

Multi-agent Product Life Cycle Environment. Interoperability Issues

Yulia Yadgarova, Victor Taratukhin, Ekaterina Skachko

► **To cite this version:**

Yulia Yadgarova, Victor Taratukhin, Ekaterina Skachko. Multi-agent Product Life Cycle Environment. Interoperability Issues. 6th International IFIP Working Conference on Enterprise Interoperability (IWEI), May 2015, Nîmes, France. pp.101-112, 10.1007/978-3-662-47157-9_10 . hal-01438394

HAL Id: hal-01438394

<https://hal.inria.fr/hal-01438394>

Submitted on 17 Jan 2017

HAL is a multi-disciplinary open access archive for the deposit and dissemination of scientific research documents, whether they are published or not. The documents may come from teaching and research institutions in France or abroad, or from public or private research centers.

L'archive ouverte pluridisciplinaire **HAL**, est destinée au dépôt et à la diffusion de documents scientifiques de niveau recherche, publiés ou non, émanant des établissements d'enseignement et de recherche français ou étrangers, des laboratoires publics ou privés.



Multi-agent product life cycle environment. Interoperability issues

Yulia V. Yadgarova¹, Victor V. Taratukhin²
and Ekaterina N. Skachko¹

¹ The Bauman Moscow State Technical University, Moscow 105045, Russia,
y.yadgarova@bmstu.ru, ekaterina_skachko@inbox.ru

² European Research Center for Information Systems (ERCIS), Leonardo-Campus 3, 48149
Muenster, Germany,
victor.taratoukhine@ercis.uni-muenster.de

Abstract. The main goal of this paper is to describe the integrated approach to product lifecycle management in the context of enterprise information system landscape. Product lifecycle management is a part of the common information area in the enterprise. In this work, the analysis of enterprise interoperability problems was presented. To classify such problems and solution approaches we use a framework for enterprise interoperability, described by D. Chen at [1]. The paper suggests reference architecture for product lifecycle management systems based on multi-agent concept. It promotes understanding of the interrelationships of different lifecycle stages for acquiring and manipulating concurrent engineering knowledge.

Keywords: Future product lifecycle management, enterprise interoperability, intelligent design and manufacturing, multi-agent framework.

1 Introduction

Nowadays the concept of the enterprise interoperability plays a major role in development of high-technology products. Complex industrial production in aerospace industry, rapid development of automotive and electronic industries requires management and collaboration of thousands of different suppliers and manufacturers. Industry 4.0 concept [2] implies reduction of time, growth of complexity and values of production and rapid customization of the production lines. Different enterprise architectures and wide range of business capabilities lead to increasing of complexity and volumes of data. This extensive development has limitations associated with computational complexity. For example, change management and configuration management are performed via workflow coordination and agreement of several responsible persons distributed over the workflow (due to their roles in the project). If quantity of versions of the parts exceeds ten, a lot of work and agreements between several roles arise, as well as a lot of down time, which, therefore, slows down the main design process. In this research

The paper analyses different types of product lifecycle management approaches and common enterprise interoperability problems for such systems and suggests multi-agent reference architecture of such system. Different phases of product lifecycle require different architectures of the agent and semantic-based protocols for their negotiation. It provides understanding of the interrelationships of different lifecycle stages for acquiring and manipulating concurrent engineering knowledge and processes.

2. Product lifecycle management in the context of enterprise interoperability

Automation of discrete stages of product lifecycle is being developed since 1970. This process includes developing programs for automated design (Computer-aided design, CAD) and manufacturing (Computer-aided manufacturing, CAM), also office and accountant's programs. With the help of information technologies, evolving since 1980, the new step, a concept of FMS (Flexible manufacturing system), was reached. In the end of 80-th - beginning of 90-th the concept of PDM (Product Data Management) and PLM (Product Lifecycle Management) was developed. PDM is the system that stores data about production process and has an interface with CAD/CAM systems. Development and integration of these systems leads to arising of PLM concept [6]. Years ago, PLM concept was understood as an integration of marketing, design, maintenance and service phases of product development [6]. However, at present time PLM systems control the overall process of developing and maintenance of production in the factory including control of innovations, configuration management and change management processes. In other works, current PLM task is not only automation of production process, but business concept for effective approval of the whole lifecycle processes. This concept, based on building an information model of the product and production process as well as workflow processes, allows collective design, improving of production processes and simulation of innovations on each stage. PLM integrates several approaches: PDM concept, collective design, digital factories. This concept focuses on the industry solutions and uses several technologies and methods. The main functions of PLM system [7] are:

- Manage CAD and process documents
- Provide an electronic file repository
- Include "attributes" - built-in and custom part and document metadata
- Construct and control bill of material (product structure) records
- Identify materials content for environmental compliance
- Change and workflow management
- Control multi-user secured access ("electronic signature")
- Export data for ERP (Enterprise Resource Planning) systems.

The model of the product lifecycle management concept is presented at fig. 1. Described functions are available on each stage of production process.

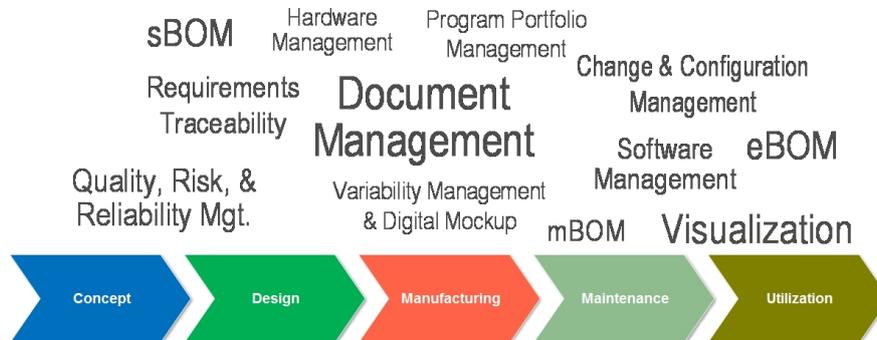


Fig. 1. Product Lifecycle Management concept [7]

One of the main functions of a PLM system is data exchange and integration between other enterprise services like MES (Manufacturing Execution System), ERP, CRM (Customer Relationship System) [7]. The location and link between these systems is presented on Fig. 2. Providing interoperability between these products is a complex task, which can be solved by using similar data models, connectors, etc. We analyze the interoperability problems appearing in the PLM systems and classify these problems according to enterprise interoperability framework, described at [1].

2.1 Enterprise interoperability problem space in PLM domain.

There are three main barriers with the interoperability of exchanging information:
Conceptual - syntactic and semantic incompatibility.

Technological - incompatibility of IT architecture and platforms.

Organizational - incompatibility of organization structure and management techniques implemented in different enterprises.

Several interoperability concepts was presented in the framework:

Interoperability of data - ability to operate together different data models which can locate on different machines with different operating systems.

Interoperability of services - refers to operate together different applications with syntactic and semantic differences.

Interoperability of processes - aims to make various processes work together.

Interoperability of business - refers to work in a harmonized way at the levels of organization and company.

The PLM system works mainly in intra-level of the enterprise and we define a specific problem space within product lifecycle managed domain in the context of interoperability:

1. Interoperability of data in the PLM domain: Within PLM system, this problem is solved by using the integrated approach (a common format for all information models, single database (for example, Windchill PLM architecture [8]). So, there are

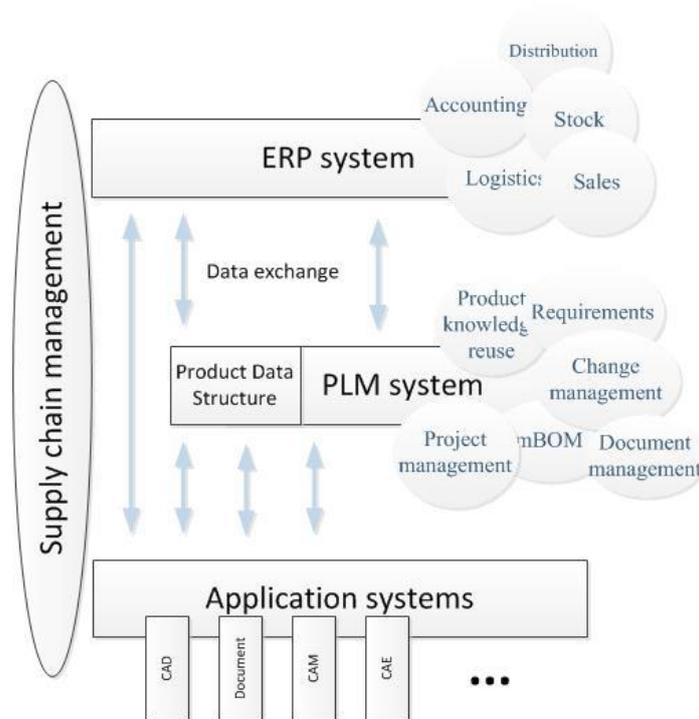


Fig. 2. Location of PLM system in the whole enterprise landscape

no problems with data interoperability within a single PLM system. But the problems appears when we aim for change a PLM vendor or integrate a PLM system in the whole information landscape (link with ERP, MES, CRM systems). Then there are syntax and semantic problems. In addition, there are several problems in technological and organizational interoperability between PLM-systems of several organizations: different information models and attributes of these models leads to mismatches in product definition (product structure).

2. Interoperability of services in the PLM domain: There is also a problem with integration between PLM and other systems. Also, interoperability of services becomes a difficult task when we integrate organizations in single virtual enterprise with different systems and vendors of systems. As an example: integration with PLM services based on SOA with accounting system with strict architecture.

3. Interoperability of processes in PLM domain: There are organizational barriers in the enterprise structure and conceptual interoperability with other systems and PLM solutions. Process description models in several PLM systems are different only if the product is same.

Our approach to problem of building reference architecture of the PLM system is based on the multi-agent concept of lifecycle stages. This reference architecture can be a good foundation both for building informational landscape of the single enterprise and for building virtual enterprise architecture. Loose coupling, unified

interfaces and protocols in agent systems' architecture allows build a good reliability solution for linking enterprise system parts.

3. Multi-agent PLM concept

One of the most appropriate technologies for developing large complex distributed systems is multi-agent concept. One of the benefits of multi-agent systems is their decentralization and simplicity of development of the agents. Also synergetic effect of such systems can be achieved. Agents, responsible for small simple parts of the system with negotiation with each other, can keep system status and achieve complex objectives together. Several common characteristics of the multi-agent systems (MAS) are:

- No explicit external control - system must be independent of external control unit
- Global order from local interactions - ability to achieve global order through local interactions
- Distributed control - in such systems control is distributed throughout the whole system. No central decision node is presented
- Robustness - self-organized systems are robust. System should thrive on randomness and fluctuations
- Adaptivity - self-organization is dynamic process. The system needs to be dynamic and reconfigurable
- Non-linearity - no direct relation between the fluctuations of the environment and system behavior [9].

Also one of the key trends in manufacturing is the ability of machines and devices to be self-organized, to communicate independently with each other and to provide agile and adaptive design and manufacturing environment.

Multi-agent systems allow to distance from the strict workflow process among the development and production processes. In proposed MAS, the first physical and second syntactic levels are well standardized by Foundation for Intelligent Physical Agents (FIPA [10]), but communication between agents in such systems builds based on semantic meaning of the message. To understand received block of information agent could parse this message at semantic level, the symbols must be understood in the same way. So, to communicate different agents with each other we use ontology describing approach. Building ontology depends on concrete specific of the enterprise and must be shared or explicitly expressed and accessible to be able to decode the information. FIPA standards includes Ontology Service Specification, which describes usage of ontologies.

According to ISO 14258 [11] there are three basic ways to relate entities together:

1. *Common format* - used in presented PLM architectures.
2. *Unified format* - at present time used in integration processes between other information systems
3. *Federated format* - no predefined common format - used in proposed approach within MAS.

The key tasks for building robust distributed multi-agent PLM environment are:

- Build general reference architecture of multi-agent system. Each stage of PLM-concept environment has different functions and attributes. Each stage of PLM-concept environment has its own agent architecture.

- Develop communication protocols between agents in similar layers and MAS with different types of agents (Horizontal and vertical integration).

- Define a data model for each stage for further integration and exchange.

- Define problems of negotiation on similar lifecycle layers and develop semantic negotiation technics based on domain ontologies for agents for conflict resolution.

A general several-layer architecture of multi-agent system with Design (DA), Manufacturing (MA), Support (SA) and Control/Change agents is presented at fig. 3.

3.1 MAS of Design phase

Concept of product as MAS was presented by [5, 12]. System of Design agents (DA) models the Design phase of product lifecycle management and represents the overall design view of the assembly (fig. 4). Each DA represents simple part of the product. Each DA has a set of states presented by technological con-versions. Present complex assembly of parts (parts of the large high-technology systems) consists of thousands simplest decomposed parts that have their own set of versions, alternates and other attributes. Every part interact with others to provide the whole assembly functionality. The utility function of the agent through negotiation process is complexity of the part. The cooperative utility function of the agent is complexity of the product $STC_{product}$

Every design agent is responsible for a simple finite part of assembly. This means that we can describe assembly as MAS. In this system, negotiation of DA represents mismatch detection in the assembly part and minimization of utility function. Every change request to this system changes the whole multi-agent equipment and leads to negotiation procedure.

In design process the part characterized by the set of elements:

$$Part = \{E_i \mid i = 1, 2, \dots, p\} \quad (1)$$

Each element E_i can be one of the several class of elements of $e_z \in E_z$ and has the set of parameters

$$P = \{P_i \mid i = 1, 2, \dots, p\} \quad (2)$$

At this, set parameters can be one of the part's class parameters (for example, length of the surface) and concrete object parameters.

We define Complexity characteristics of the part (Structural technological complexity):

$$STC_p = f(E, R_p, K_m) \quad (3)$$

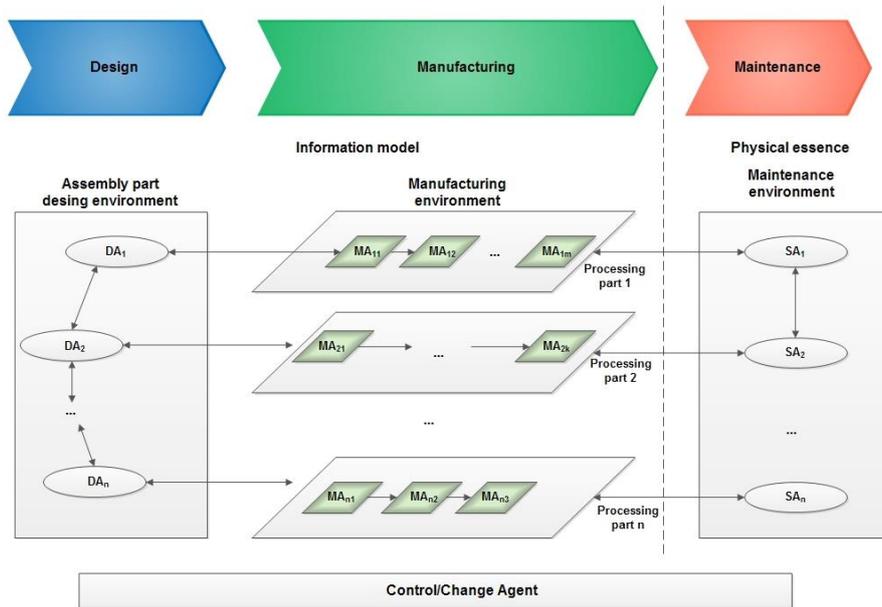


Fig. 3. Multi-agent Product Lifecycle System

So, Complexity characteristic of assembly is:

$$STC_a = f(S, STC_p, R, P, TC) \quad (4)$$

where S - set of assemblies, STC_p - set of parts, R - set of relations between structural parts, P - set of parameters, TC - set of technological conversions:

$$STC_{product} = \sum_{i=1} STC_a + \sum_{j=1} STC_p \quad (5)$$

MAS of Design agents has a strong link to MAS of manufacturing. It represents vertical integration of the product lifecycle stages and performs feedback between manufacturing and design layers

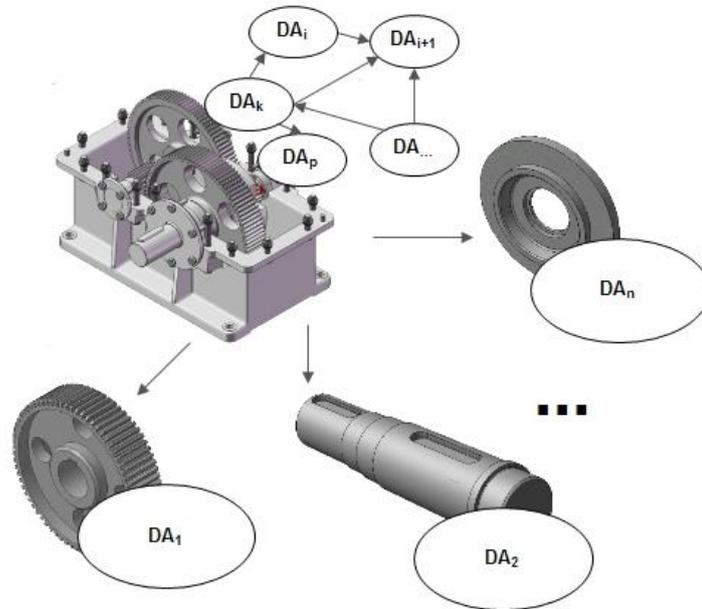


Fig. 4. Design agent's overview

3.2 MAS of Manufacturing phase

One of the main task in the manufacturing phase of production process is control and scheduling tasks in shop floor. The production Manufacturing Execution Systems (MES) provide mechanisms to control the manufacturing shop floor in real time. These functions in modern PLM systems are performed by the MES-module (preparing and control processing). Integration between MES and PLM means that data values from PDM are transferred into MES [13] and, based on this data, MES builds a schedule of the processing. The meaning of processing part is decomposition of the whole manufacturing process on several simple sub-processes [14]. The single agent in this phase models the simplest process from the decomposition. The sequence of the processes makes the technological process of the part.

At the task's entry on the shop or the task agent finds in the systems' database the relevant manufacturing process. The process interoperability is provided by independence of process model language stored in the database. Agent can works with ARIS diagrams, BPMN models and other concepts with multi-level decomposition (fig. 5). Ontology-based approach provides semantic of the messages. While this manufacturing process is specified with entry parameters of the part (information from design agent) and manufacturing capabilities. The relevant manufacturing agent (fig. 5) presents each manufacturing operation. After finding the corresponding set of the manufacturing agents, each of them performs its own

function. Each agent is responsible for finding the resources, sequence of processing and operations. The negotiation process in this system is specified in [14].

Each manufacturing process (set of MA) has a link to the design agent (Each DA is responsible for the simple part). Therefore, several MA linked with one DA form the vertical integration of this system. In addition, from another side, single manufacturing process can be described as MAS subsystem. This subsystem is the part of the whole Manufacturing MAS.

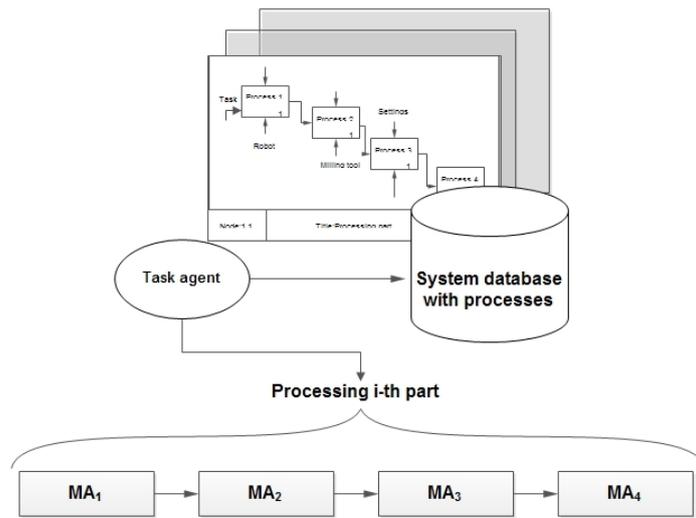


Fig. 5. Manufacturing agent's function overview

3.3 MAS of the Maintenance phase

In the maintenance phase, we have physical instance of the product. The main task of product lifecycle management in this phase is making a closest link between physical state of the product and its informational model [15]. Part of the multi-agent system responsible for the maintenance phase consists of the set of agents that represent information model of the product. Each maintenance agent has a one-to-one link to the DA and can store statistic and history data about product's functionality, maintenance, repair and other.

Every type of agents described below is under control of the Control/Change agent. Control/Change agent controlled the whole assembly part (one-to-one link) and is responsible for the changes and overall production process.

4 Evaluation and implementation of the system

Based on the below model the part of integrated MAS system was presented. MAS design agents subsystem is implemented as a module to PLM system Windchill (PTC company) [8]. Java-based application is based on Spring and JSP-technologies. System consists of several modules including agent platform, link to product information model, user interface and standard information module. MAS module is centralized and consists of several agents such as shaft model. The negotiation process between the agents starts manually and finds the overall configuration of the assembly with minimal constructional-technological complexity (fig. 6).

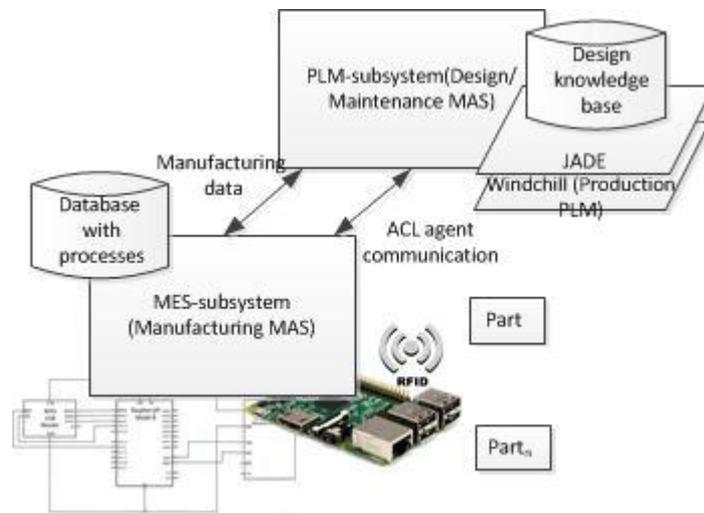


Fig. 6. Hardware-software architecture of the system

Among the variety of MAS designs, we choose FIPA standards, which describe the overall architecture of the MAS and agent's ways of interactions. The reference architecture of the design agent includes several layers. The first layer is communication module. The second layer is run-time module, which performs the transformation of product module part. The third level is control that performs the transformation of overall product module.

The manufacturing agent's layer is presented by hardware-software subsystem. In our subsystem, the universal circuit board can be embedded in any device, controlled by a UNIX class operating system, which contains an agent platform. This multi-agent architecture will support FIPA standards.

Agent communication is simulated at higher abstraction level than traditional data communication. Messages between processing devices, based on speech act theory (ACL-FIPA language) are transmitted across the network. Each module is capable of responding to a message from other agents by means of LEDs, connecting to other modules via IP network.

4.1 Position of MAS in the framework

Proposed architecture contributes to remote several barriers according to [1]. The main benefits are eliminating conceptual barriers concerning data (due to single database usage), processes (due to usage the unified ACL language for agent communication and negotiation agents' protocol) and services. The position of the proposed multi-agent architecture is described on fig. 7.

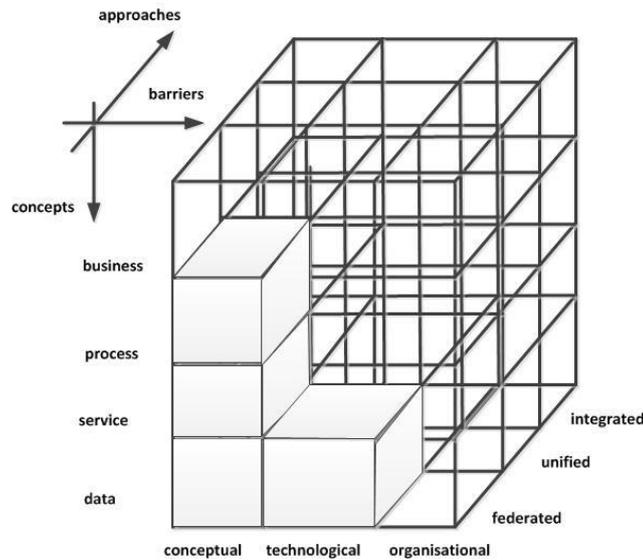


Fig. 7. Position of the MAS in the framework [1]

5 Conclusion

In this research, we describe the integrated methodology for building an enterprise architecture in the product lifecycle management domain. Applying this architecture allows eliminating conceptual and technological barriers [16], due to use of standard semantic and architecture technologies. In addition, this approach bases on the multi-agent concept and provides the class of intelligent systems that helps to perform the full management of engineering production lifecycle and exchange data between several enterprises. Semantic approach to communication between agents allows increase interoperability and communication both between enterprises as between information parts within enterprise. In the further work, we are going to present negotiation protocol between several multi-agent systems to extend this framework on virtual enterprise.

References

1. Chen, D.: Enterprise interoperability framework. <http://ceur-ws.org/Vol-200/19.pdf> Accessed: 2015-02-15.
2. BMBF: Zukunftsbild industrie 4.0. http://www.bmbf.de/pubRD/Zukunftsbild_Industrie_40.pdf Accessed: 2015-02-16.
3. Tarchinskaya, E., Taratoukhine, V., Matzner: Cloud-based engineering design and manufacturing: State-of-the-art. In: Proceedings of the IFAC Conference on Manufacturing Modelling, Management, and Control. (2013) 353-358.
4. Scheer, A.W., Nuttgens, M.: ARIS architecture and reference models for business process management. Springer (2000).
5. Taratoukhine, V., Bechkoum, K.: Towards a consistent distributed design: A multi-agent approach. In: Proc Information Visualisation. (1999).
6. Kolchin, A.: Product lifecycle management [Upravlenie zhiznennym tsiklom produktsii]. Anakharsis M. (2003).
7. Cimdata: All about plm. <http://www.cimdata.com/en/resources/about-plm> Accessed: 2014-12-15.
8. PTC: Windchill plm architecture. <http://ptc.com> Accessed: 2015-02-15.
9. Ferreira, J.: Bio-inspired Self-Organisation in Evolvable Production Systems. Tekn. Lic. PhD thesis, dissertation, Royal Institute of Technology, Sweden (2013).
10. Poslad, S.: Specifying protocols for multi-agent systems interaction. <http://www.fipa.org/subgroups/ROFS-SG-docs/2007-TAAS-specifying-MAS.pdf> Accessed: 2014-12-15.
11. ISO14258: Industrial automation systems-concepts and rules for enterprise models. In: ISO TC184/SC5/WG1. (April 14, 1999).
12. Evgenev, G.: Systemology of engineering knowledges [sistemologiya inzhenernykh znaniy]. M.: BMSTU 376 (2001).
13. Ben Khedher, A., Henry, S., Bouras, A.: Integration between mes and product lifecycle management. In: Emerging Technologies & Factory Automation (ETFAs), 2011 IEEE 16th Conference on, IEEE (2011) 1-8.
14. Skobelev, P.: Multi-agent systems for real time resource allocation, scheduling, optimization and controlling: industrial applications. In: Holonic and Multi-Agent Systems for Manufacturing. Springer (2011) 1-14.
15. Fedotova, A., Taratoukhine, V., Kupriyanov, Y.: Enterprise asset management systems: Methods and techniques for life-cycle management of high-value complex products. In: Very Large Business Applications (VLBA), Aachen, Shaker Verlag (2014) 69-78.
16. Chen, D., Vallespir, B., Daclin, N.: An approach for enterprise interoperability measurement (2008).