ExPlanTech: Applying Multi-agent Systems in Production Planning

Michal Pechoucek[1], Ales Riha[1], Jiri Vokrinek[1], Vladimir Marik[1], Vojtech Prazma[2]

[1] Gerstner Laboratory, Department of Cybernetics, Czech Technical University, Technicka 2, Prague 6, 166 27,Czech Republic

{pechouc, marik}@labe.felk.cvut.cz; {riha, vokrinek}@protys.cz

[2] Modelarna Liaz, Kamenicka 453, Liberec 6, 460 06, Czech Republic, jakost@modelarna-liaz.cz

Abstract

This paper presents work carried out within the 'ExPlanTech' project (IST-1999-20171) funded in part by the European Commission's Information Technologies Programme. The mission of the ExPlanTech technology transfer project is to introduce, customize and exploit the multi-agent production planning technology (ProPlanT multi-agent system research prototype) in two specific industrial enterprises. An agent driven service negotiations and decision process based on usage-centred knowledge about task requirements substitutes the traditional production planning activity. We introduce methodology for integration of the project-driven production planning based on agent-based engineering within the existing enterprise resource planning system. This novel production planning technology will facilitate optimisation of resource utilization and supplier chain while meeting the customer demands. This paper describes a FIPA-compliant implementation of the ExPlanTech technology at the LIAZ Pattern Shop manufacturing company. We describe the structure of the agent community, types of agents, implementation of the planning strategy and its incorporation within the real production environment.

Keywords: agent, multi-agent system, FIPA, production planning

1 Introduction

The concept of multi-agent system as a software engineering implementation paradigm emerged just recently. While there is a number of successful applications in various areas, in our contribution we will present and comment results of applying the multi-agent paradigm in the area of production planning.

Multi-agent system is a collection of heterogeneous, encapsulated applications (agents) that participate in the decision making process. Agents communicate, collaborate and negotiate in order to meet its own design objective but also a goal that is shared within the community.

In the Gerstener Laboratory (CTU) there was developed a multi-agent system prototype ProPlanT (Marik et al. 1998) that assists in elaboration of production plan in the domain of project-driven manufacturing. The architecture of the system reflects the organizational structure of the respective enterprise. There are various classes of agents (e.g. planning agents, configurators, decomposition agents, database agents but also agents integrating various pieces of hardware).

A multi-agent system (Wooldridge and Jennings 1995, Zhong et al. 1997, Singh et al. 1998) is not a static, hard-wired collection of applications that exchange data. Organization of a multi-agent system is dynamic, agents form teams and coalitions with respect to the specific goal and their actual availability. This contribution will focus on agent interaction intelligence. We will discuss how much is expected an agent to know about its collaborators and how such a piece of information may be utilized for fast and effective production planning. As much as the plan (the result of the system's planning activity) also the problem solving knowledge is distributed within multi-agent community.

However, there are certain patterns of inter-agent interaction that are hard to analyse from the point of view of a single agent. Therefore, we introduce a complementary concept of the intelligence within the multi-agent system. Let us have an agent, that does not participate directly in the planning process, whereas it only observer how agents interact and how they carry out distributed decision-making. This agent, we call him a meta-agent, has the capacity to detect production bottlenecks, communication loops and other possible inefficiencies.

In this paper we will explain the theoretical concepts and at the same time it will give an account on our experience with applying this technology in the real industrial enterprises.

2 ProPlanT

2.1 ProPlanT Multi-Agent Platform

ProPlanT agents have been implemented mostly in C++ and they run on WinNT 4.0 operation system. Each agent is an independently running application. The agents exchange messages via TCP/IP sockets. The agents use a specific PMTP (ProPlanT message transport protocol), KQML as an ACL (agent communication language) and KIF as the ACL content language (Finin et al. 1995).

The ProPlanT multi-agent platform provides the developers with several mandatory/optional agents – we call them **ProPlanT Principal Agents**. The **Facilitator** agent maintains a white-page-list of the existing community members. Upon a registering of a newcomer, the Facilitator broadcasts its IP address and its port within the existing community and provides the newcomer with the white-page-list. The **Agent Factory** constructs a community of agents by running a specific community-description-script (equally it can stop/kill the agents). The **Meta-agent** sniffs the communication messages among agents and re-constructs the agent's states. The meta-agent has been primarily used for visualization of the community organizational structure and communication flow among the agents.

2.2 ProPlanT Production Planning Architecture

In the ProPlanT the classical planning and scheduling mechanisms have been substituted by the processes of negotiation, job delegation and task decomposition within a community of autonomous agents, each of which represents production or information unit(s) of the modelled factory. There is no central agent or any central control mechanism. ProPlanT system relies upon two fundamental super-classes of agents: **intra-enterprise** (IAE) agents and **inter-enterprise** (IEE) agents. In the category of the IAE agents we distinguish among the following basic classes as seen in figure 1:

– **Production Planning Agent** (PPA) is in charge of project planning. Its aim is to construct an exhaustive, partially ordered set of tasks that need to be carried out in order to accomplish the given project. It contracts PMA agents.

– **Production Management Agent** (PMA) performs project management in terms of contracting the best possible PA agent (in terms of operational costs, the delivery time and current capacity availability). PMA delegates its responsibility either to another PMA or it controls work of a group of PA agents contracted for the considered task.

– **Production Agent** (PA) represents the lowest level production unit that simulates or encapsulates shop floor production process on the IAE. PA carries out the parallel-machinery scheduling of given tasks and manages resources allocation via a special type of database agents.

– **Customer Agent** (CA) is another instance of an IEE agent that provides customers interface into the system.

– **Meta Agent** (MA) is a special monitoring agent which visualizes information, material and workflows across the agents' community and advises on optimal system's efficiency. Such architecture provides the user with decision support regarding distributed planning of manufacturing processes.

Shall be noted that only the IAE agents have been implemented in the ProPlanT multi-agent system prototype. Special kinds of IEE agents were subject of ExPlanTech project implementation.

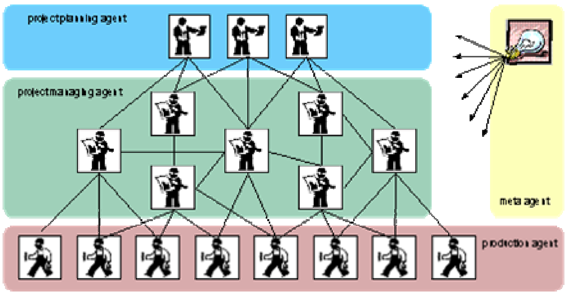

Figure 1. ProPlanT General Architecture

2.3 Tri-base Acquaintance (3bA) Model Technology

As production planning is usually a problem solving activity with rather high cardinality of its complexity, we have suggested a coordination model that contributes to saving computational resources (Pechoucek et al. 2001). As the architecture proposes, the important part of the problem solving complexity is transferred into negotiations and job delegations among the PMA agents. We have investigated possible ways of saving communication traffic in the PMA coordination process, by which the overall complexity would be reduced. Each PMA agent has been equipped with a specific knowledge structure (tri-base acquaintance model) that administers agents computational model of its social awareness. This social knowledge (Marik et al. 2001, Pechoucek et al. 2000, Cao et al. 1997) (stored in three separate bases) keeps precompiled important pieces of information that would have to be acquired through communication otherwise. The ProPlanT technology provides implemented mechanisms for social knowledge representation and maintenance throughout the lifecycle of the community.

3 Problem Description

The application domain of this technology transfer exercise has been project-oriented type of production. Unlike mass-production, in the case of project-oriented production there is always a limited series of possible products of one type manufactured (e.g. space shuttles, power turbines, TV broadcasters or patterns and forms). Accordingly, an important amount of resources has to be devoted to design-related activities such as quotation and configuration, design and production planning. Production-oriented planning consists of three separate (while interrelated) phases as seen in figure 2:

– **quotation and configuration**, where the quotation engineer agrees with a customer on detailed specification of the product to be manufactured (this phase is very often supported by different areas of artificial intelligence such as, constraint programming (Wielinga and Schreiber 1997), theorem proving applications (Lowe et al. 1998), or possibly evolutionary computing (Kubalik and Lazansky 1998),

– **project specification**, where the complete product configuration is transformed into the project specification and the appropriate resource specification (this phase is rather case specific and it is driven by the domain knowledge), and

– **resource allocation**, where the required resources are allocated in time to appropriate resource providers, such as machines, technicians, departments, etc.

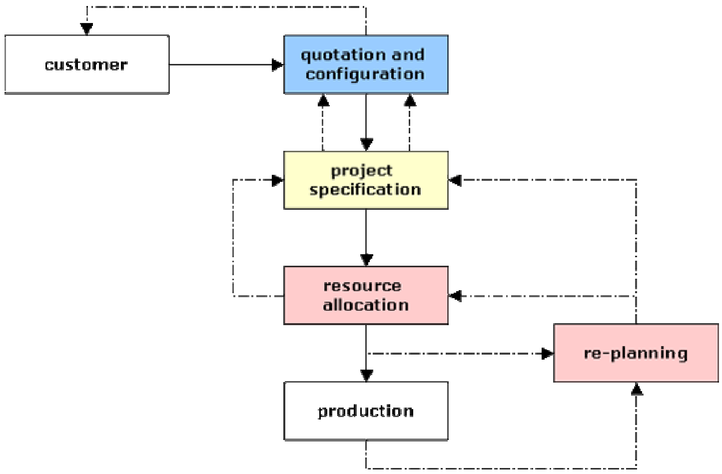

Figure 2. Phases of the production-oriented planning

Our multi-agent solution concerns primarily the latter phase, during which it becomes rather difficult to allocate a project-related resources to partially booked resource providers, to allow re-planning due to resource providers malfunction or scheduling higher priority projects.

As already noted this paper will show how the ProPlanT multi-agent planning technology has been applied in industrial sites of our partner factory. LIAZ Pattern Shop Ltd belongs to the leading European producers of patterns and forms mainly for automobile industry. LIAZ has to process approximately 120 production orders per month with production lead-time from 3 to 6 weeks. The structure of orders is extraordinary wide and continuously changing. Therefore the ProPlanT technology integration with the information system at the LIAZ enterprise will focus on decision-making support for the company management in order to advise on the optimal production planning process with respect to customer requirements while optimally allocating available resources and while making the maximal profit.

Note: Hereafter we will talk about a particular, case specific instance of the ProPlanT multi-agent system that has been implemented in LIAZ. As this implementation is substantially different from the ProPlanT prototype we will refer to it as an ExPlanTech multi-agent system.

3.1 Role of ExPlanTech

The optimal production plan should balance the available LIAZ resources while maximum number of orders is produced. Considering the limited LIAZ workshops/shop floors capacity the decision has to be made whether the specific task/subtask will be provided internally or subcontracted externally. Such decision might be crucial in order not to threaten successful completion of other orders and misbalance the whole production flow. The efficient supply chain/service provision management can be handled using the agent-based approach as described in the ProPlanT technology within the *inter-enterprise and extra-enterprise level.*

The ExPlanTech technology integration is aimed at the improved strategic decision-making, e.g. to efficiently use available resources (human, material supply, workshop capacity) while the maximum number of clients' orders have been satisfied and carried out, as seen in figure 3. Moreover, orders are categorized by an internal priority value ensuring that orders with higher priority is manufactured before the lower ones. The ExPlanTech system shall observe each workshop load and consequently recommend the suitability of the requested order.

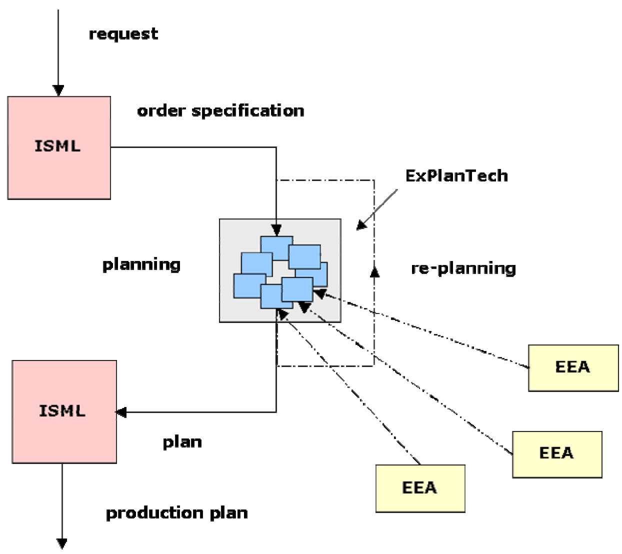

Figure 3. Role of ExPlanTech in LIAZ

3.2 MAS Structure

Whole IT solution is created by ExPlanTech system together with information system ISML, see figure 4. ISML stores necessary data about production and provides complete business solution. It is going to support our system with proper data for production planning as well. In further text only ExPlanTech multi-agent system is going to be described. From ISML point of view exchanging production data and interface between ISML and ExPlanTech are mentioned. The ExPlanTech prototype for LIAZ Pattern Shop consists of several different types of agents – Configurator Agent, Scheduler Agent (Workshop), Database Agent and Extra-Enterprise Agents.

3.2.1 Database Agent

Database agent (DBA) has several roles within the agent community. The main one is to support the other agents with production data (orders, preorders, calendar, workshops) DBA is used as a communicational bridge between ExPlanTech system and ISML. The data flow both ways between the systems. There are sent information describing production (orders, pre-orders) from ISML side to ExPlanTech. There are sent ready plans for each workshop other way. Data are sent in XML format via TCP/IP connection. Received data from the outside of the community (ISML) are pre-processed and then delivered to other agents (Configurator, Workshops). Pre-processed means that received XML document is parsed, data are extracted, some data are stored in local database (Interbase) and new FIPA messages contents are created. This agent doesn't have its own GUI interface. Local database mirrors agents' data (knowledge) and other auxiliary data necessary for proper functionality of the system.

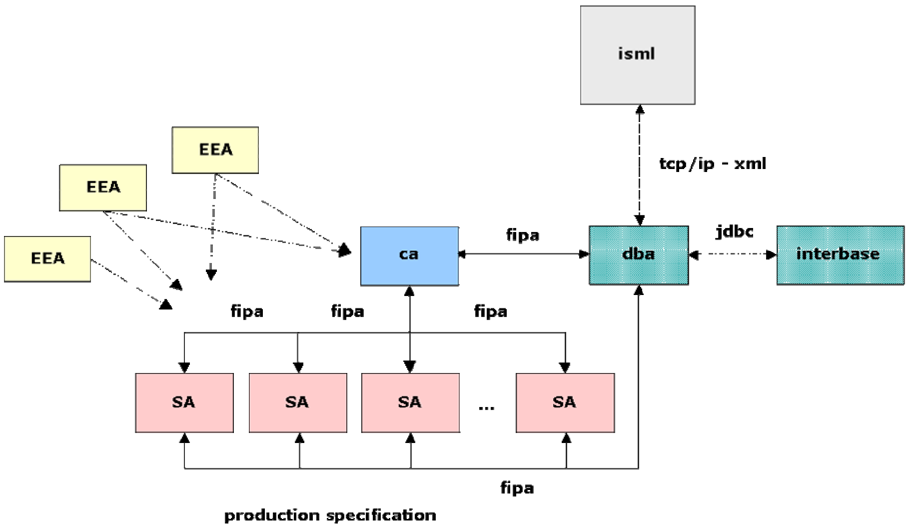

Figure 4. IS solution in LIAZ Pattern Shop *(The figure depicts scheme of the entire solution in LIAZ. ISML stands for ERP system in LIAZ. The other blocks represent the multi-agent system ExPlanTech, CA is agent configurator, SA is agent scheduler, dba is database agent, the block called interbase represents the local database and EEA stands for the instance of the extra enterprise agent – agent monitor. There are also communication and data flows figured, where fipa means the inter-agent flow of fipa messages, tcp/ip is communication channel between the ISML and the system ExPlantech.)*

3.2.2 Configurator Agent

This agent stands over the group of Scheduler agents and provides them with production data about orders. The agent itself shows two plans: a) order before planning in relative time, and b) order after planning in absolute time. CA agent has implemented simplified 3bA model described above. It has functionality kind of both agents PPA and PMA. The 3bA model, because of three knowledge bases and their structure, facilitates agent community behaviour and increase communication traffic. It has information about workshops (their load, free spaces in plan) and tasks (deadline of orders etc.).

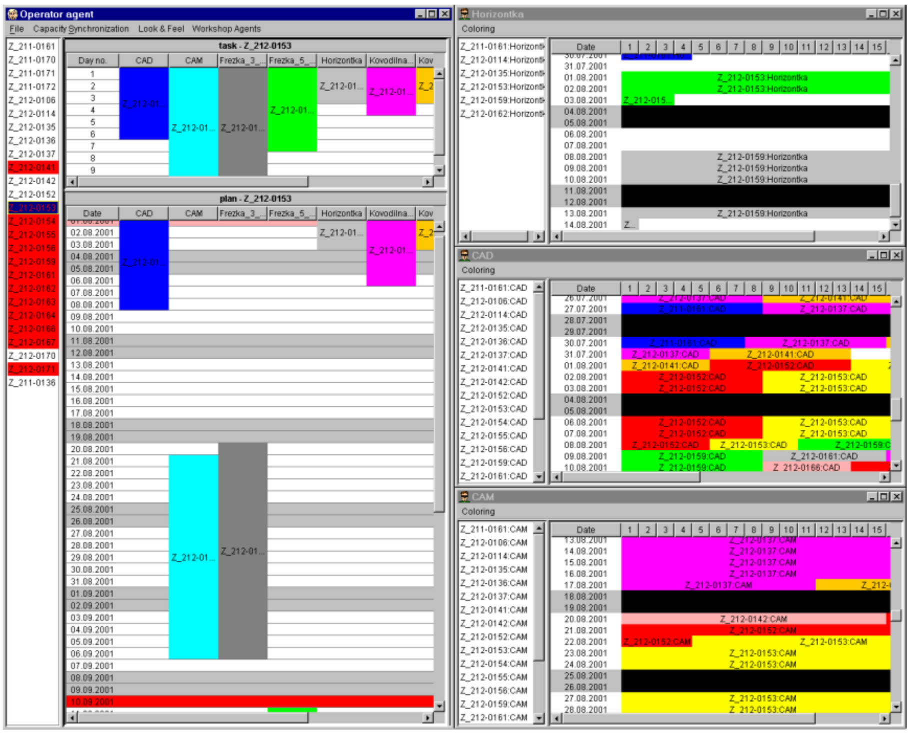

Figure 5. Screenshot of ExPlanTech multi-agent system *(Leftmost window represents Operator agent with list of orders, order definition window in the upper part and planned order in the lower part of the window, there are three workshop agents on the right side of this figure.)*

3.2.3 Scheduler Agent (Workshop)

There are as many Scheduler Agents as workshops within the factory and stand for PA agents. The ExPlanTech system depicts the entire production process in the LIAZ factory. The main role of this group of agents is to create plans or better said schedules for workshops. Scheduler Agent tries to put new order (pre-order) coming from ISML through DBA and Configurator Agent into schedule so that all constraints are not violated. It takes into account deadlines of each order, priorities, precedence dependencies, daily capacity of each workshop etc. SA agent shows how could the future production look like, see figure 5, what order might not be finished in time because of accepting new order. These agents respectively whole ExPlaTech help to a user to exploit production capacity in optimal way. Optimal way is on the one side to minimize void times and on the other side to accept correct number of orders to all of them be completed in time. There are going to be used Eager and Lazy planning strategies. SA agent sends computed plan to DBA agent and some data about planned tasks are sent to Configurator Agent.

3.3 Extra-enterprise agents

The ProPlanT technology has been extended with the class of extra-enterprise agents (EEA). There are two classes of EEAs suggested: the **monitor agent** and the **resource agent**.

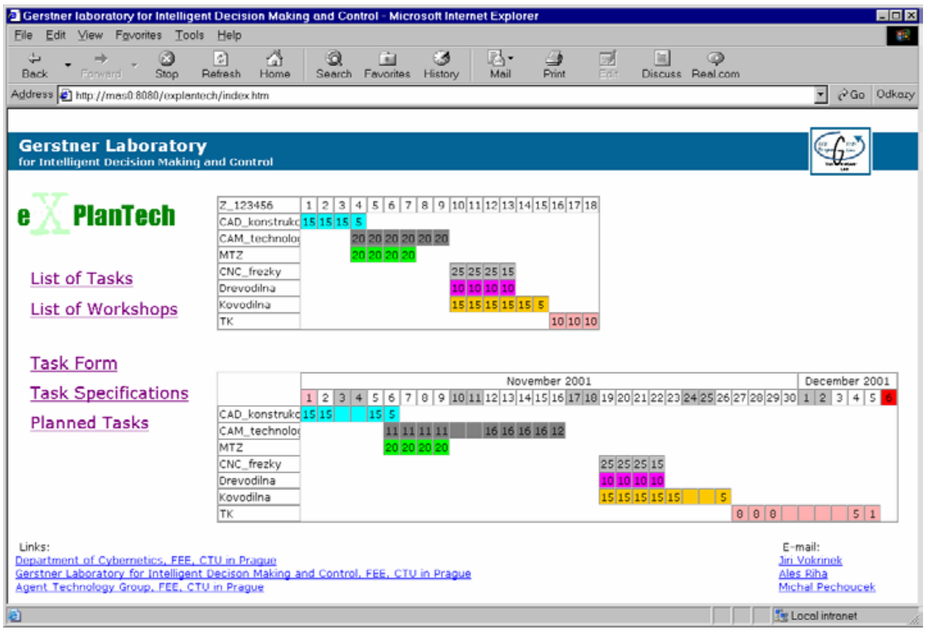

Figure 6. Monitor Agent *(Upper table shows order specification in relative time and the table below represents the particular schedule for this order.)*

3.3.1 Monitor Agent

The Monitor Agent is going to serve customers of a factory, who use the ExPlanTech system for production planning in an Internet browser as seen in figure 6. According to his/her access rights the customer is able to trace his/her orders and watch their statuses (whether producer can make orders in time). Important is that data are not stored in the database but it is gathered on-line directly from the agent community. Technically, there is a web server on the side of the manufacturer exploiting ExPlanTech and a web agent that communicates between the agent community and the web server. There is going to be some sign-in authentication and security provided in the near future. Secondly, the monitor agent is planned to be used by the management members of the factory, who can inspect operations of particular workshops of the factory while they are off-site. They access not only the information about particular orders, but unlike the customers they can be provided with the information about loads of the workshops and their plans for the future.

3.3.2 Resource Agent

This agent should work on the side of an enterprise's suppliers or co-operators (in the case of outsourcing). A partner enterprise (organization) could announce status of services and resources, which are currently available. This shared (public) knowledge should be prepared in some standard form with respect to common agreed ontology. The Resource agent could, for example, read data from the partner enterprise database or could be provided with this data by the ERP system. The ExPlanTech system can then have more precise and actual data for computation (delivery dates, prices, amounts). ExPlanTech could contract such an agent in case there are not enough resources and postponing of the deadline is not possible.

4 Implementation

We were looking for such an environment that would maintain basic multi-agent necessities, which are communication standards, agent administration etc. FIPA (Foundation for Intelligent Physical Agents) (FIPA 1998) provides a set of such standards and reference implementation that have emerged from industrial needs and achievements of the research community. Each FIPA agent must be registered on a platform in order to interact with other agents on that platform or inside other platforms. Each FIPA agent must be registered on a platform in order to interact with other agents on that platform or inside other platforms. An agent platform (AP) consists of several mandatory capability sets namely Agent Communication Channel (ACC), Agent Management System (AMS) and Directory Facilitator (DF). AMS is an agent, which manages lifecycle of other agents, such as creation, deletion, suspension, resumption, authentication and migration. It provides a ”white pages” directory service for all agents resident on an agent platform. It stores address book, which maps globally unique agent names and transport addresses used by the platform. ACC is an agent, which acts as a message router between agents within the platform and to agents resident on other platforms. DF provides a “yellow pages” directory service for the agents. It stores description of the agents and the services they offer.

The Extra Enterprise Agent is also an extended JADE agent as the intra enterprise agents in the ExPlanTech. We use the technology of servlets for assuring remote access to the ExPlanTech community. The servlet class is located on the standard web server. When the request comes from the static HTML page the servlet starts running.

First, monitor agent who communicate with the rest of the community by servlet, is created. Servlets' role is to dynamically create HTML page according to the data received from the community via the monitor agent.

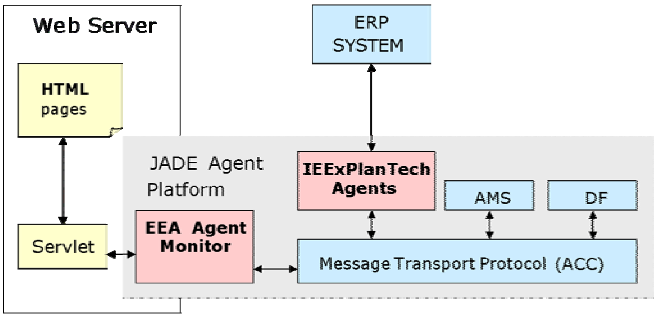

Figure 7. Scheme of the ExPlanTech implementation

4.1 Jade

There exist several software implementation of FIPA, for instance JADE (CSELT-Centro Studi e Laboratori Telecomunicazioni S.p.A) (Bellifemine et al. 2000), FIPA-OS (Nortel Networks). The ExPlanTech system is built using JADE software framework, which is fully implemented in JAVA language. The agent platform can be distributed across machines (which not even need to share the same OS) and the configuration can be controlled via a remote GUI. The configuration can be even changed at run-time by moving agents from one machine to another one, as and when required. The platform independency is important advantage of JAVA solutions. JADE has several interesting features that at least make process of implementation easier. The one of these features is the agent Sniffer that enables user to observe message flow among agents.

4.2 XML

Extensible Markup Language (XML) is an evolutionary meta-data language, approved as a standard by the World Wide Web Consortium in February 1998. XML evolved from the Standard Generalized Markup Language (SGML) as a compromise between the complex SGML and the simple, but non-extensible HTML.

XML is actually a language for creating markup languages that describe data and rules about the data. It requires applications to be defined to it before it can become truly useful. The process of defining applications is done through the use of the Document Type Definition, which defines the tags and rules within XML for a well-formed XML document. In the context of the project, and taking into account the available resources as well as the overall project aims, we will head towards defining representative tags and rules for business software component interoperability in the supported applications.

Since XML is data base-neutral, operating system-neutral, and device-neutral, it is an effective tool for defining heterogeneous interoperability.

We use XML documents for exchanging data between ExPlanTech system and ISML in LIAZ factory. Since, there exit XML parsers for JAVA we have content of ACL (Agent Communication Language) messages is in XML format as well for our convenience.

5 Conclusion

Creating a software solution that provides generic and efficient production planning automation is a very complex problem. In this contribution we are giving a progress report on how a university implemented prototype of a multi-agent system (ProPlanT) may be exploited in a real-life industrial environment – in the LIAZ Pattern Shop enterprise.

ExPlanTech, a ProPlanT implementation, has been developed implemented in JAVA and is fully FIPA compliant. While the ProPlanT prototype has been implemented in C++ and inter-agent communication has been implemented by an original (KQML-like) communication protocol, the ExPlanTech implementation reuses JADE, an open-source implementation of a FIPA-compliant multi-agent platform that is reliable and complies with latest standards in a the multi-agent community.

ExPlanTech is a powerful tool that supports long-term production planning process in the factory. It produces plans and gives views on the production, how it could look like in future if some order is going to be accepted or refused. It provides information about workshops' load, creates plan with respect to deadlines, priorities and precedence dependences of orders.

An important bottleneck was incorporating our system into the IT solution represented by information system ISML. ISML is source of data for our planning tool. We solved this task with DBA agent that serves as interface between ISML and

ExplanTech. This interface DBA agent transforms messages received via TCP/IP into FIPA messages and vice versa.

References

Bellifemine, F., Poggi, A., Rimassa, G., 2000, Developing multi-agent systems with JADE. In Seventh International Workshop on Agent Theories, Architectures, and Languages (ATAL-2000), Boston, MA

Cao, W., Bian, C.-G., Hartvigsen, G., 1997, Achieving Efficient Cooperation in a Multi-Agent System: The Twin-Base Modelling. In P. Kandzia and M. Klusch (eds), Co-operative Information Agents - LNAI 1202, (Heidelberg, Springer Verlag), pp. 210–221

Finin, T., Labrou, Y., Mayfield, J., 1995, KQML as an agent communication language. In Jeff Bradshaw (ed), Software Agents, (Cambridge: MIT Press)

FIPA: Agent Management. http://www.fipa.org.

Kubalik, J., Lazansky, J., 1998, Genetic Algorithms and their Tuning. In D. Dubois (ed), Proceedings of CASYS '98, pp.217-229

Lowe, H., Pechoucek, M., Bundy, A., 1998, Proof Planning for Maintainable Configuration Systems. In Artificial Intelligence for Engineering Design and Manufacturing, (Massachusetts: Cambridge University Press), pp. 345-356

Marik, V., Pechoucek, M., Stepankova, O., 2001, Social Knowledge in Multi-Agent Systems. In M. Luck et. al (eds), Multi-Agent Systems and Applications - LNAI 2086 Tutorial, (Heidelber: Springer Verlag), pp. 211-245

Marik, V., Pechoucek, M., Hazdra, T., Stepankova, O., 1998, ProPlanT – multi-agent system for production planning. Proceedings of XVth European Meeting on Cybernetics and Systems Research, Vienna, pp. 725-730

Pechoucek, M., Marik, V., Stepankova O., 2001, Towards Reducing Communication Traffic In Multi-Agent Systems. In Special Issue on Virtual Organizations and E-Commerce Applications, Journal of Applied System Studies, (Cambridge: Cambridge International Science Publishing)

Pechoucek, M., Marik, V., Stepankova, O., 2000, Role of Acquaintance Models in Agent-Based Production Planning Systems. In M. Klusch and L. Kerschberg (eds), Cooperative Information Agents IV – LNAI 1860, (Heidelberg: Springer Verlag), pp. 179-190

Singh, M. P., Rao, A., Wooldridge, M. J., 1998, Intelligent Agents IV (LNAI 1365) (Heidelberg: Springer Verlag).

Wielinga, B. J., and Schreiber, A., 1997, Configuration-design problem solving. In IEEE Expert, Special issue on AI and design 12(2),pp.49-56

Wooldridge, M., and Jennings, N., 1995, Intelligent Agents: Theory and Practice. The Knowledge Engineering Review, 10, No.2, pp. 115-152

Zhong, N., Kakemoto, Y., Ohsuga, S., 1997, An Organised Society of Autonomous Discovery Agents. In P. Kandzia and M. Klusch (eds), Cooperative Information Agents - LNAI 1202, (Heidelberg: Springer-Verlag), pp.183-194