ExPlanTech: Exploitation of Agent-based Technology in Production Planning

Aleš Říha, Michal Pěchouček, Jiří Vokřínek, Vladimír Mařík

Gerstner Laboratory, Department of Cybernetics
Czech Technical University in Prague
Technická 2, Prague 6, 166 27 Czech Republic
{riha, pechouc, marik}@labe.felk.cvut.cz
{xvokrine}@fel.cvut.cz

Abstract. The mission of the ExPlanTech technology transfer project is to introduce, customize and exploit the multi-agent production planning technology in two distinct industrial cases. The traditional production planning activity is substituted by agent driven service negotiations, intelligent decomposition and distributed decision-making. 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.

1 Introduction

This paper presents research and technology transfer activities carried out within the "ExPlanTech" Trial project (IST-1999-20171) funded by the European Commission. In the context of the ExPlanTech project, we work on transferring the already designed and implemented ProPlanT technology [10] into industrial enterprises. The ProPlanT technology has been provided in the form of a prototype of an integrated multi-agent system (MAS) for planning of the project-oriented production. As already noted, the application domain of this technology transfer exercise has been the project-oriented type of production. Unlike the mass-production, in the case of the 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 unique 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. The production-oriented planning consists of three separate (while interrelated) phases:

– **quotation and configuration**, during which 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 [15], theorem proving applications [8], or possibly evolutionary computing [7]),

- **project specification**, during which 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.

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 due to scheduling of higher priority projects.

1.1 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 modeled 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:

- **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 the 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 the 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 work flows across the agents' community and advises on optimal system 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 the ExPlanTech project implementation (see Section 2.3).

1.2 Tri-base Acquaintance (3bA) Model Technology

As production planning is usually a problem solving activity with rather high cardinality of its complexity, a coordination model that contributes to save computational resources [13] has been suggested. 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 the 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 agent's computational model of its social awareness. This social knowledge [3,9,12] (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.

1.3 ProPlanT Multi-Agent Platform

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 number 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 agents' states. The meta-agent has been primarily used for visualization of the community organizational structure and the communication flow among the agents.

The ProPlanT agents have been implemented mostly in C++ and they run under 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 the ACL (agent communication language) [5] and KIF as the ACL content language.

2 Implementing ProPlanT in LIAZ

As already noted, this paper describes how the ProPlanT multi-agent planning technology has been applied in the plant of our industrial partner. LIAZ Pattern Shop Ltd. (Czech Republic) belongs to the leading European producers of patterns and forms mainly for automotive industry. LIAZ has to process approximately 120 production orders per month with production lead time from 3 to 6 weeks. The spectrum of orders is extraordinary wide and continuously changing. Therefore the ProPlanT technology integration with the information system at the LIAZ enterprise focuses on decision-making support for the company management. The goal is 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.

2.1 Role of ExPlanTech in LIAZ

The optimal production plan should balance the available LIAZ resources while maximum number of orders is processed. 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 not to 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 ProPlanT technology integration (see Figure 1) is aimed at the improved strategic decision-making, e.g. how to efficiently use available resources (human, material supply, workshop capacity) while the maximum number of clients' orders have been satisfied and carried out. Moreover, orders are categorized by an internal priority value ensuring that orders with a higher priority are manufactured before those with lower ones. The ExPlanTech system shall observe each workshop load and consequently provide the suitability measure of the requested order.

The core of information solutions in LIAZ is their custom tailored ERP system called **ISML** (Information System Modelarna Liaz). ISML stores necessary data about the production and provides a complete business solution. The idea was not to have the ExPlanTech as a key component that others will strongly depend on.

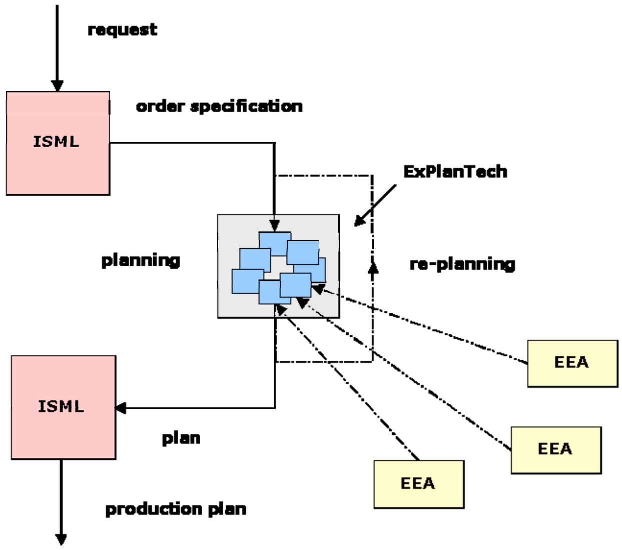

Fig. 1. Role of ExPlanTech in LIAZ

If from any reason it ceases to operate, the human user – a project planning engineer – can develop the plan by himself and insert it into ISML.
However, once a **request for an offer** arrives to the factory, a project planning engineer stores it in the ISML database. After having specified the necessary production components, the respective **order specification** gets uploaded into the ExPlanTech. This system is supposed to suggest particular resource allocation with respect to the order and planned operations of the company. This results in a distributed plan within the community of agents and its numerical and visual representation. Had the planning engineer liked the plan (i.e. deadline, costs, loads of machines) he/she downloads the **plan** back to ISML, where it is stored and further manipulated. Alternatively, the planning manager can manipulate the capacities of workshops, change priorities of orders, cancel other planned orders in order to form a plan as he/she likes. Each update triggers the re-planning mechanism that will keep all the plans logically consistent. Once the planning manager is happy with it, the overall distributed plan is propagated into the ISML. There are several communication flows between the ISML and the ExPlanTech:

- order specification: ISML –>ExPlanTech
- plan specification: ExPlanTech –> ISML
- set of plans' modifications: ExPlanTech –>ISML
- resources modifications: ISML –> ExPlanTech and ExPlanTech –>ISML

There are two aspects of integrating the company ERP into a MAS solution (or vice versa): **Logical Integration**, where it has to be decided upon functionality of the integrated components, relationships among components, semantics of communication and its flow, etc.

On the other hand when discussing **Physical Integration** we mean making the multi-agent system and the company ERP system interoperable in terms of physical communication – ACL, content languages, communication protocols and mechanisms (see Figure 2).

2.2 ExPlanTech Logical Integration

The production planning process in LIAZ is simpler than envisaged in the original ProPlanT system. PA agents represent workshops. The amount of variation how a plan may be implemented in various workshops is limited (the manufacturing process is rather deterministic). The question for the system is not how to plan the order optimally, but how will the order- if accepted- affect the other planned/potential orders.

The intra-enterprise ExPlanTech community in LIAZ consists of the Configurator Agent – an instance of PPA inheriting also some properties from the PMA class, several Workshop Agents – instances of a PA, and finally a Database Agent, also an instance of PA agent.

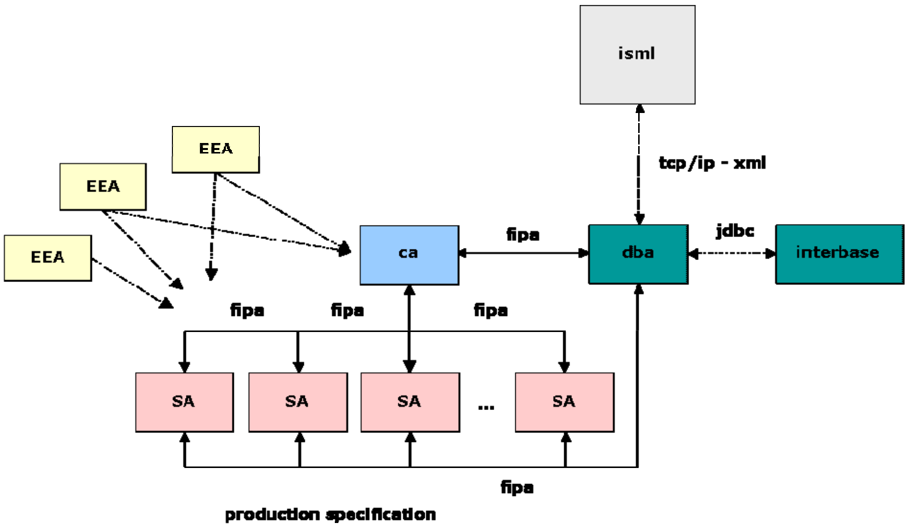

Fig. 2. The entire solution in LIAZ Pattern Shop *(The figure depicts the architecture of the entire solution in LIAZ. ISML stands for the ERP system in LIAZ. The other blocks represent the multi-agent system ExPlanTech, CA is the Configurator Agent, SA is the Scheduler Agent, dba is the Database Agent, the block called "interbase" represents the local database and EEA stands for the instance of the extra enterprise agent – the Monitor Agent. There are also communication and data flows figured, where FIPA means the inter-agent flow of FIPA messages, TCP/IP is a communication channel between the ISML and the system ExPlantech.)*

We did not implement the ProPlanT-like meta-agent as the JADE environment [1] provides the sniffer agent that sniffs the messages and visualizes what is going on in the community.

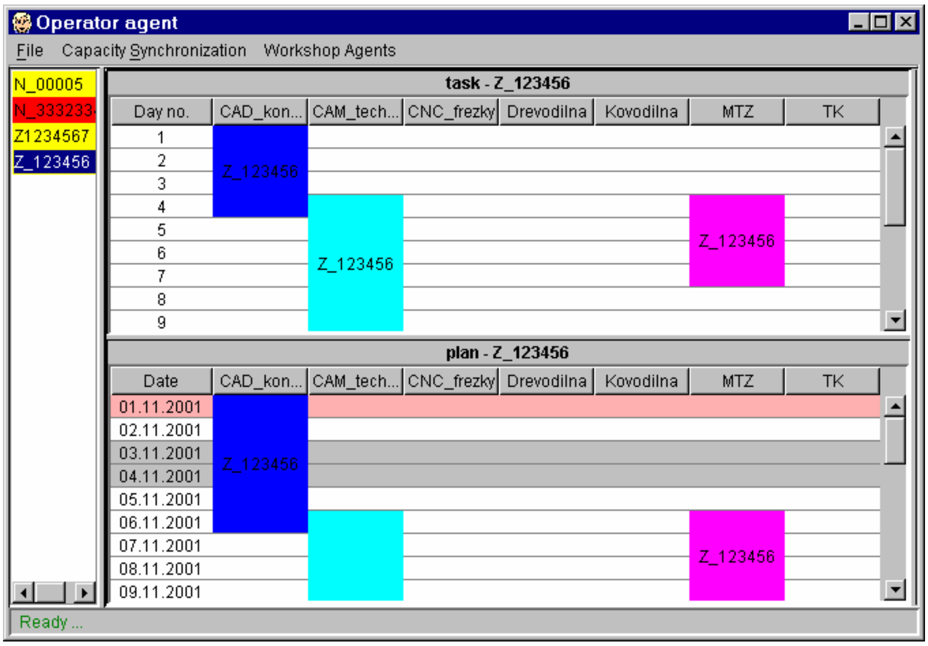

Fig. 3. Screenshot of the Configurator Agent (the *leftmost column represents the list of orders, order definition in the relative time window is placed in the upper part and the planned order in the lower part of the window)*

2.2.1 Database Agent

The Database Agent (DBA) plays several roles during the lifecycle of the ExPlanTech agent community. The main one is to supply other agents with production data (orders, preorders, calendar, workshops), the DBA is a communication bridge between the ExPlanTech system and the ISML. The data in XML format flow both ways between the systems via TCP/IP connection. On one side, there is sent information describing production (orders, pre-orders) from ISML to ExPlanTech, and on the other side, ready plans are sent to ISML. Received XML documents are preprocessed and then delivered to other agents (Configurator, Workshops). Preprocessing includes parsing, data extraction, storing into the local database (Interbase) and creating new XML contens for inter-agent messages. This agent doesn't have its own GUI interface. The local database, which is maintained by DBA, mirrors agents' data (knowledge) and also maintains other auxiliary data that are necessary for proper functionality of the system.

2.2.2 Configurator Agent (CA)

This agent stands over the group of Scheduler Agents and provides them with production data about orders. The agent itself shows two plans: a) an order before planning in relative time, and b) an order after planning in absolute time (see Figure 3). The CA agent contains a simplified 3bA model defined in the ProPlanT. It aggregates the functional aspects of both the PPA and the PMA agents. The 3bA model facilitates the agent community behavior and decreases communication traffic. It owns the information about workshops (their load, free spaces in a plan) and tasks (deadlines of orders etc.).

2.2.3 Scheduler Agent (SA)

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

2.3 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**.

2.3.1 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 (see Figure 4). According to his/her access rights the customer is able to trace his/her orders and watch their statuses (whether the producer can make the 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 expected to be used by the managers 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.

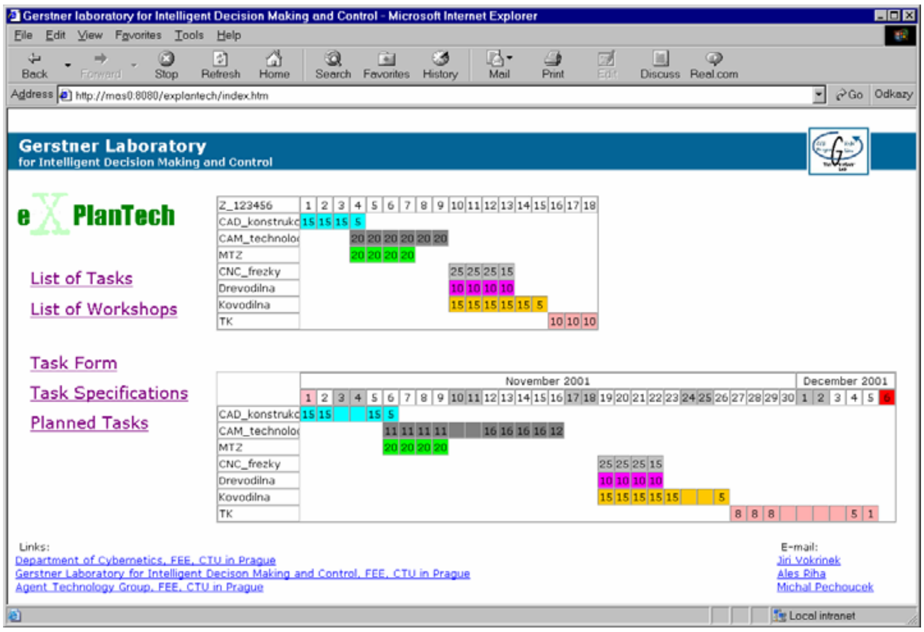

Fig. 4. Monitor Agent (*The upper table shows the order specification in the relative time and the table below represents the particular schedule for this order.)*

2.3.2 Resource Agent

This agent should work on the side of enterprise suppliers or cooperators (in the case of outsourcing). A partner enterprise (organization) could announce the status of services and resources, which are currently available. This shared (public) knowledge should be prepared in a standard form with respect to the 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 of different parameters (delivery dates, prices, amounts). The ExPlanTech could contract such an agent in case there is not enough resources and postponing of the deadline is not possible.

3 Exploration of Social Knowledge in ExPlanTech

3.1 Subcontracting, Cooperation

Even though the production planning in LIAZ has very little variability within the factory, there are possibilities to subcontract various cooperating enterprises, which may take over a part of an order. The LIAZ management wants the ExPlanTech to provide it with an analysis of an order that is subcontracted in part. This option significantly increases variability of the planning process and thus reveals potential for the 3bA decomposition and coordination. Thus, the concept of social knowledge-based contraction is explored in the extra-enterprise coordination.

3.2 Forward/Backward Planning Strategy

The configurator (or any other PMA) may also exploit the stored social knowledge differently. In communities, where the number of PMAs is not critical, the agent, which is responsible for the task decomposition may use the information stored in the acquaintance model for suggesting the most profitable request. Had the configurator known about availability of possible workshops, it may ask the contractor for keeping the proper deadline by when it wants the task to be finished. Suggesting inappropriate deadline may end up in either (i) workshop finishing its part too late even though it can be accomplished faster or (ii) workshop failing to accomplish its part at all. The decomposing agent (configurator) will than have all possible information to decide whether to ask the contractor to plan its part according the **forward** or **backward** planning policy (sometimes called eager and lazy) (see Example 1). While forward planning tries to plan all the tasks as soon as it can – it starts from the start-time and the preceding tasks, the backward planning plans tasks as lately as possible – it plans from the due time and allocates the depending tasks first.

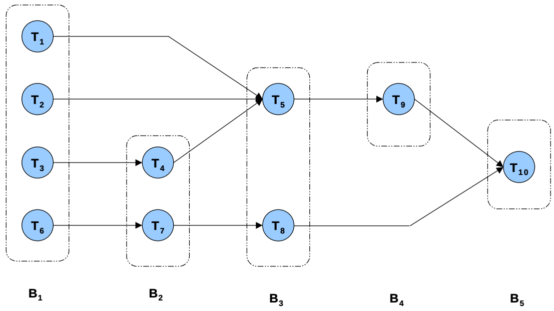

Fig. 5. Forward planning strategy

Example 1: There is a forward and backward implementation of a nonlinear plan $\{T_1 \rightarrow T_5, T_2 \rightarrow T_5, T_3 \rightarrow T_4, T_4 \rightarrow T_5, T_6 \rightarrow T_7, T_7 \rightarrow T_8, T_5 \rightarrow T_9, T_9 \rightarrow T_{10}, T_8 \rightarrow T_{10}\}\rangle$where $\rightarrow$ is a symbol for a time precedence. Figure 5 depicts the forward plan described by a totally ordered set of sets $\{B_1 \rightarrow B_2 \rightarrow B_3 \rightarrow B_4 \rightarrow B_5\}$ of a task. Tasks in each of the sets can be carried out in parallel while no task from a set B_2 can be executed before a task from a set B_1 provided that $B_1 \rightarrow B_2$. The forward instantiation of the nonlinear plan is given as follows:
$\{\{T_1 \mid T_2 \mid T_3 \mid T_6\} \rightarrow \{T_4 \mid T_7\} \rightarrow \{T_5 \mid T_8\} \rightarrow \{T_9\} \rightarrow \{T_{10}\}\}$,
where the | symbol stands for tasks implementable at the same time. Analogically, Figure 6 shows the backward solution corresponding to the sequence:
$\{\{T_3\} \rightarrow \{T_1 \mid T_2 \mid T_4 \mid T_6\} \rightarrow \{T_5 \mid T_7\} \rightarrow \{T_8 \mid T_9\} \rightarrow \{T_{10}\}\}$
Here, the tasks that have no "direct follow-ups" are executed as late as possible in order to minimize the overall "waiting" time of semi-products.

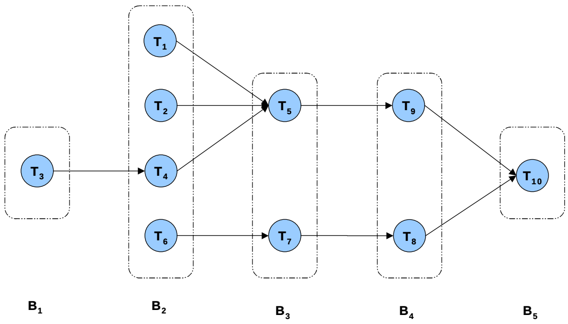

Fig. 6. Backward planning strategy

4 Implementation

We were looking for such an environment that would a) allow implementation of interoperable agents, b) standardize their communication and agent management, c) suggest ontologies and communication protocols, etc. FIPA (Foundation for Intelligent Physical Agents) [6] provides a set of such standards and reference implementations that have emerged from industrial needs and achievements of the research community. Each FIPA agent must be registered on the 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 the 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 the agents resident in an agent platform. It stores an 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 among agents within the platform and to agents resident on other platforms. DF provides a “yellow pages” directory service for the agents. It stores descriptions of the agents and the services they offer. FIPA also defines the structure of ACL (Agent Communication Language), which consists of several mandatory parameters such as for example “sender”, “receiver”, “content”, “language” (content language) etc. As a content language there is suggested for instance SL0. In our case we exchange messages with the content encoded in the XML format.
The Extra Enterprise Agent is an extended JADE agent (see Section 4.1) similar to the intra enterprise agents in the ExPlanTech. We use the technology of servlets for assuring remote access to the ExPlanTech community (see Figure 7). The servlet class is located on the standard web server. When the request comes from the static HTML page the servlet starts running. First, the Monitor Agent which communicates with the rest of the community by a servlet, is created. The servlet role is to dynamically create the HTML page according to the data received from the community via the Monitor Agent.

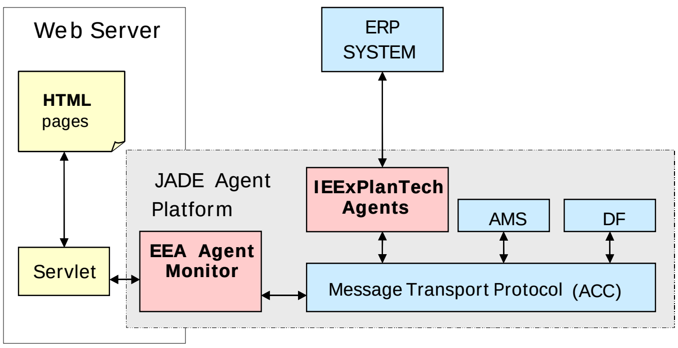

Fig. 7. Architecture of the ExPlanTech implementation

4.1 JADE

There exist several software implementations of FIPA, for instance JADE[1] (CSELT-Centro Studi e Laboratori Telecomunicazioni S.p.A), FIPA-OS (Nortel Networks), ZEUS. The ExPlanTech system is built using the JADE software framework, which is fully implemented in the 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 an important advantage of the JAVA solutions. JADE has several interesting features that at least make the process of implementation easier. One of these features is the availability of the Sniffer Agent that enables the user to observe the message flow among agents. The Sniffer Agent fully substitutes the role of the meta-agent used in the ProPlanT technology.

4.2 XML

XML is actually a language for creating markup languages that describe data and rules about the data. It requires applications to be defined 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 head towards defining representative tags and rules for business software component interoperability in the supported applications.

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

4.3 Physical Integration of the ExPlanTech in LIAZ

The most standard solution for integrating the ISML with the ExPlanTech would be implementing FIPA-ACC (Agent Communication Channel) on the ISML side. By doing so, the ISML would be agentified and encapsulated within FIPA-ACC (which would serve as a wrapper). That would result in ISML being a complete FIPA compliant agent that may fully participate in the community. However, implementing the FIPA-ACL for FoxPro did not seem to be a perspective step (large resources required for this development). Instead we have used the DBA agent as a stand-in agent that represents the ISML to the ExPlanTech agents and communicates with the ISML using a predefined mechanism. Therefore, the ISML has no social awareness and it communicates exclusively with the DBA agent. The DBA – ISML communication flows are carried out through sockets via TCP/IP protocol. There is no standard ACL used in between the components and they exchange messages only using XML with predefined semantics.

[1] http://sharon.cselt.it/projects/jade/

5 Conclusion

Creating the software solution for planning of tasks is a very complex problem. We have faced a little bit different task. We already had the production-planning tool (ProPlanT) and the task was to reuse the ProPlanT technology in the LIAZ Pattern Shop enterprise. The crucial problem is to find common language with the people from the factory and it means to understand the whole production process very well. We defined the real task, which is to prepare the production plan for a long time horizon. The ExPlanTech gives advices and views on the production, how it could look like in the future if some order is going to be accepted or refused. It provides the information about workshops' loads, creates plans with respect to deadlines, priorities and precedence dependencies of orders. One subtask was to incorporate our system into the IT solution represented by the information system ISML that is a source of data for our planning tool. We solved this task by developing a DBA agent that serves as an interface between the ISML and the ExPlanTech. This interface DBA agent transforms messages received via TCP/IP into FIPA messages and vice versa. The ExPlanTech itself was implemented in the JAVA language and the FIPA compliant development tool JADE was used as a multi-agent framework.

There is an interesting problem, whose solution would improve the planning process on the side of the LIAZ Pattern Shop. The task is to observe "gaps" in the production plan and according to the previous experience to find the most suitable order (a typical order) that would fit the best and to contact the typical customer with a special offer. In the future, we plan to use machine learning mechanisms for this purpose.

6 Acknowledgments

This research work is supported by the European Commission contract No. IST-1999-20171 (ExPlanTech) and by the Ministry of Education, Youth and Sports of the Czech Republic within the frame of the project No. MSM212300013. The authors also wish to acknowledge a substantial help and assistance of Dr. Vojtěch Pražma and Jaroslav Körner from the LIAZ Pattern Shop Ltd.

References

1. Bellifemine, F., Poggi, A., and Rimassa, G.,: Developing Multi-agent Systems with JADE. In: *Seventh International Workshop on Agent Theories, Architectures, and Languages (ATAL-2000)*, Boston, MA, 2000.
2. Camarinha-Matos, L.M., Afsarmanesh, H., and Lima, C.: Hierarchical Coordination in Virtual Enterprises. *Journal of Intelligent and Robotic Systems,* vol. 26 (1999), Issue 3/4, pp. 267-287, ISSN 0921-0296.
3. Cao, W., Bian, C.-G., and Hartvigsen, G.: Achieving Efficient Cooperation in a Multi-Agent System: The Twin-Base Modelling. In: *Co-operative Information Agents* (Kandzia, P., Klusch, M. eds.), LNAI No. 1202, Springer Verlag, Heidelberg, 1997, pp. 210–221.

4. Chandra, P.,Fisher, M.L.: Coordination of Production and Distribution Planning. *European Journal of Operational Research*, 72(1994), pp.503-517.
5. Finin, T., Labrou, Y., and Mayfield, J.: KQML as an Agent Communication Language. In: *Software Agents (Jeff Bradshaw ed.)*, MIT Press, Cambridge, MA, 1995.
6. FIPA: Agent Management*. In http://www.fipa.org*, Geneva, Switzerland, 1998.
7. Kubalik, J., and Lazansky, J.: Genetic Algorithms and their Tuning. In: *Proceedings of CASYS '98 (Dubois D., ed.)*, Liege, Belgium, 1998, ISBN 1-56396-863-0, pp. 217-229
8. Lowe, H., Pechoucek, M., and Bundy, A.: Proof Planning for Maintainable Configuration Systems. In: *Artificial Intelligence for Engineering Design and Manufacturing*, Cambridge University Press, Massachusetts, 12, No. 4, ISSN 8900604, September 1998, pp. 345-356,
9. Marik, V., Pechoucek, M., and Stepankova, O.: Social Knowledge in Multi-Agent Systems, In: *Multi-Agent Systems and Applications (M. Luck et. al, eds.)*, LNAI 2086 Tutorial, Springer-Verlag, 2001, pp. 211-245.
10. Marik, V., Pechoucek, M., Stepankova, O., and Lazansky, J.: ProPlanT: Multi-Agent System for Production Planning. *Applied Artificial Intelligence*, vol. 14, No.7(2000), pp.727-762.
11. Maturana, F., Shen, W., and Norrie, D. H.: MetaMorph: An Adaptive Agent-Based Architecture for Intelligent Manufacturing. *International Journal of Production Research*, vol. 37, No. 10(1999), pp. 2159-2174.
12. Pechoucek, M., Marik, V., and Stepankova, O.: Role of Acquaintance Models in Agent-Based Production Planning Systems. In: *Cooperative Information Agents IV (M. Klusch L. Kerschberg, editors),* LNAI No. 1860. Springer Verlag, Heidelberg, 2000, pp. 179-190.
13. Pechoucek, M., Marik, V., and Stepankova, O.: Towards Reducing Communication Traffic in Multi-Agent Systems. In: *Journal of Applied System Studies*, Cambridge International Science Publishing, ISSN 1466-7738, Cambridge, UK, No.1, 2001.
14. Shen, W.: Issues in Developing Agent-Based Collaborative Design and Manufacturing Systems. In: *Proceedings of the 5th International Conference on Information Sytems Analysis and Synthesis*, 1999.
15. Wielinga, B. J. and Schreiber A.:Configuration-design Problem Solving. *IEEE Expert,* 12(2), 1997, pp. 49-56.