

A Deployed Multi-Agent Framework for Distributed Energy Applications

Geoff James¹, Dave Cohen², Robert Dodier², Glenn Platt³, and Doug Palmer¹

¹CSIRO ICT Centre
Locked Bag 17
North Ryde NSW 1670
Australia
+61 2 9325 3276
{first.last}@csiro.au

²Infotility Inc
1327 Spruce St. Suite 201
Boulder CO 80301
USA
+1 720 210 1984
{first}@infotility.com

³CSIRO Energy Technology
PO Box 330
Newcastle NSW 2300
Australia
+61 2 4960 6120
{first.last}@csiro.au

ABSTRACT

In this paper, we describe the adaptation of an open-source multi-agent platform for distributed energy applications and the trial deployment of resource-controller agents. The platform provides real-time, two-way communication and decision making between distributed energy resources – loads and generators – in electricity distribution networks. Such a decentralized architecture improves grid reliability, allows consumers to play a more active role in their energy usage, benefits the network by alleviating the effects of peak wholesale prices and network constraints, and creates new business opportunities in a deregulated market. Agents have been deployed to control hardware at trial locations in Australia, providing a realistic test environment for the platform, and medium-scale trials are anticipated in the near future.

Categories and Subject Descriptors

J.2 [Physical Sciences and Engineering]: Engineering.

General Terms

Algorithms, Management, Design, Experimentation.

Keywords

Applications of autonomous agents and multi-agent systems; collective and emergent agent behavior; frameworks, infrastructures and environments for agent systems.

1. INTRODUCTION

The GridAgents Framework is being developed by Infotility, under contract with the U.S. Department of Energy, and CSIRO. It consists of an adaptive, intelligent agent-based software system which can provide real-time, two-way communication and

decision making between distributed DER system nodes. Distributed Energy Resources (DER) have the potential to provide significant benefits to the electric grid infrastructure and to society as a whole. DERs are typically smaller-scale energy and storage technologies sited close to electrical and thermal loads. DER technologies are increasingly being considered to augment the existing power delivery infrastructure as part of grid reliability programs (emergency response, price response, ancillary services, etc.). They also offer a solution to electricity market participants exposed to peak wholesale prices and constrained network infrastructure. DER has the potential to provide reactive power (generated close to loads) which is highly beneficial for maintaining grid voltage stability. DER combined with next-generation sensing, communication, and control technologies will enable a “path change” by providing the ability to intelligently respond to grid reliability and market signals in a cost-effective manner. The traditional site-specific benefits of DER (efficiency, environmental, local reliability, etc.) can be enhanced by the use of next-generation distributed computing technologies such as those used in this work. Another important aspect of the proposed agent-based software is that it enables the ability to create on-the-fly “aggregate” blocks of capacity for presentation to the energy markets. This functionality along with intelligent islanding and other control concepts has been proposed as various forms of the “microgrid” concept by several research organizations including the DOE. The design is based on a distributed, bottom-up approach to responding to grid contingencies and is intended to supplement the existing power distribution network communication infrastructure.

2. SELECTING A PLATFORM

Agent-based software applications require a foundation to manage basic operations. Various agent toolkits, platforms, toolkits, and methodologies, both proprietary and open-source, were reviewed in order to select a platform to build upon for the GridAgents Framework under development. An agent platform acts as a middleware layer, providing services such as lifecycle management, resource management, messaging, security, directory services, and behaviors. By using such a platform, a programmer is not required to re-invent (and debug) the wheel for each agent project.

Permission to make digital or hard copies of all or part of this work for personal or classroom use is granted without fee provided that copies are not made or distributed for profit or commercial advantage and that copies bear this notice and the full citation on the first page. To copy otherwise, or republish, to post on servers or to redistribute to lists, requires prior specific permission and/or a fee.

AAMAS'06, May 8-12, 2006, Hakodate, Hokkaido, Japan.

Copyright 2006 ACM 1-59593-303-4/06/0005...\$5.00.

The available platforms offer a wide range of programming models and assumptions about locality, communications and agent sophistication. The GridAgents Framework is expected to operate over a large range of hardware platforms and sites. In particular, the CSIRO trial deployments use personal digital assistants (PDAs) as computing resources attached to various loads and generators. The use of PDAs requires an agent platform capable of operating on limited devices, a requirement that quickly eliminates most platforms from consideration. Three agent platforms capable of being deployed on PDAs were evaluated: Cougar [5], Cybele [6], and JADE [7]. In addition, the general-purpose .NET platform [8] was also evaluated as a sanity-check, to ensure that using an agent platform really did result in a reduction in effort.

A set of detailed criteria were developed in order to choose the agent platform as part of the requirements assessment process. The evaluation methodology used was an informal version of the i-Mate process [3]. Although .NET and Cybele were quickly eliminated from consideration, Cougar and JADE provided similar facilities, different implementation models and contrasting strengths and weaknesses. Eventually, JADE was chosen on the grounds of a cleaner design and implementation, better documentation and the ease with which agents could be deployed on limited devices. JADE is by no means perfect; a considerable amount of effort has since gone into architectural issues, such as publish/subscribe notification, agent discovery and deployment. However, JADE has stood up as the “least-worst” choice.

3. ARCHITECTURE AND MESSAGING

DER agents appear to each other to fulfill certain functions which we have termed “roles”. In this system, a role is the ability to send and receive a certain fixed set of message types. The means by which an agent determines its message content is invisible to other agents, and other agents know only the allowed message types. The roles include buyer, seller, bulletin board, physical device, and others. An agent which fulfills the seller role will respond to a request for a price function or a request to conclude a transaction. An agent which fulfills the physical device role will respond to requests for status information or a request to set the device state. As other agents see only these functions, the DER agent system is a “plug and play” system, in which an agent enjoys broad latitude within which it can implement the role functions in various ways, so long as all functions for a given role are implemented.

An agent may fulfill more than one role. As agents interact only through the role interfaces, a new type of agent can coexist happily with others so long as the role interfaces are implemented appropriately. There is no need to restart the whole agent system when a new agent type comes on-line. This is an especially important consideration in systems comprising multiple hosts, as it may be impractical to coordinate the operations of different hosts. It is feasible, though perhaps somewhat involved, to create agents which are not JADE agents, by constructing FIPA messages of appropriate types and contents. At this time we have no plan to implement non-JADE agents, but we see no need to rule out the possibility.

Data persistence is accomplished in the DER agent system through the bulletin board role. A bulletin board is just a repository of messages organized according to a label, which is

called the topic. Agents can query the bulletin board by topic, receiving the messages under that topic in response. Aside from messages exchanged through role interfaces, public information in the DER agent system is published via a bulletin board. The types of information available may include prices, parameters, notices, and measurements. It is anticipated that the contents of a bulletin board may be stored in a conventional database, or published in a format accessible outside the DER agent system, such as an XML-based syndication format.

4. APPLICATIONS AND ALGORITHMS

Whether the application is grid security or services for electricity market participants, the fundamental algorithmic requirement of a DER multi-agent network is the provision of aggregated capacity through load shifting, load shedding, or distributed generation by a multiplicity of resources. Aggregated capacity is characterized by a quantity of kW or MW and also time and timeliness, reliability or “firmness”, and traceability to a known set of customers. Note that reliability may be represented as a profile rather than a single value: 20 MW at 99.99% firmness, 40 MW at 99% firmness, and 70 MW at 90% firmness would be an example. This is enough information for a user to make an informed risk calculation. At the resource agent level the ability to shift or curtail load or dispatch local generation requires planning ahead, with a planning horizon long enough to account for time constants inherent in the loads and generators.

Each agent in the DER agent system attempts to choose among available alternatives according to the criterion of minimum operating cost and subject to constraints. This local optimization may lead to a solution with identifiable global properties, such as a Nash equilibrium, although we have not proved any such properties yet. We assume that each agent has a planning and execution cycle, which may be regular or irregular, and coordinated or uncoordinated with other agents. The result of the planning phase is a series of actions to be carried out in the future. In the execution phase, the agent carries out the plan by executing commands to a physical device, or perhaps issuing commands to subordinate agents which result in a command to a device.

The planning problem is generically phrased as a constrained optimization problem. The appropriate method for solving such problems varies according to the form of the cost function, and thus agents may employ various techniques for planning. Agents which aggregate or broker the output of other agents may employ integer linear programming to solve their planning problems, as their costs are the sum of the prices stated by their subordinate agents and their resources are the sum of the resources supplied by their subordinates. Aggregating agents may also employ genetic optimization [1] to optimize subsidiary agents for soft-real-time application.

As an exact solution to an optimization problem may be very time-consuming, we are also developing methods which provide “any-time” solutions based on selfish agents seeking to optimize local cost functions. Although the resulting global solution may be suboptimal, this may be a worthwhile sacrifice if the net result of the selfish agents’ actions – their emergent behavior – converges to a useful solution in a short time.

Planning algorithms present a scalability problem: there are many free parameters for each resource, combinatorial optimisation of any kind tests computational resources quite severely, and our aim

is to coordinate many thousands of agents. Decentralised algorithms are therefore of great interest because they use the multi-agent network as a parallel computer and thereby promise better scaling properties provided that messaging overheads do not defeat this purpose. A decentralised algorithm does not yet exist for planning as we've defined it, although a candidate with many of the capabilities we require is the market-based algorithm of Kok et al. [2], and other algorithms under development will be reported in the near future. All candidate planning algorithms presume a defined group of agents that should plan together and a defined connectivity between agents in the group.

The approach we are taking is to construct a meta-algorithm that governs the groups of agents within which planning is used. Decentralised clustering is an example of a multi-agent algorithm that creates groups, and it has been shown to scale better than an equivalent centralised algorithm [4]. We are developing an adaptation of decentralised clustering to govern the formation of planning groups of energy agents and the topology of messaging connections within each group. The crucial requirement is to interleave the two levels of algorithm so that time scales are compatible and sufficiently short for the system responses desired.

5. DEPLOYMENT

To test the behavior of various multi-agent system approaches and showcase our technology, we have established a comprehensive experimentation and testing hardware platform. The platform is made up of a variety of loads and generators, configured for agent control, and subject to real-world constraints:

- Two cool rooms typical of a small business setting.
- An heating, ventilation and air-conditioning (HVAC) system in an office building.
- A 30 kW gas turbine or "microturbine".
- A large photovoltaic (PV) system.
- A small wind turbine and solar-tracking PV array, 1 kW each, combined with battery storage.

The interacting set of loads and generators are electrically connected to form a microgrid, their controlling agents being designed to coordinate demand and supply while reacting intelligently to national electricity market price signals. Agents controlling loads and generators are run on personal digital assistants (PDAs) while personal computers (PCs) are used for implementing more computationally intensive agent roles:

- A web scavenging agent which obtains current and forecast electricity market prices and weather data.
- A snapshot agent which monitors various real-time parameters of the agent system.
- An aggregation agent which offers a price schedule for resources from subordinate agents.
- A manager agent which directs the operation of a number of loads and sources.

The hardware platform has a comprehensive Internet Protocol (IP) based infrastructure for inter-agent communications. This infrastructure uses a variety of physical layer technologies, including Ethernet and Wireless LAN mechanisms, with Bluetooth and cable based connections used for serial communications to distributed energy devices.

Throughout this implementation work, and the subsequent experimental phase, a number of challenges and their solutions have been encountered. We believe many of these challenges would exist in any geographically distributed multi-agent system implemented in common computing and communications hardware. Space does not allow detail here, but the authors are happy to be contacted about such issues as agent communication, firewalls, network addressing, processing overheads, unique and diverse hardware installations, reliable and fail-safe behavior, and message content encoding.

6. CONCLUSIONS

We have developed the GridAgents Framework for managing distributed energy resources – customer loads and generators – by adapting the JADE multi-agent platform. Several aspects peculiar to our application are described in this paper, many of them based on experience of deployment on distributed processors controlling typical small-business loads and generators. We have developed algorithms to aggregate capacity and provide a distributed energy service to meet network and market goals. This work continues.

Future work will include advanced demonstrations of the GridAgents Framework for Microgrid applications, automated demand response applications, and electric distribution network control applications. A "smart grid" project is planned for 2006 with a large U.S. utility for embedded intelligence and control for compact networks. In Australia our next goal, in collaboration with utility, hardware, and telecommunications partners, is a moderate-scale deployment of the order of 1,000 agents.

7. ACKNOWLEDGMENTS

Everyone in the development and deployment team made an important contribution to the results reported here. We gratefully acknowledge the work of Matt Blyde, Matthew Chadwick, Yan Kitsis, Marteen Rojo, Peter Wang, and Sam West.

8. REFERENCES

- [1] Guo, Y., Li, J., and James, G. Evolutionary Optimisation of Distributed Energy Resources. *Proceedings, 18th Australian joint conference on Artificial Intelligence (AI'05)*, Sydney, December 2005.
- [2] Kok, J.K., Warmer, C.J., and Kamphuis, I.G. PowerMatcher: Multiagent control in the electricity infrastructure. *Proceedings, 4th international joint conference on Autonomous Agents and Multi Agent Systems (AAMAS'05)*, Utrecht, July 2005.
- [3] Liu, A. and Gorton, I. Accelerating COTS Middleware Acquisition: The i-Mate Process. *IEEE Software*, 20, 2 (2003), 72-79.
- [4] Ogston, E., Overeinder, B., van Steen, M., and Brazier, F. A Method for Decentralized Clustering in Large Multi-Agent Systems. *Proceedings, 2nd international joint conference on Autonomous Agents and Multi Agent Systems (AAMAS'03)*, Melbourne, July 2003.
- [5] <http://www.cougaar.org/>
- [6] <http://www.i-a-i.com>
- [7] <http://jade.tilab.com/>
- [8] <http://www.microsoft.com/net>