to test the application of BDI agents in SYSLAB, a smart grid laboratory designed for testing advanced control and distributed solutions to power systems. SYSLAB is placed at DTU campus and involves 4 interconnected electric sites/buildings. Among other components, SYSLAB facilities includes a 400 V grid with flexible configuration, renewable energy units such as wind turbines and photovoltaic (PV) modules, embedded computing power and flexible communication platform, real-time monitoring, back-to-back converter, controllable loads, vanadium battery and electric vehicles.

Fig. 5 shows the SYSLAB layout with the used configuration marked in red, and the corresponding single-line diagram highlights the components utilised in the experiment. The properties of the components are listed in Table I, where \( P_{\text{min}} \) and \( P_{\text{max}} \) are the minimum and maximum powers, and \( P_0 \) stands as the operating points where applicable. These experiments are related to evaluating the feasibility of managing, with simple reasonings, cell power imbalances using load and battery control, aiming at following tie-line power flow setpoints. In the figure, closed breakers form a connected grid comprising DTU interconnection, 1 PV module (PV319), 1 bank of controllable loads (building 319), 1 PV module (building 117), 1 controllable vanadium battery and 1 additional PV module (PV117). The grid connection between buildings has been achieved by closing breakers C1 and A2. The remaining adjacent breakers have been opened to guarantee isolation from other experiments.

Table I: Properties of Devices Used in the Experiment.

<table>
<thead>
<tr>
<th>Device</th>
<th>Cell</th>
<th>Busbar</th>
<th>( P_{\text{min}} ) (kW)</th>
<th>( P_{\text{max}} ) (kW)</th>
<th>( P_0 ) (kW)</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>Battery</td>
<td>117</td>
<td>117-2</td>
<td>( \approx 15 )</td>
<td>15.0</td>
<td>15.0</td>
<td>Vanadium redox type, 190 kWh</td>
</tr>
<tr>
<td>Cable A2</td>
<td>319</td>
<td>319-2G</td>
<td>319-SC</td>
<td>( G = -117.7 ) S, ( B = -28.6 ) S</td>
<td></td>
<td></td>
</tr>
<tr>
<td>Cable C1</td>
<td>319</td>
<td>319-2G</td>
<td>117</td>
<td>( -2 ) S</td>
<td>Solar PV117, ( G = -7.74 ) S, ( B = -4.46 ) S</td>
<td></td>
</tr>
<tr>
<td>Dumpload</td>
<td>319</td>
<td>319-2G</td>
<td>117</td>
<td>( \approx 0 )</td>
<td>( \approx 78.09 )</td>
<td></td>
</tr>
<tr>
<td>Grid</td>
<td>319</td>
<td>319-2G</td>
<td>0</td>
<td>0.0</td>
<td>200 kVA</td>
<td></td>
</tr>
<tr>
<td>Solar</td>
<td>319</td>
<td>319-2G</td>
<td>0</td>
<td>0</td>
<td>10.1</td>
<td>Orientation ca. 180°, cl. 40°</td>
</tr>
<tr>
<td>Solar</td>
<td>117</td>
<td>117-2</td>
<td>0</td>
<td>0.0</td>
<td>1.0</td>
<td>Orientation ca. 100°, cl. 20°</td>
</tr>
</tbody>
</table>

In Fig. 6, the injected active power (generator convention) is automatically updated through artifact observable properties. If there is a problem during these processes in which (i) the cellagent117 cyclically targets achieving a state of affair related to normal operation. If there is not a literal (or set of literals) in its belief base indicating such state of affair, (iii) it starts the execution of a plan which begins with the unification of auxiliary literals \( I \) and \( E \) with cell imbalance and imbalance error, respectively.

The referred unification is performed by reasoning over literals corresponding to the observable properties of artifacts, namely those related to tie-line power flows, through the process called theoretical reasoning. The imbalance data is then used to alter properties of artifacts in real time, such as load consumption (cell 319) and battery charging (cell 117). In case of identifying any problem during these processes (e.g. communication failure, device failure), the belief base is automatically updated though artifact observable properties and a discussion between neighboring cellagents initiate in order to try solving the problem at least partially.

In Fig. 6, the injected active power (generator convention) is then used to alter properties of artifacts in real time, such as load consumption (cell 319) and battery charging (cell 117). In case of identifying any problem during these processes (e.g. communication failure, device failure), the belief base is automatically updated through artifact observable properties and a discussion between neighboring cellagents initiate in order to try solving the problem at least partially.

Within the frame of this work, buildings 319 and 117 are considered cells and cellagents, named cellagent319 and cellagent117, are assigned to each building. Each cellagent must establish and maintain the operation of its assignee. Among sets of intentions, in order to achieve such target, each cellagent must monitor and control power imbalances of its assignee. If tie-line power flow setpoints are changed, total cell inner imbalance must be changed as well. Hence, actions must be taken aiming to return to a “normal operation” where setpoints are followed adequately.
on 29-Apr-2016 starting at 12:50. Before initialising the agent system, SYSLAB power grid has provided roughly 4 kW to the external grid and a power flow of around 1 kW has been measured in the interconnection from cells 117 to 319. Although solar radiance was nonnegligible during the day of the experiment, PV117 production has been clamped to 1 kW due to a controller malfunction. On the other hand, PV319 production varies from 1.8 kW to 3.7 kW during the day since its inverter maximum power generation has been specified at 5 kW.

The agent system initialisation is marked at [a], where an auxiliary agent corresponding to the Distribution Management System (DMS) sends updated instructions to the cell agents, demanding null power flow at the DTU grid interconnection and 2 kW flowing from cell 319 to cell 117. Using the real-time monitoring of power flows and BDI plans, cellagent117 identifies the necessity of increasing inner production in 1 kW. Following a priority order embedded in external action setResource(I,E), this increase has been ordered to the battery control artifact. In cell 319, a demand increase is required to the controllable load to values of around 5 kW, aiming at absorbing all inner generation from both cells. In [b], the DMS imposed a sudden requirement indicating the inversion of power flow between cells. Each cellagent controlled its cell resources accordingly, where cellagent117 required the battery artifact to charge at 3 kW and cellagent319 ordered load reduction to 1 kW. BDI reasoning operated adequately and DMS requirements have been met, as shown in Fig. 6.

In [c], the DMS required the power flow from cell 117 to cell 319 to return to the value of 2 kW. Setpoint orders have been sent to the resources accordingly. However, after some minutes, the battery flagged failed operation and shut down. This process has been included in the experiment through the battery artifact, where the shut down has been scheduled to occur after a maximum number of inner operations. Additional setpoint orders have been sent by cellagent117 to the battery artifact to confirm the updated belief of failed operation. Concomitantly, cellagent319 has identified an abnormal operation state by analysing tie-line power flows. Since its inner

inbalances have been reasoned to be adequate, a belief has been generated indicating that a problem occurred out of its cell. After confirming the problem, cellagent117 conveys a message to cellagent319 flagging “abnormal operation, inner solution not found”. Hence, cellagent319 confirms message reception indicating that it will take actions to at least nullifying DTU grid power flow. The controllable load is utilised to nullify DTU grid power flow and, some minutes after, the battery returns to normal operation and all setpoints are followed adequately.

V. ACKNOWLEDGEMENTS

The work in this paper has been supported by the European Commission under the FP7 project ELECTRA (grant no: 609687) and INESC P&D Brasil. The authors would like to thank all involved people at the Technical University of Denmark and SYSLAB.
REFERENCES


