Design and Implementation of Multi-Agent System for Analysis of Electrical Power Networks

Miroslav Prýmek and Aleš Horák

Faculty of Informatics, Masaryk University Brno
Botanická 68a, 602 00 Brno, Czech Republic
E-mail: \{xprymek,hales\}@fi.muni.cz

Abstract. In this paper, we describe the first stages of implementation of a multi-agent system for simulation and analysis of an electrical power network.

The communication among the defined agents is based on standards in multi-agent systems – the communication protocols CORBA (Common Object Request Broker Architecture) and KQML (Knowledge Query and Manipulation Language). The system itself economizes the open source implementation of the protocols.

The whole system is able to perform an active simulation of the energy flow in a power system and its visualization. Even at the beginnings of the implementation process, we take into account possible future replacement of any particular agent with an on-line power equipment monitoring facility (ad-hoc sensors), which allows to monitor the whole power system in real time.

1 Introduction

The fault-tolerance and quality assurance of the power supply is very important and constantly evolving science branch which demands intensive research. Because of the geographic dissipation and big financial cost of the electrical power systems (EPS) facilities maintenance, the required reliability cannot be reached simply by redundancy. It is necessary to search for the new low-cost but effective means of the required reliability assurance.

One of the possible approaches to this problem is regular control of the facility operation. Failure data are gathered in failure databases with a precise description of the failure cause, time, severity and the description of the failed component – its manufacturer, model, serial number etc. This effort allows a statistical analysis of the reliability of the particular facility series, types etc.

This type of the fault database is for the whole Czech republic built and maintained by the research team for creation and categorization of failures records for

* This work has been partly supported by Grant Agency of the Academy of Sciences under the project 1ET400300414.
distribution equipment and outages of supply at all voltage levels in the Technical University of Ostrava. Utilities for the analysis of the data in this database are also developed there. The aim of this effort is to develop a new methodology for the power system facility maintenance based on the condition-centered approach rather than the contemporary used preventive time-based approach. Power system facilities are well tested in the process of their design and manufacturing but there’s a serious lack of testing in the process of their real operation. This project tries to overcome this lack.

The new methodology should be enough flexible to allow a fast and effective control even for facilities that lack measured reliability data. The Technical University of Ostrava with the help of other universities in the Czech Republic has made an interdisciplinary team which is focused on the development of new methodology for the power system facility condition checks, outage prediction, black-out risk reduction and early post-fault recovering.

The team at FI MU Brno is focused on the usage of the multi-agent systems in the power system facility simulation, monitoring and control.

2 System architecture

The design of our multi-agent framework is based on the following basic requirements:

1. decentralization
2. platform independence
3. performance scaling
4. modularity and extensibility
5. open standards
6. security
7. low-cost practical application

After the specification of all the system requirements (see [1]), we have decided that the best results will be acquired by implementing the system as a multi-agent system based on the standard technologies commonly used in this area – a combination of two communication protocols, CORBA and KQML. CORBA standard has freely available specification and there are many implementations – proprietary ones (e.g. Visibroker, PeerLogic, IONA) and even free ones (e.g. OmniOrb, Orbit, Mico). In our prototype implementation, we use OmniOrb (in Python) and Sun ORB which is part of the Java SDK. An indispensable advantage of CORBA standard is that all ORB implementations conforming to the specification should integrate well within the system.

Usage of the CORBA-KQML solution was many times published in the literature (e.g. [4], [5], [6]). The multi-agent approach has been used even in the area of the power systems monitoring and control (e.g. [7], [8], [9]).

As discussed in [8], old “hardwired” solutions of the power systems monitoring have many problems caused by their limited flexibility which we are going to
overcome by using the open and extensible multi-agent system implementation. This approach will be strong enough to satisfy all our requirements.

Our proposed communication architecture is made up of many autonomous and self-deciding agents (see the Figure 1).

![System architecture](image)

Fig. 1. System architecture

### 2.1 General agent characteristics

A general description of an agent-based software engineering methods, that form the basis of the presented system, can be found e.g. in [2]. Case studies of actual applications with agent-based architecture in the industrial control systems are presented in [3].

The main principle of our system is that each agent can be implemented by a stand-alone process or even stand-alone computer in a computer network. If there is a huge amount of communication between agents, it is possible to implement two or more agents within one process and thus lower the communication overhead.

All agents in the system have the ability to communicate through CORBA and KQML (see the Section 3). Each agent has a defined type (denoting a set of messages which are accepted and understood by the agent) and it is identified within the network by its distinct name and a given identification number. The agent types are implemented as a hierarchy in which each level is assigned a set of mandatory KQML messages which every agent of this type must understand and must be able to respond to it (this is similar to the inheritance in object programming languages). The hierarchy looks like that:

- Agent
  - Real-life agent (represents a particular power system facility)
  - Line
Each agent that belongs to some category must be able to respond to every message from the defined set of messages. For example, agent of the type Source must be able to respond to all messages mandatory for types Source, Facility, Real-life agent and Agent.

3 Agents communication

The basic assumption is that agents can be located on separate machines or at least separate processes within one machine. Hence the communication between them must be constituted by some kind of network protocol – in our case it is TCP/IP. But this protocol represents only the lowest layers of the communication which is extended by three other protocols on the top two levels of the OSI model [10] – they are CORBA, KQML and the content language itself.

3.1 CORBA and KQML

The inter-agent message-transporting layer is constituted by CORBA (Common Object Request Broker Architecture [11]). This technology implements a transparent middleware for data transport and function calls between processes in
one machine or two separate machines. All operations all strictly platform independent – in the sense of hardware and software equipment of the machines. Moreover, there are very good standard specifications of a data-structure mappings into many programming languages (we are using mappings into Java [12] and Python [13]).

A mapping into particular language is possible because all data structures and object interfaces are defined with a neutral meta-language IDL (Interface Definition Language [14]). This language is formally similar to structures used in data modelling meta-language UML (Unified Modelling Language).

Under the CORBA layer which facilitates the connection between agents, there is a KQML (Knowledge Query and Manipulation Language [15]) layer. The KQML language is based on the linguistic theory of the speech act [16], published by Searle in [17]. This theory in short says that every communication act can be categorized as for instance: an announcement, a query, a demand etc.

KQML communication is strictly divided into two levels – level of the speech act resolution and the message content resolution. For each type of the speech act there is one or more so called “KQML performatives.” Performatives give us a basic information about what type of information or action an agent demands.

KQML has two fundamental concepts: (1) each agent is autonomous and only it decides what to do in a particular time, (2) agent A can ask agent B for information from its VKB. According to these information and according to the content of its own VKB the agent drives its behaviour. KQML contains a set of performatives which express a desire of an agent A for agent B to make an effort to achieve some state of the environment. These performatives can be used for agent activity control but at present time they are not used in our framework.

![Fig. 3. Example of EPS agents’ communication](image)

### 3.2 An Example of Agent Communication

For an illustration of the inter-agent communication we introduce a typical situation in a power system simulator (the example is simplified for better clearness): agent A represents a transformation station transforming a very high voltage to a high voltage. Agent A is connected to agent B which represents high voltage
line and which is connected to agent C representing a distribution point (see the Figure 3).

Agent B asks agent A for a notification for every A’s state change and every A’s output voltage change:

```
(subscribe
   :sender   B
   :receiver A
   :timestamp 1113340454
   :reply-with query_1
   :language KQML
   :ontology KQML_ontology
   :content (ask
                :sender B
                :receiver A
                :in-reply-to query_1
                :reply-with 2
                :language Prolog
                :ontology Power_system
                :content out_voltage(X), state(Y)
           )
)
```

As soon as the agent forms a KQML message, it contacts a CORBA Naming Service and asks it for a network address of the recipient and contacts the recipient (in the case that it has not done it already before). By this technique a dynamic connection between the agents A and B is established. Similar connection is established between agents B and C.

If there is an outage on line B and the current supply is canceled, B sends to C a message of this format:

```
(tell
   :sender B
   :receiver C
   :timestamp 1113341454
   :in-reply-to query_2
   :reply-with query_3
   :language Prolog
   :ontology Power_system
   :content out_voltage(0), state(fatal_failure)
)
```

C reacts to this message in this way: it sets all its outputs voltage to 0 V too or switches to another (backup) input line.

The big advantage of KQML is that the agent B can send a message with the same format to the visualization agent or to an agent which loads data into the failure database. An example of such visualization is displayed in the Figure 4.
4 Conclusion

We have described design and prototype implementation of a multi-agent system for active simulation of energy flow in an electrical power network. The implementation framework meets all the required criteria which were identified in the process of power systems simulation analysis. It has already proven to be enough flexible, open, dynamic and robust even for possible real-time power systems monitoring.

The system is general and adaptable enough to be applied not only in the case of a power system simulation. Namely, its architecture is suitable for usage in a largely decentralized networks of autonomous data gathering and processing units. Within the future development, we plan to obtain real data of an example power system and adjust the parameters of the energy flow characteristics in the implemented system regarding its possible connection to thin-client sensors for the possibility of real-time data acquisition.

References


5. COBALT project, [http://www.irit.fr/recherches/SIERA/GRS/coop.frame.shtml](http://www.irit.fr/recherches/SIERA/GRS/coop.frame.shtml)


