Multiagent System Solutions for Distributed Computing, Communications, and Data Integration Needs in the Power Industry

*Zhong Zhang *zzkerry@iastate.edu*
*James D. McCalley *jdm@iastate.edu*
§Vijay Vishwanathan *vijayav@ureach.com*
*Vasant Honavar *honavar@iastate.edu*

*Iowa State University, Ames IA USA §Open Systems International, Minneapolis, MN USA

Abstract-- **We identify three fundamental issues underlying many problems in power systems today: distributed computing, communications, and data integration. We review the characteristics of intelligent agents and multi-agent systems (MAS) technologies and argue that MAS offer a modular, extensible, flexible, and integrated approach to address all three of these issues and the problems resulting from them. The MAS design methodology is summarized, and an illustrative MAS application scenario in electric power systems is presented.**

Keywords--**Multiagent systems, distributed computing, communication, data and information, design methodology, negotiated decision-making**.

I. Introduction

A system is centralized if its components are restricted to one site, decentralized if its components are at different sites with no or limited coordination, and distributed if its components are relatively autonomous entities which work together to achieve some overall objective [1]. The notion of autonomous components and coordination are the basic ingredients of any distributed systems. A multiagent system is a distributed system consisting of multiple software agents, which form "a loosely coupled network, called a multiagent system (MAS), to work together to solve problems that are beyond their individual capabilities or knowledge of each entity" [2]. Of particular interest are MAS in which the individual agents display significant intelligence and autonomy. In the past few years, MAS technologies have found applications in many distributed systems such as distributed problem solving, distributed information fusion, and distributed scientific computing [3,4,5]. Electric power systems, being geographically distributed but with fairly extensive communication networks, is a good application domain for innovative distributed solutions. Recently, we have been investigating MAS based approaches to power systems problems [6,7,8] and see significant potential in them for contributing to power systems problems in distributed computing, communications, and data integration.

This paper is organized as follows: Section II describes characteristics of various traditional distributed computing techniques. Section III introduces communication and information integration problems encountered in power systems and current industry efforts to solve them. Section IV argues that MAS is well suited as an integrated solution approach to these problems. Section V presents a MAS design methodology. Section VI describes a prototypical MAS solution to a power systems problem. Section VII concludes.

II. Distributed Computing

Distributed computing refers to computing that involves multiple loosely coupled processors working together to solve an overall problem [9]. Distributed computing offers a natural approach to solving complex data and computation intensive problems that arise in power systems analysis and control. Many approaches to distributed computing have been developed over the past decades. These include *Socket Programming*, *Remote Procedure Calls (RPC)*, object-oriented *DCE*, *DCOM*, *CORBA*, Java *RMI*, and *Message-Oriented Middleware (MOM)* [9]. These approaches have their relative advantages and disadvantages [9]. Some major limitations with respect to the design of robust, networked applications that involve many autonomous, heterogeneous entities are: (1). The interactions among participating entities are fixed a-priori through explicitly coded instructions by the application developer. As a result, they lack run-time adaptive behavior. (2). Ongoing interaction requires ongoing communication, making them unsuitable for applications that have to operate in environments where maintaining continuous communication is expensive or infeasible and connections are unreliable. These considerations have motivated the development of approaches to distributed computing based on agents which provide ways to maintain ongoing interaction without ongoing communication [10]. MAS and related technologies offer an attractive paradigm for design of distributed networked applications that involve many relatively autonomous, heterogeneous entities [5].

III. Communication And Data Integration

The emerging presence of computer and digital technologies has brought much greater efficiency and operational potential to the electric utilities. However, as both hardware and software vendors design systems suited to their own specific applications, the number of data storage platforms/formats and corresponding access/retrieval/interface methods have burgeoned. This has resulted in numerous and heterogeneous "information islands" at different levels throughout the power system, which are labor-intensive to identify, access, and integrate for a given purpose. In response, the power community has begun efforts to standardize communication protocols and information/data storage.

A. Utility Communications Architecture (UCA)

The *Utility Communications Architecture (UCA)* [11,12] was developed under the sponsorship of the Electric Power Research Institute (EPRI) through a process of broad industry involvement. The objective has been to allow for seamless integration across the utility enterprise using off-the-shelf international standards to reduce costs. UCA is a flexible and scaleable architecture that provides communications solutions from simple devices to control centers all based upon compatible, standard and interoperable communications protocols and device object models. The UCA protocols are organized according to the *Open Systems Interconnection* (OSI) reference model. The UCA Version 2.0 includes profiles employing protocols from both the OSI and TCP/IP families of protocols. The *Inter-Control Center Communications Protocol* (ICCP, also known as Telecontrol Application Service Element 2, TASE.2) defines a standardized use of *Manufacturing Message Specification* (MMS) in UCA Version 2.0 compliant networks for real-time exchange of data within and between control centers, power plants, and SCADA masters. The *Generic Object Models for Substation and Feeder Equipment* (GOMSFE) contains detailed object models of common field devices, including definitions of their associated algorithms and communications behavior visible through the communication system. The device models developed within the UCA 2.0 effort make use of a common set of services to describe the communications behavior of the devices. A standard mapping of these services onto the UCA application layer protocol (MMS), when used in conjunction with the device models, completely specifies the detailed interoperable structure for utility field devices. The services and mapping to MMS are defined in UCA *Common Application Service Models* (CASM). CASM is the document that illustrates the step-by-step processes that must be followed for a communications service to be performed within UCA.

B. Common Information Model (CIM)

Electric utility organizations have long needed to exchange system information with one another in order to construct simulation environments for power system economics and security analysis. Even though most Energy Management Systems (EMS) and Distribution Management Systems (DMS) are now supplied with standard operating systems on standard computer platform hardware, these systems are still built on proprietary databases. The consequences of this led to boundaries between different EMS systems and locked the user out of the environment.

Between 1993-1994, the Electric Power Research Institute (EPRI) Working Group on Control Center Application Interfaces (CCAPI) developed objectives to publish a set of guidelines for application interfaces, to develop associated support tools, and to promote the use of open software engineering approaches in EMS. The *Common Information Model (CIM)* [13] is the foundation of the overall CCAPI framework. The CIM provides a standard for representing power system objects along with their attributes and relationships. The CIM is partitioned into a number of submodels, or packages, for convenience: a Wires Model, SCADA Model, Load Model, Energy Scheduling Model and a Generation Model. The CIM facilitates the integration of EMS applications developed by different vendors; entire EMS systems developed by different vendors; or EMS systems and other systems concerned with different aspects of power system operations, such as generation or distribution management.

C. Data Integration Needs for Asset Management

Recently there has been a great deal of investment in developing asset management tools. These tools may be classified by function. There are several which provide work-flow functions, work-order tracking, and data storage. Examples of these tools are Maximo [14], Cascade [15], and Asset-Sentry [16]. Typical data stored includes equipment data (nameplate, maintenance histories, and condition data). Some companies have several additional data repositories that house such information as outage schedules, operating histories (e.g., a process-information or PI-historian), and equipment-specific condition data (e.g., dissolved gas analysis results, tap changer temperatures, etc.). Because of the number and diversity of the asset management data repositories, EPRI has developed the maintenance management workstation (MMW) that acts as a database integrator providing a number of functionalities among which is the ability to bring data from multiple sources to a consolidated data set.

D. An alternative and unifying approach

UCA, CIM, and the asset management tools represent current efforts to facilitate communication needs and information processing needs within an information-intensive industry. An underlying, common theme is to *standardize* and *centralize* by defining and utilizing standard, interoperable communication protocols, by providing common object-models of power system data items, and by aggregating data into warehouses such as MMW. Therefore, the focus has been on aggregation of the data itself. MAS represent an alternative where the focus is on the processing rather than on the data, leaving the data both heterogeneous and distributed. We argue that MAS represents within a single technology a unified solution to the problems that drive the need for UCA, CIM, and many of the various asset management tools.

IV. Potential of Multiagent Systems

Multiagent systems have proven to be an effective paradigm in a number of distributed networked applications that require information integration from multiple heterogeneous autonomous entities [5, 17]. More recently, MAS have begun to emerge as an integrated solution approach to distributed computing, communication, and data integration needs for deregulated power systems.

A. Agent-based Computing and Agent-oriented Programming

A multiagent system consisting of multiple agents can take advantage of computational resources and capabilities that are distributed across a network of interconnected entities. An agent-based approach allows the creation of systems that are flexible, robust, and can adapt to the environment. This is especially helpful when components of the system are not known in advance, change over time, and are highly heterogeneous. Agent-based computing offers the ability to decentralize computing solutions by incorporating autonomy and intelligence into cooperative, distributed applications. Each agent *perceives* (the state of its environment), *infers* (updates its internal knowledge according to the newly received perceptions), *decides* (on an action), and *acts* (to change the state of the environment). Agent-oriented programming (AOP) is the software paradigm used to facilitate agent-based computing and extends from object-oriented programming (OOP) by replacing the notions of class and inheritance with the notions of roles and messages, respectively [18].

B. Knowledge-level Communication Capability

Within a multiagent system, agents can communicate with each other using agent communication languages (ACLs), which resemble human-like speech actions more than typical symbol-level program-to-program communication protocols. This capability enables agents to distill useful knowledge from voluminous heterogeneous information sources and communicate with each other on the basis of which they coordinate their actions. By enabling performance of computation where computing resources and data are located, and allowing for flexible communication of relevant results to relevant entities as needed, MAS offer significant new capabilities to power systems, which have for so long depended on various forms of expensive telemetry to satisfy most communication needs.

C. Distributed Data Access and Processing

Another benefit offered by MAS is the distribution of agents across a network. Software agents may have different levels of intelligence, ranging from data agents and functional agents to decision agents (corresponding to what were termed data view, function view, and dynamics view in [18]). Special agents can be designed to capture heterogeneous and proprietary information/data, such as the application described in [19]. This application uses a three-layer architecture consisting of the physical layer, the ontological layer, and the user-interface layer. The physical layer allows the system to communicate with the distributed information sources. It is based on a federated database architecture. The ontological layer automatically bridges the syntactic and semantic mismatches among the heterogeneous data sources. Finally, the user interface layer enables users to interact with the system, define ontologies, post queries and receive answers. Because each agent is designed to perform a specific role, with associated knowledge and skills, distributed and heterogeneous information may be efficiently assimilated locally and utilized in a coordinated fashion in distributed knowledge networks [5], resulting in reduced information processing time and network bandwidth in comparison to that of more traditional centralized schemes.

D. Integratability

The industry maintains a rich plethora of power system software applications, developed in many different computer languages, intended for use on many different platforms. Extending old applications or developing new ones usually involves integrating legacy systems, and doing so is cumbersome and labor-intensive. This problem is largely overcome by encapsulating legacy systems into autonomous agents for interoperability within a larger infrastructure [5].

E. Distributed Decision Support

MAS also offers a powerful task decomposition approach to problem solving through interaction among agents. This is facilitated by the ability of different agents to coordinate behavior through cooperation (agents have established and mutually agreeable objectives), negotiation (agents negotiate until agreement is reached), or mediation (agents resolve conflicts that cannot be resolved by negotiation by appeal to a third, neutral agent) [18]. Each of these coordination mechanisms finds ubiquitous application in power systems. We investigated the use of mechanisms and applications for negotiated decision-making [7,8], where autonomous distributed agents seek to achieve global objectives that are consistent with individual goals. A multiagent negotiation system [8,20] was built in which software agents, armed with coded negotiation models, represent different decision-makers, and conflict resolution is achieved via inter-agent message exchange until agreement is reached. This negotiation system not only provides the technology necessary to facilitate actual negotiation scenarios but also provides the ability to study the effects of varying degrees of decentralization in decision problems by comparing solutions obtained assuming full information and centralized optimization to solutions obtained based on sequential, bilateral negotiations.

V. Multiagent System Methodology

Several MAS paradigms and methodologies have been proposed in the literature, e.g. MASSIVE [18], DESIRE [21], Gaia [22] and MaSE [23], based on different notions of agents and multi-agent organizations. Our group uses a 4-stage methodology for constructing MAS for power systems applications: *Analysis*, *Design*, *Implementation*, *and Deployment*.

A. Analysis: environment and tasks

This is the first stage which identifies the application domain, overall problem, objectives, MAS application environment, i.e., information that will be available to an agent, actions required of the agents, and operational (e.g., security) and performance constraints. Task decomposition is performed to determine what the system is supposed to do (and not how it is supposed to do it) to achieve overall MAS objectives.

B. Design: roles, interactions, and organizations

Having decomposed the problem into constituent tasks, the next stage is to identify the agents required to effectively perform the tasks in terms of (a) definition of agent roles (data, functional, decision, mediator, facilitator, etc.) linking domain-dependent application features to appropriate agent technology, and specifying services to be associated with each agent; (b) identifying the types of interactions needed between different agents in order to achieve individual or joint goals; and (c) specifying the organization of the different agents in terms of a society of agents that is consistent with the various defined roles and that achieves the overall objectives.

C. Implementation: architecture

A key requirement for implementing a MAS is the selection of system and agent architectures. System architecture includes such aspects as multi-agent organization (e.g., hierarchical versus flat), agent management, and coordination mechanisms, including such things as directory services (or yellow pages) that enable each agent to know the capabilities of other agents, and the Agent Communication Language (ACL) that provides the common basis for inter-agent communication. The most common ACLs include Knowledge Query and Manipulation Language (KQML) [24] and Foundation for Intelligent Physical Agents (FIPA) ACL [25]. There are a number of available agent platforms for implementing MAS including Voyager [26], Concordia [27], Aglets [28], SMART [29] among others. Based on an agent platform, individual agents can be extended with abilities to process specific messages and communicate with other agents. In order to enable inter-agent communication, besides ACL, it is also essential to define an appropriate ontology, or vocabulary, for the MAS that specify all possible message contents. In addition, some kind of inter-agent coordination strategy must be in place.

A broad range of architectures for agents (including reactive, deliberative, adaptive, communicative) have been studied. Properties that distinguish the various agent architectures include reasoning capabilities, resource limitations, control flow, knowledge handling, autonomy, user interaction, temporal context, and decision making.

D. Deployment

Here, actual agents are instantiated to cooperatively solve the problem. Testing is done to validate the model.

VI. An Illustrative Application in Power Systems

Based on the above MAS methodology, we have developed a Java™-based software infrastructure, *MASPower*, for instantiation of *generic agents* capable of persistent interaction with environment, inter-agent communication, task management, and accessing local and remote information sources. The distributed computing components of MASPower are engineered using Voyager ORB™ [26]. We have used MASPower to explore a prototypical power systems application of MAS in electric equipment condition monitoring and maintenance scheduling. The resulting Multiagent based Condition Monitoring and Maintenance System (MCMMS) for power transformers is shown in Fig. 1.

Large amounts of equipment monitoring data are gathered by monitoring equipment, operational hardware, software systems and databases that are not easily accessed or generally available. Intelligent communication agents, capable of accessing distributed, heterogeneous, proprietary data sources, can extract all related transformer condition monitoring information and communicate with diagnostic agents. Diagnostic agents possess knowledge of the necessary monitoring techniques. Based on the queried monitoring data, diagnostic agents can cooperatively perform diagnostic functions. Because monitoring systems continuously collect real-time data, the amount of data is enormous, and the diagnosis can be data and computation intensive. MAS architecture enables diagnostic agents to cooperate to detect abnormal situations and identify possible transformer failure modes. Once certain predefined operating thresholds have been violated, the alarm agent alerts the operating personnel at a central control room. Based on the diagnosis, maintenance agents recommend appropriate maintenance tasks for each piece of equipment. Then we schedule these tasks subject to constraints on economic resources, available maintenance crews, and restricted time intervals. We have developed a system-wide centralized maintenance scheduling optimization procedure by maximizing cumulative system risk reduction [30]. However, with recent organizational disaggregation and functional balkanization in the industry, facility ownership is heavily fragmented, and information access and decision-making authority is quite limited for any one particular organization. Decision problems, such as maintenance scheduling, once solved using centralized optimization are now more difficult due to distributed information and the multiplicity of competing stakeholders. We are currently interested in using multiagent negotiated decision-making to solve maintenance scheduling optimization problem. Different maintenance software agents represent different independent utilities. Each maintenance agent performs the centralized maintenance scheduling optimization in its own territory first. Conflicts among these maintenance schedules are then resolved via inter-agent negotiations according to the overall system security constraints imposed by the ISO-Agent. The maintenance schedule obtained through agent negotiations satisfies all involved parties. Results are illustrated by comparing maintenance schedules, risk reduction, and resource allocation between the centralized solution using a single optimization and the solution achieved using MAS-based negotiations.

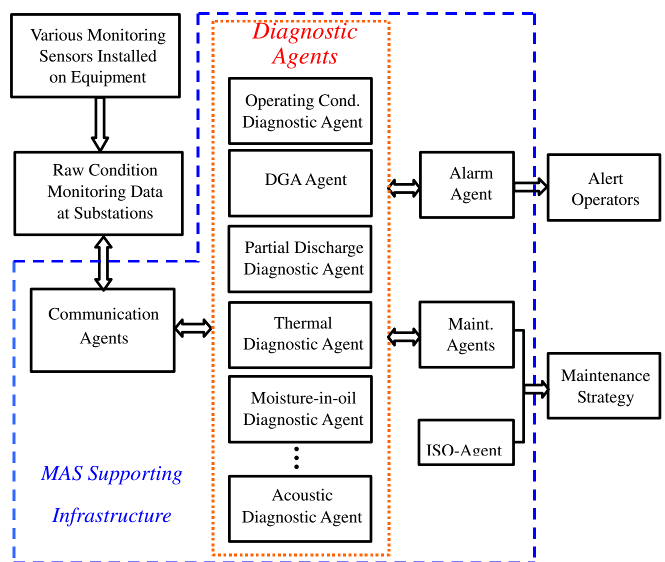

Fig. 1: Multiagent based Condition Monitoring and Maintenance System

VII. Conclusions

This paper suggests that intelligent software agent and multiagent systems technologies offer an integrated approach to the design and implementation of data/information management and distributed decision support infrastructure for deregulated electric power systems. As such, it represents an alternative solution approach, or at least a way to implement certain types of solutions, for power system problems related to distributed computing, communications, and data integration. A MAS design methodology has been presented, and an illustrative application in power systems has been briefly described. We conclude that multiagent systems technology has significant potential for facilitating the complex information processing and decision-making problems inherent to deregulated electric power systems.

VIII. Acknowledgements

This research has been supported by the EPRI and Department of Defense (DoD) through the Complex Interactive Systems/Networks initiative (Grant WO8333-01), National Science Foundation (Grant 0087152, 0219699) and Power Systems Engineering Research Center (PSerc).

IX. References

[1] Amjad Umar, *Distributed Computing: A Practical Synthesis*, Prentice-Hall, Inc. Englewood Cliffs, New Jersey, 1993, Page 5.
[2] Durfee, E. H. & Montogomery, T. A., "MICE: A Flexible Testbed for Intelligent Coordination Experiments," *Proceedings of the 1989 Distributed Artificial Intelligence Workshop*, pp. 25-40, 1989.
[3] T. Drashansky, E. N. Houstis, N. Ramakrishnan, J. R. Rice, "Networked agents for scientific computing," *Communications of the ACM,* vol. 42, NO. 3, March 1999, pp. 48-53.
[4] R. Khosla, T. Dillon, "Intelligent hybrid multi-agent architecture for engineering complex systems," *Proceedings of the 1997 IEEE international Conference on Neural Networks,* vol. 4, pp. 2449-2454.
[5] Honavar, V., Miller, L., and Wong, J. "Distributed Knowledge Networks" In: Proceedings of the IEEE Information Technology Conference. 1998.
[6] V. Vishwanathan, V. Ganugula, J. McCalley, and V. Honavar, "A multiagent systems approach for managing dynamic information and decisions in competitive electric power systems," In *Proc. of the 2000 North American Power Symposium*, Oct. 2000, Waterloo, Ontario.
[7] V. Vishwanathan, J. McCalley, and V. Honavar, "A Multiagent Infrastructure and Negotiation Framework for Electric Power Systems," Power Tech Proc., 2001 IEEE Porto, Volume: 1, 2001 Page(s): 438–443.
[8] Z. Zhang, V. Vishwanathan, J. McCalley, and V. Honavar, "A Multiagent Security Economy Decision Support Infrastructure for Deregulated Electric Power Systems", Proc. of 7th Probabilistic Methods Applied to Power Systems (PMAPS), Vol. 1, Page(s): 39-44, Naples, Italy, Sep. 2002.
[9] Van Steen, M. and Tanenbaum, A., *Distributed Systems: Principles and Paradigms*. Englewood Cliffs, NJ: Prentice Hall (2002).
[10] White, J.E. (1997). Mobile Agents. In: Bradshaw, J.M. (ed.), *Software Agents*, Cambridge, MA: MIT Press.
[11] Fundamentals of Utilities Communication Architecture, IEEE Computer Applications in Power, Volume: 14 Issue: 3, July 2001, Page(s): 15 –21
[12] Introduction to UCA Version 2.0, EPRI, September 1998.
[13] Lee, S.T., "The EPRI common information model for operation and planning", Power Engineering Society Summer Meeting, 1999. IEEE, Volume: 2, 1999, Page(s): 866 -871 vol.2.
[14] http://www.mro.com/corporate/products/maximo.htm
[15] CASCADE project: http://www.sis.pitt.edu/~cascade/
[16] ABB Asset Sentry, http://www.abb.com
[17] Caragea, D., Silvescu, A., and Honavar, V. (2001). Invited Chapter. Towards a Theoretical Framework for Analysis and Synthesis of Agents That Learn from Distributed Dynamic Data Sources. In: *Emerging Neural Architectures Based on Neuroscience*. Berlin: Springer-Verlag.
[18] J. Lind, "Iterative software engineering for multiagent systems – The MASSIVE Method," in "Lecture notes in Artificial Intelligence," Springer-Verlag, Berlin, 2001.
[19] Jaime A. Reinoso-Castillo, "Ontology-driven information extraction and integration from heterogeneous distributed autonomous data sources: A federated query centric approach", M.S. thesis, Department of Computer Science, Iowa State Univ., Ames, 2002.
[20] J. McCalley, Z. Zhong, V. Vishwanathan, and V. Honavar, "Multiagent negotiation models for power system applications," in "Autonomous systems and intelligent agents in power system control and operation," C. Rehtanz, editor, Springer-Verlag, Berlin, 2003, pp. 49-74.
[21] F. M. T. Brazier, B. M. Dunin-Keplicz, N. R. Jennings, J. Treur, "DESIRE: Modelling Multi-Agent Systems In a Compositional Formal Framework", International Journal of Cooperative Information Systems, 6(1), pp 69-94, 1997.
[22] M. Wooldridge, N. R. Jennings, D. Kinny, "The Gaia Methodology for Agent-Oriented Analysis and Design", Journal of Autonomous Agents and Multi-Agent System, 3(3), pp 285-312, 2000.
[23] M. F. Wood, S. A. DeLoach, "An Overview of the Multiagent Systems Engineering Methodology", *in Proc. Of the First Int. Workshop on Agent-Oriented Software Engineering*, 2000, pp 207 – 221.
[24] M. R. Genereth, S. P. Ketchpel, "Software Agents", Communications of the ACM, 37 (7), pp 48-53, 1994.
[25] FIPA ACL Specifications. Online available at: http://www.fipa.org/repository/index.html.
[26] http://www.recursionsw.com/products/voyager/voyager.asp.
[27] http://www.meitca.com/HSL/Projects/Concordia/.
[28] http://www.trl.ibm.co.jp/aglets/.
[29] Wong, J., Helmer, G., Naganathan, V. Polavarapu, S., Honavar, V., and Miller, L. (2001) SMART Mobile Agent Facility. *Journal of Systems and Software*. Vol. 56. pp. 9-22.
[30] Yong Jiang, Zhong Zhang, J. D. McCalley and Tim Van Voorhis, "Risk-based Maintenance Optimization For Transmission Equipment," under review by IEEE Transactions on Power Systems, November, 2003, available from the authors.