MASEVA: IMPLEMENTATION OF THE DAI/MAS IN THE PROCESS AUTOMATION

POLNICKÝ JAN, KMÍNEK MILOŠ
Institute of Chemical Technology (ICT) Prague,
Department of Computing and Control Engineering
Technická 5, 166 28 Praha 6 - Dejvice, Czech Republic
Tel.: +420 224 355 021; Fax: +420 224 355 053
E-mail: Jan.Polnicky@vscht.cz; Web: http://uprt.vscht.cz/polnicky

Abstract: This paper presents one of the various approaches of implementation of the Distributed Artificial Intelligence / Multi-Agent System (DAI/MAS) in a heterogeneous environment of the process automation. A relatively simple SCADA/HMI system known as TomPack designed for measuring, data acquisition and control of laboratory evaporator, has been softly integrated with other autonomous systems (Oracle, Matlab), which form a complex technological-information system (TIS). From the MAS point of view, extended TIS can be understood as a distributed, multi-agent system. The final TIS should be able to fulfill the original tasks as well as the new ones: on-line archiving & direct calculations, prediction of system variables, data presentation and even the diagnosis of the whole system. The aim of the first part of the project was to suggest fragmentation of the system among agents and their competences, to decide MAS framework applicability and finally to establish methods and facilities of communication, coordination and cooperation. The second step was to recommend an organization of agent's community and build a suitable relational database (RDBMS) structure for shared storage of metadata – in other words: platform and hierarchy of the agents & architecture of the black-board.

Keywords: process automation, multi-agent system, black-board structure, RDBMS, evaporator

1 INTRODUCTION – THEORY, STANDARDS & FRAMEWORKS FOR DAI/MAS

Distributed artificial intelligence deals with collaboration of several identical or various systems for shared problem solving. It is based on the idea of the co-operation, co-ordination of single-minded, autonomous biological individual's activities in closed community. DAI focus on questions of representation, processing and utilization of knowledge in these systems and it also looks for suitable decision making models. There are some advantages which collaborative systems brings along:

- a reducing of the solution time
  - parallel processing/technique possibilities
- a reducing of the communication requirements
  - specialized member carry out selected, relevant tasks only
  - the sharing of final results
- an increase of the operability and reliability
  - team may take more "specialists"
  - team members can substitute each other

The DAI can be divided to two branches: Distributed problem-solving and Multi-agent system (MAS). Distributed problem-solving system insists on decentralization of solution, where the community of modules is tightly bundled. In contrast, MAS is consisting of several autonomous "agents", which are softly integrated in favour of general problem solutions.

What “agent” means in this context? The agent may be regarded as independent computer system, sensor or automaton. Here is my definition of agent in MAS:

- **Agent is permanent active entity, which knows its specific function. Agent is able to feel and interact with surroundings to reach its purposes. It is also able to communicate with one agent together at least.**

There are some software MAS frameworks like JADE (Java Agent DEvelopment), JADE-LEAP, JAF, JESS, Zeus, etc [1]. These agent’s frameworks simplify the implementation of multi-agent systems through a middle-ware that claims to comply with the FIPA specifications [2] and though a set of tools that supports the debugging and deployment phase. For example, JADE [3] as open source software is completely implemented in Java language and it has minimal system requirements. That is why; the agent platform can be distributed across several machines/nodes with different operating system, which supports Java Virtual Machine (JVM). JVM as complete Java run-time environment (JRE) is executed on each host for one Java application. All agent communication is performed through message passing, where FIPA-ACL is the language to represent messages. JADE agent platform provides a Graphical User Interface (GUI) for the remote management, monitoring and controlling of the status of agents. The GUI allows controlling other remote FIPA-compliant agent platforms also.

In this concept, all agents of MAS are based on same pattern – Java containers and they have to be located on the platforms supporting JVM. Therefore this attractive model may not be applicable for process automation – “industrial MAS”, where non-uniform agents are not “running” on computers with OS and methods of communication are variant.

2 CASE STUDY – MASEVA: MULTI-AGENT SYSTEM FOR EVAPORATOR

Since 2001 laboratory evaporator station is expanding progressively not only at the side of technical equipment, but also its technological-information system (TIS) is growing as well. Technical description and specifications of TIS, evaporator unit and relational database structure for process data storing are closely described in my previous papers [4, 5].

How to take the advantages of MAS in process automation? Needs of the parallel task processing are characteristic in the process control. A secondary unit on the background can perform a great deal of computing processes. So we decided to implement some ideas of DAI/MAS methodology to our TIS for advanced control problem solving on the evaporator unit – such extended system was called “**MASEva**”.

Task 1 – Ancestral system decomposition and expansion: The original system was logically fragmented to the three particular units – *Evaporator*, *PLC* and *TomPack*. Gear of our TIS is strong, well known *Oracle Database Server*, which is providing a great potential for system integration and development.

Next step was incorporation of *Oracle Application Server, Oracle Management Server* and *Matlab* to work with our TIS. We used a component of Oracle iAS - Portal as complex environment for development, administration and presentation of information. The *Oracle Management Server* as toolkit for advanced; centralized administration was designed mainly for monitoring of condition and tuning of Oracle components. The *Matlab* is the language for technical computing, which integrates computation, visualization and programming in easy-to-use environment.

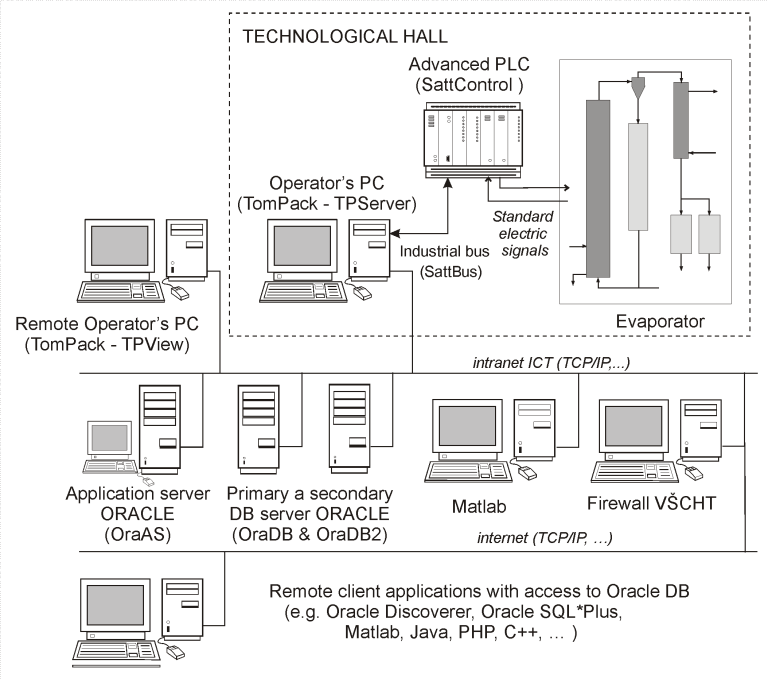

Fig. 1 Communication architecture in the TIS

Task 2 – MAS definition - Agents: For each particular, autonomous unit, which has been called as "agent", we assigned competences. Competences are used in MAS to identify the behaviours that agents must have, or be able to acquire, in order to achieve their specific targets. Now there are *seven agents*, which were distributed into *five layers* - Technological, Control, Data, Diagnostic and Presentation layer:

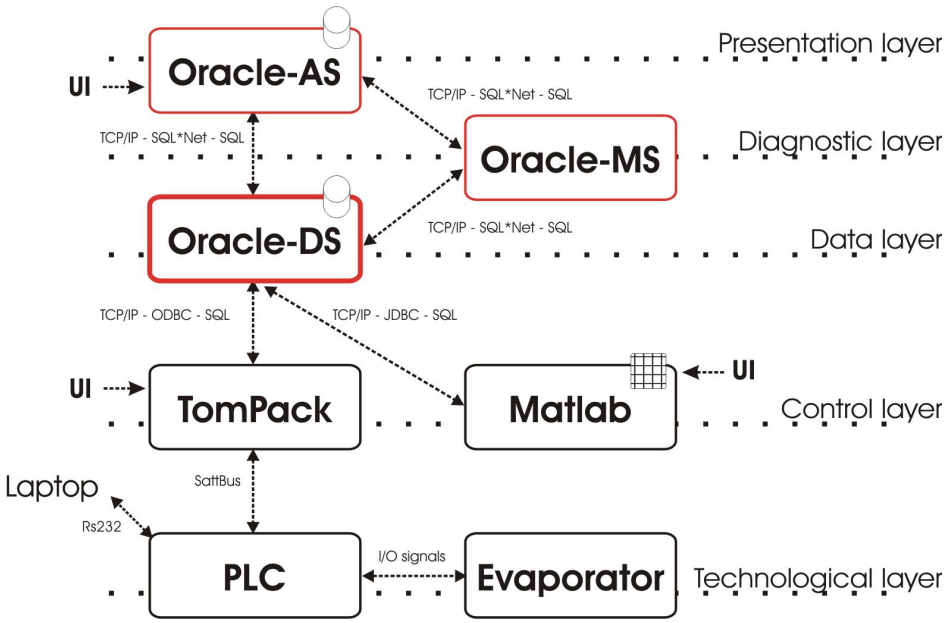

Fig. 2 Schema of MAS for Evaporator

Agent called *Evaporator* is consisting of the group of plant instrumentation devices as sensors and actuators. There are several sensors for measuring the temperature, pressure, liquid level and flow volume. The actuators, most often valves are ensuring flow of water or steam in our evaporator unit. Agent *Evaporator* belongs to group of **interface, reactive agents** in technological layer together with agent *PLC*.

Second **interface** agent *PLC* - advanced programmable logical controller (SattControll OP45SB) is interconnected with each of plant instrumentation devices and it gathers all the data periodically. The controller is able to communicate with certified software on laptop via RS232 interface to record a new sequential code to its memory. *PLC* can also influence with control circuits on evaporator with use of virtual, software PSD regulators (digital PID) – it is **proactive** agent.

Agent *TomPack* is SCADA/HMI software running on operator's computer under operating system MS Windows. It consists of modules TPServer, TPView, TPConfig and UniServer, which can be distributed across several nodes. TPServer is a static, master unit and it is **proactive** agent. UniServer is dynamic data exchange (DDE) server. The rest of modules are able to exchange the information with TPSever from any client PC using the intranet – they are **user-interactive** and **mobile** subagents.

Next agent *Matlab* can consider to be also mobile agent, which is indirectly controlling the actuators (valve position) according to operator's demands with reference to predicted system behaviors. It is consisting of number of scripts (m-files), where one main, permanent script is running as background process. This ever-ready script is a parent process for executing the child scripts on requests. *Matlab* as modeling and predictive agent belongs to group of **proactive**, **user-interactive** and **mobile** agents.

Agent *Oracle Database Server*, abbreviated to *Oracle-DB* is a global **super-agent**. This agent is based on relational database management system (RDBMS) - Oracle9i, which maintains the structures of tables including all data in two logical groups of database objects. First group is dedicated [4, 5] to process data storing (dynamic and static, measured values) - process data archive of the evaporator. There are also saved meta-data as detailed information about sensors (including its location, measurement methods and variables) and physical-chemical tables with one descriptive table. Second group of the database objects is exclusively intended for DAI/MAS and its operations. More information about this data model part is in the next chapter.

Agent *Oracle Application Server*, abbreviated to *Oracle-AS* was introduced as server for data presentation of the evaporator. It is composed of small data repository and Portal – middle tier. Portal is presenting content of remote database server via dynamic HTML pages on integrated web-server accessible from campus intranet. The layout of graphical user interface (GUI) was designed in this software tool. Operators are using thin client – Internet browser (preferably MS-IExplorer) to access the Evaporator portal. *Oracle-AS* can be classified as **user-interactive** agent and its clients as **mobile** in addition.

Last agent *Oracle Management Server*, abbreviated to *Oracle-MS* was implemented to MASEva for whole system diagnosis. It will provide a testing of status of all agents of TIS running on operating system like Linux or MS Windows including condition of whole environment (the CPU, memory and disk usage). The *Oracle-MS* data repository containing test results with history and another auxiliary information about monitored nodes is also located in Oracle database. Client of *Oracle-MS* is called Oracle Enterprise Manager and it is a Java based application console. *Oracle-MS* behaves as **proactive** agent, because it is pursuing the MAS system without any superior human command once preceding configuration.

Tab. 1 Agents and their competences in the MASEva system

Agent	Layer	Description	Type	Competence
EVAPORATOR	Technological	Plant instrumentation devices: Sensors Actuators	i,R	Measuring – primary data source Regulating – actuating signal
PLC	Technological	Advanced Programmable Logical Controller – SattControll OP45SB - local super-agent)	i,P	Process data acquisition Regulation – virtual PSD (digital PID)
TOMPACK	Control	SCADA/HMI software – TomPack central module TPServer (UniServer) (Clients: TPView, TPConfig)	u,P (M)	Process control & visualization User interface for process control (GUI) Computer control (algorithmic)
MATLAB	Control	Matlab – computing environment Neural Network toolbox	u,P (M)	Predictor model (Neural Networks) On-line regulation with use of prediction
ORACLE-DB	Data	Oracle Database Server EE - global super-agent (Client: Oracle Enterprise Manager)	P,*	On-line data storing (process, DAI data) On-line calculations Historical process data archiving
ORACLE-AS	Presentation	Oracle Application Server EE Portal components (Client: Internet browser)	u (M)	Data presentation User interface for visualization (GUI) Process data export (Excel, HTML…)
ORACLE-MS	Diagnostic	Oracle Management Server (Client: Oracle Enterprise Manager)	P (M)	Centralized diagnostic System monitoring and testing

Legend for column "Type": i~interface; u~user-interactive; R~reactive, P~proactive; M~mobile; *~global agent

Task 3 – Methods, facilities and formulism of communication: Methods and facilities for communication between these heterogeneous agents inside of MAS are very various. That is why the application of MAS framework in this system would be very unadvisable.

Low-level communication between plant instrumentation devices of the *Evaporator* and *PLC* is performed by I/O signals. The PLC (SattControll OP45SB) is connected to operator's computer though special industrial I/O bus – SattBus. On the TomPack side, there is communication adapter SattBus ISA card mounted in the operator's computer. DDE server – UniServer as a part of *TomPack* agent using SattBus protocol to read data stored in *PLC* memory buffer. Main module TPServer exchanges information with DDE server and it uses TCP/IP protocol and ODBC interface to communicate with *Oracle-DB*. TomPack scripts are querying/storing essential information from/to database via built-in ODBC functions in combination with SQL language. Matlab scripts go on the same way; only the difference is the JDBC driver (using the JBDC/ODBC bridge), which is used by multiplatform Matlab to communicate with relational databases.

We decided to use a DAI/MAS **blackboard table**, which simplifies communication between particular units. Agents are strictly routing its messages to one address – to shared data structure, which is available to all agents. At bottom line, agents communicate indirectly through the use of messages. Each message contains information about the sender, recipient, about given time-stamped task, its status and inputs and outputs. Blackboard structure and relations between tables are described in next figure.

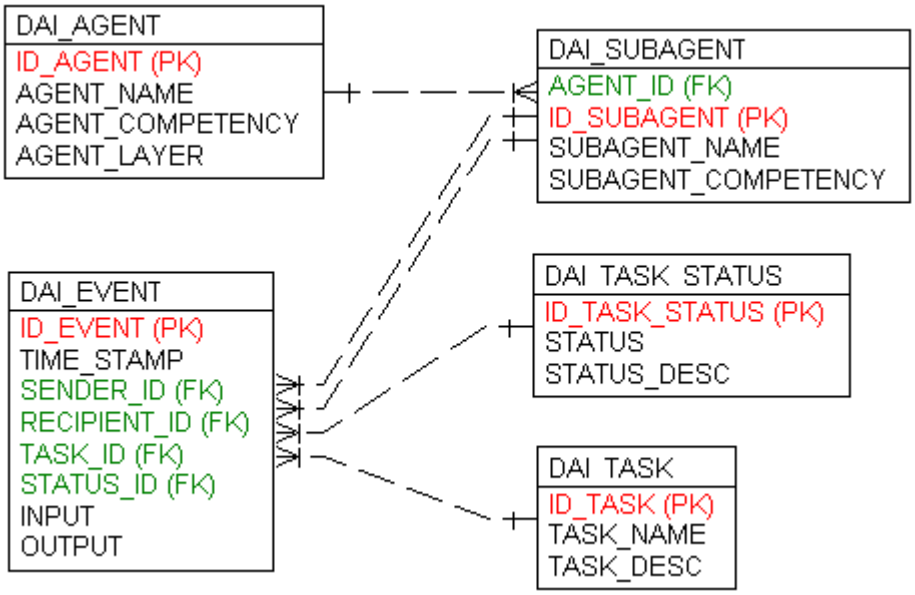

Fig. 3 The blackboard table structure - ER data model for MASEva

We designed software **dispatchers** at *TomPack*, *Matlab* and *Oracle-DB* agent's side for the MASEva system **coordination.** Dispatcher is an algorithm coded in the native language (i.e. TomPack dispatcher is a TPServer server code in "simplified C" language; Matlab dispatcher is a m-file script written in Matlab technical language or Oracle-DB dispatcher is a database trigger/procedure coded in PL/SQL). Basically, it is a ever-ready, permanent process, which is periodically checking the blackboard table for the new tasks. If new task for current agent is found, dispatcher as a parent process is executing appropriate child subroutine for the task processing. Hence the agents are unequal and they hold different amount of the information; agent community is living in the **hierarchical organization**:

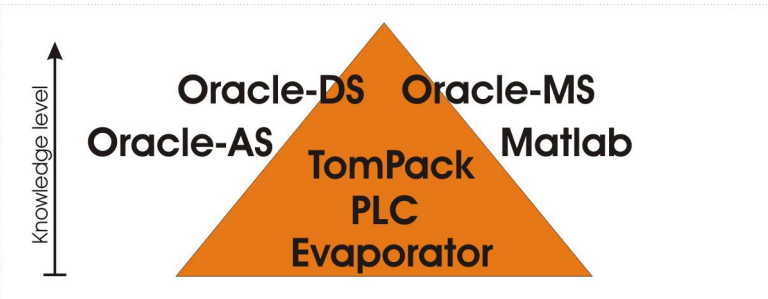

Fig. 4 The hierarchy of agents in MASEva system

3 CONCLUSION – DISCUSSION

We have introduced the pilot project called MASEva – a system for the supervision & advanced control of the evaporator unit. The project is based on cooperative autonomous systems – the agents with shared database repository, where two datamodels are located. One of the datamodels is oriented to the technological process (sensors on the unit, methods of measurement, measured values and variables). The second datamodel – the structure of the blackboard is exclusively dedicated to DAI/MAS system and its operations. Generalized tables of the blackboard maintain the information about agents, tasks and events. The communication ways between the agents were tested and proved successful; now we are focusing on the designed structure of the blackboard and enhancement of the dispatchers. The described concept of the system may be applied to any other technological problems.

In the next phase, we would like to set-up a list of event-tests inside the *Oracle Management Server*, which will provide a centralized diagnostics of MASEva components. Our goal is to integrate an user-friendly supervision DAI/MAS interface for the to the *Oracle Application Server*, which will show how the agents cooperate (all MAS events, agent tasks and statuses) and will also provide a web-based monitoring and diagnostics tracing tool.

4 ACKNOWLEDGEMENTS

The work has been supported by the fund No MSM 223400007 of Ministry of Education of the Czech Republic. This support is very gratefully acknowledged.

References

[1] List of the agent software platforms [online].
URL: http://www.agentlink.org/resources/agent-software.php, [2004].

[2] Homepage of the FIPA specifications [online].
URL: http://www.fipa.org, [2004].

[3] Homepage of the JADE project [online].
URL: http://sharon.cselt.it/projects/jade/home.htm, [2003].

[4] POLNICKÝ, J. *On-line databáze procesních dat pro laboratorní odparku (Diploma thesis)*. ICT Prague, 2002.

[5] POLNICKÝ, J. Brief developer's conception of information system for soft real-time processes. In.: *Conference proceedings – Process Control*, Štrbské pleso, Slovakia, 2003.