The Management of Manufacturing-Oriented Informatics Systems Using Efficient and Flexible Architectures

Constantin-Daniel AVRAM, Cecilia CIOLOCA, Laurentiu CIOVICA, Mihai CURTEANU, Mihai GEORGESCU
Academy of Economic Studies, Bucharest, Romania
costin.avram@gmail.com, cecilia_cioloca@yahoo.com, laurentiu.ciovica@gmail.com, mike@synirgy.com, mihai.georgescu@europe.com

Industry and in particular the manufacturing-oriented sector has always been researched and innovated as a result of technological progress, diversification and differentiation among consumers' demands. A company that provides to its customers products matching perfectly their demands at competitive prices has a great advantage over its competitors. Manufacturing-oriented information systems are becoming more flexible and configurable and they require integration with the entire organization. This can be done using efficient software architectures that will allow the coexistence between commercial solutions and open source components while sharing computing resources organized in grid infrastructures and under the governance of powerful management tools.
Keywords: *Manufacturing-Oriented Informatics Systems, Open Source, Software Architectures, Grid Computing, Web-Based Management Systems*

1 Introduction

Today, production lines are already working with powerful technologies based on intelligent robots, artificial intelligence and genetic algorithms. They will become capable of real-time equipments configuration and transformation in order to execute differentiated and personalized products. In order to translate technological innovations into profitable business, the authors consider that manufacturing-oriented informatics systems should use customer-oriented architectures that will allow clients to design their own personalized products, order these products and purchase them at acceptable prices. Such architectures will need increased computing power which is no longer a problem nowadays and it can get even better because of grid computing. The software architecture we propose for the manufacturing industry can be consider an extend ERP system and the original contribution is represented by the customer integration in the system functionality. Not only each specific software component will grow in complexity and functionalities but integration between commercial applications and open source components is necessary. This kind of integration means high business complexity and real-time communication between software applications. The authors consider that these issues could be solved using *ServiceOrientedArchitecture*. Because building and implementing a software architecture based on the customers' demands is a very complex process, the authors suggest that a web based project management system should be developed and used. Such system would eliminate or at least mitigate risks regarding communication, coordination, poorly understood requirements and poor time management.

Supporting manufacturing-oriented business with new informatics systems is a recent concern for the biggest software developers. At the IEEE conference, 2009 edition, ERP developers such as SAP, announced that they will analyze, together with the industrial segment and academic partners, the future of manufacturing-oriented organizations and the newest market trends and demands. At the IEEE conference, 2010 edition, there were discussions about reconfigurable production lines and their impact at organization, plant and shop floor level.

This paper is build on qualitative methods as participatory observation and case studies due to the authors' experience in the field of software development, software architecture, software integration, industry software solutions and project management.

2 Manufacturing-oriented informatics systems

The software applications within the area of manufacturing oriented industries had represented a point of interest since the early '60s. In the mid '70s the information systems for materials and raw materials requirements planning in the production process appeared, Material Requirements Planning, MRP. The MRP concept has expanded and evolved so quickly that in the same period appeared complex and powerful systems for planning and management of production resources. The concept is called Manufacturing Resource Planning, MRP II and is the foundation of today's ERP systems. The main function of an MRP II is to manage the necessary raw materials and materials in accordance with the production requirements, but the system also integrates functionality such as: accounting, sales and distribution, human resources etc.

The manufacturing process is automated through the *Manufacturing Execution Systems,* MES. The MES are responsible with the management and optimization of the manufacturing process from the moment the order is placed until the products are ready. The MES systems are used in the auto industry, pharmaceutical industry, textile industry, aerospace, medical equipments and any manufacturing oriented industry.

From [1] are identified the functions of the MES systems that are bond to production resource allocation, machines and equipments, materials and raw materials, execution technical documentations. It ensures the correct installation of the manufacturing equipments and their planning according to the established manufacturing scopes. The details and operations programming function establishes the work sequence based on priorities, attributes, characteristics and the technical documents associated to the manufacturing units for an operation such as establishing the form, the color sequence, etc. It can identify alternative and parallel operations to compute in detail the exact moment of equipment installation. The function of dispatching the manufacturing units allows the management of the manufacturing units flows in the form of jobs, orders or batches. This function supplies information about the sequences in which the operations are executed. Information change in real time depending on the events that happen on the manufacturing line. The dispatching function of the MES systems gives them dynamics as they can modify an already established program, can launch repetitive processing sequence if needed and can supply in any moment information about the point the production is in, *work in progress*, WIP. The MES systems allow the control of the documents needed by the manufacturing units such as execution instructions, technical documentations, reengineering details, sequence communication. They have the ability to understand the information about what has been planned and what has been realized. They allow sending the information to the next operations, supply data to the operators and technical documentations for the control devices. The MES systems can manage documents related to the control and integrity of the work environment, the regulations regarding health and safety and the correction procedures. All these information are stored as historical. The data collection function is an interface between operations and allows the gathering and transmission between operations of parameters and data saved during the manufacturing process. Data is then collected manually or automated from equipments. The work management function supplies the status of the staff involved in the manufacturing process, presence and participation reports, information about the indirect activities such as preparing materials and equipments. This function interacts with the function of resource allocation for an efficient planning. The MES systems fulfill the function of quality management through supply of analysis made on the data collected in the manufacturing process. The quality control is thus ensured through the problem identification, advices regarding problem correction, correlations between symptoms, actions and results aiming at determining the cause of the apparition of problems. The process management function monitors the production and can correct automatically the production line and can offer information needed by the operators for applying corrections to improve ongoing activities. This function is applied both between the processes of an operation and between different operations that run in an established sequence. It includes alarm signals meant to get the attention of the staff that certain measurements are off scale. The process management function behaves like an interface between the MES system and the intelligent equipments through the function of data collection and acquisition. Equipment maintenance management is the function that allows the following and targeting of activities for the equipment and instruments maintenance ensuring though availability for manufacturing process and periodic planning for maintenance. It sends real time alarms in the moment of problem identification. It keeps an historic of events and problems for each equipment to help diagnose future incidents. Genealogy is the function that monitors the execution stage products. It shows who is working at the product at a certain moment, the product's components, the used materials depending on supplier, lot, series number, current manufacturing conditions, signaled alarms or any other exceptions recorded for that product. The performance analysis function offers minute information regarding the current manufacturing operations compared with past manufacturing cycles and expectations for the current production cycle. They allow the reporting of indicators such as resource utilization, resource availability, time needed for a manufacturing cycle, the level of accordance with the planning and performance standards.

MES systems are the bond between the systems dedicated to the production lines, *Shop Floor Systems*, SFS and the organization's management systems such as ERP. The organizations tend to integrate more and more the IT systems such as ERP, Business Intelligence, Supply Chain Management, Client Relationship Management and the systems dedicated to the production lines to increase the efficiency. The objective of integrating an ERP systems with the manufacturing systems is to diminish the time needed for making an inventory, reduce the manufacturing cycles' duration and the time between order placement and finishing the products. This is made through complete data flows in the structure of organizations able to interact with all business functionalities from the operations specific to the manufacturing units, the management of the manufacturing orders up to the financial operations and those of programming and planning. The MES systems can reengineer the present architectures through the chaining of ERP, CRM, SCM applications and also DCS, *Distributed Control Systems*. A classic integration model of an ERP system with DCS system through the MES systems is described in Figure 1.

After an order is placed by customers, inside the ERP system a sales order is created. The ERP system can serve the order from stock, *make to stock* or by launching a manufacturing flow, *make to order*. If the client is served from the stock, the process is managed only by the ERP system. If the manufacturing of the ordered good is needed, the ERP system generates a production batch and sends the characteristics of type, quantity, delivery date towards the MES system. If the needed materials and raw materials are not available in stock, an order is launched towards the SCM, *Supply Chain Management*, to buy these elements from the suppliers. The order towards the suppliers is recorded in the ERP system as it will modify data from the materials management module and from the suppliers' accounting. When the acquiring of materials is over, the MES system can program and plan the resources aiming at consumption efficiency. During the execution of the product at the level of the manufacturing line under the management of the DCS, *Distributed Control System*, the MES system monitors permanently the manufacturing process and updates the ERP system. When the production is over, the product is packed, labeled and delivered to the buyer. This way the ERP system completes itself with the MES system through the real time monitoring of the execution level at a given moment, by ensuring quality management, by supplying real time metrics of the manufacturing lines. Sending this information towards the ERP systems ensures a context that is more correct and updated in taking decisions at an organizational level.

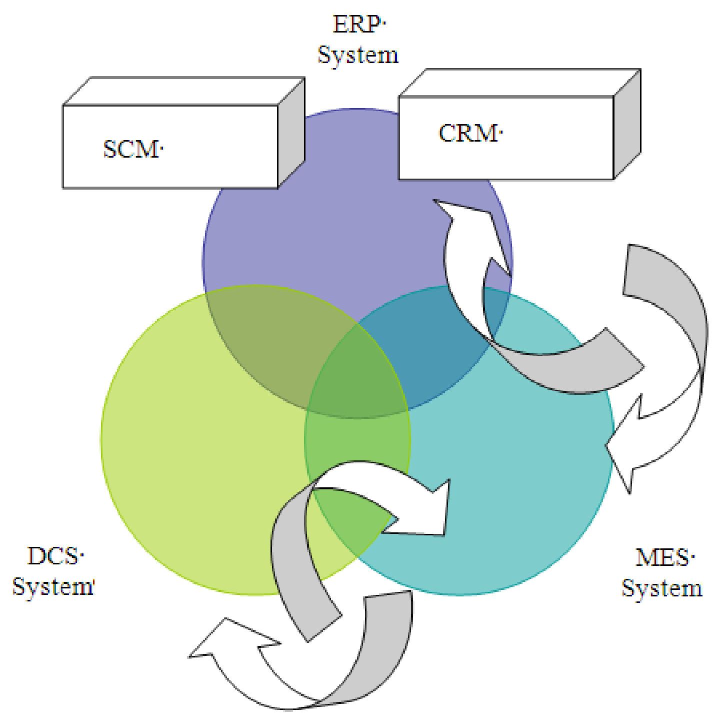


Fig. 1. Integrated platform ERP, MES, DCS

A series of limitations and disadvantages regarding the integration of various systems were identified in [3] and [4]. Furthermore, in [2] more difficulties are identified when integrating ERP systems with manufacturing systems because this raises new series of limitations that derive from the differences between ERP and DCS products, from the selection of the sharable information, from the format differences of data used by most systems, from reasons of organizational culture related to sharing information with confidential character such as technical documentations or products recipes. Problems rise regarding the security of information access and reengineering of the rights and roles in the system. When choosing an ERP system even if no integration with other systems is needed, a very serious analysis should be done first. Different criteria, as stated in [5], should be take into consideration before investing into an ERP implementation.

3 A customer-oriented software architectures for manufacturing systems

Lately, with the development of internet and the increasing business complexity of information systems, their real-time communication of software applications has becomeanecessity. This need has required development of new efficiency oriented architectures, able to transmit messages between components via the network. Among the architectures that have appeared on the market, RPC, Remote Procedure Call, based architecture and SOA, Service Oriented Architecture, are the most important. Unlike EJB and DCOM architectures, both based on RPC and which are dependent on the platform so that the interaction between components based on these architectures is very difficult to obtain, SOA architecture was designed based on the idea of interoperability. Likewise, SOA is based on the universal language for XML data representation, unlike other technologies that use proprietary languages for data representation. Interesting is also that SOA encourages communication between independent components unlike other models of architecture that focuses on sharing distributed objects via the network. To achieve a platform based on SOA architecture is necessary to evaluate and to select appropriate software products. In order to keep trace with the new industry requirements, the systems must become more flexible, configurable and very secure [7]. Figure 2 presents schematically a SOA architecture adapted for manufacturing industry.

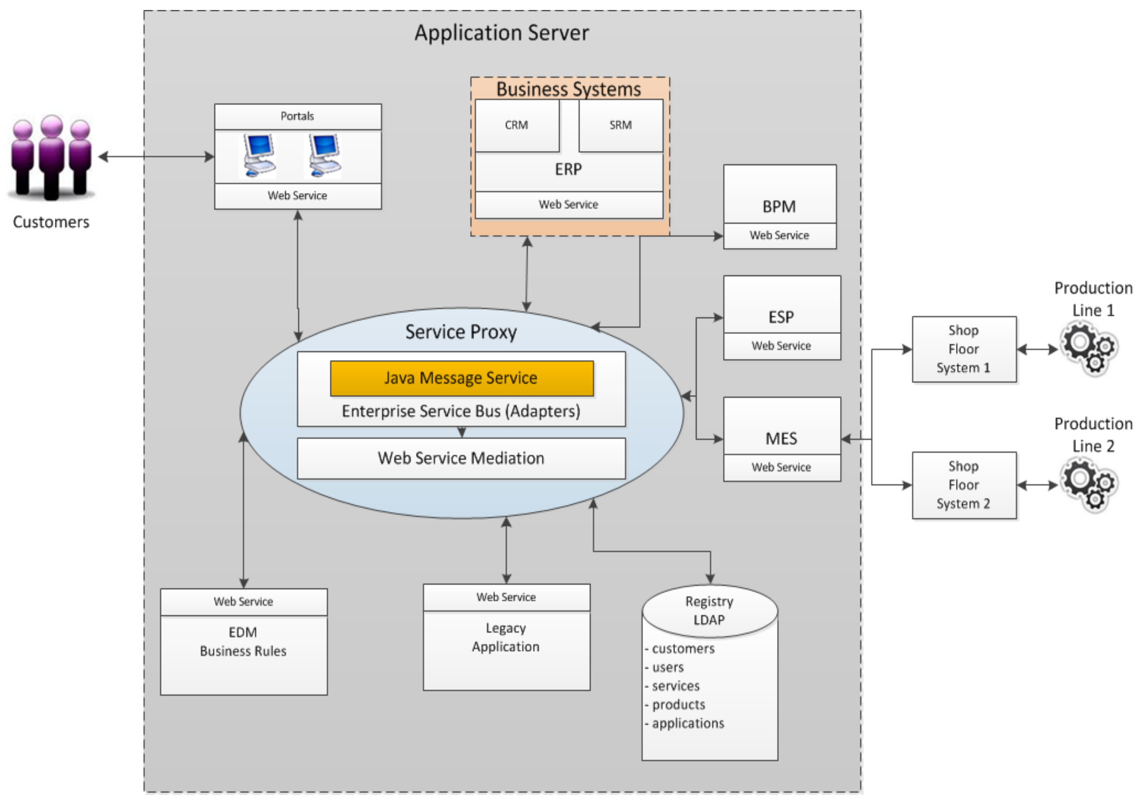


Fig. 2. SOA architecture customized for manufacturing systems

The central point of the architecture is the Proxy Server, which routes all the messages between the components of the architecture. The main advantage of this architecture is the autonomy of each component. All the systems in the architecture are independent, they act like separate applications, they can be located anywhere in the network and they communicate with third party application via web services. The architecture is an open one, new components can be added anytime without the need of changing anything to the existing components. This way the architecture becomes a flexible and configurable one, being able to be competitive on the market. Another step ahead is the Business Process management that allows the organization to improve and adapt its processes to the new requirements.

Next, we will shortly describe each component that makes up the architecture presented before:

- C1. Portals, represent the entry point for the customers, by using a user interface the end user is able to create a customized order and send it to the production;
- C2. Business Systems, contains applications like Enterprise Resource Planning, Customer Relationship Management, Supplier Relationship Management and Business Intelligence;
- C3. JMS, Java Massage Service, is an API used for sending messages between components. It is part of the Java Enterprise Edition and it enables components to create, send or receive messages. It is associated with the idea of asynchronous massaging, so that the requests, response couple in SOA are treated as messages. The mechanism that JMS uses for sending the messages is publishing the messages to a queue. The components are listening to the message queue. Once a message is consumed by a client the massage is no longer available on the queue;
- C4. WSM, Web Service Mediation, the role of this component is to translate messages from different type of protocols into one single general language so that the data can be redirected to the appropriate endpoint. It actually bridges all the protocols. It is also responsible for logging all the requests and responses. This component also acts like a Proxy Server, intercepting messages from third parties, longing the messages and providing a web service security envelope;
- C5. ESB, Enterprise Service Bus, its main role is to receive data from third parties components and system using different types of protocols. It contains a set of adaptors in order to be able to receive data using different types of formats. It interacts with the WSM, Web Service Mediation component in order to translate the data into one single and general format XML. The massages received from all the components are then routed to the JMS in order to be published on a specific communication queue;
- C6. BPM, Business Process Management, enables the organization to define processes by using visual flow steps. This component is the orchestrator of the entire architecture. By using BPM, the organization is able to adapt its processes to the new markets requirements in order to be competitive. As stated in [6], the fundamental impetus behind BPM is cost savings and improved business agility;
- C7. ESP, Event Stream Processor, the role of this component is to receive in real time a bulk of stream and to detect different patterns for the events. In order to detect patterns, ESP uses filters, time-based aggregations, joins and triggers. ESP input comes from JMS, Java Message Service, and the output of the ESP is used by the business activity monitoring;
- C8. EDM, Enterprise Decision Management incorporates a set of business rules and a tool for managing these business rules. A business rule is a set of conditions that have to be accomplished so that a certain statement is true. For example, in order to have access to the administration mode a user must have the admin role;
- C9. Registry LDAP is a repository containing that contains all the implementation artifacts. The data in this repository contains data regarding data, regarding customers, user, products, or even applications;
- C10. MES, Manufacturing Execution Systems, are responsible of the management and the optimization of the manufacturing process. They are the contact points for manufacturing lines and they integrate the manufacturing systems with the other components.

- S11. Shop Floor System are those applications that manage the production lines.

4 Open source components

Developing a software application requires time and resources. Large amount of resources are invested in commercial applications to obtain a high quality product. The need to develop a software product is based on the customers' need and implies the existence of a complex team of people, each with a well-defined role. The number of team members is limited by the company's resources involved in the development, thus completing the project varies accordingly to the team size. Once completed, the software product is used by the customer based on a license agreement which often has a rather high price.In parallel with commercial companies that produce software applications there is the open source community, which is composed of technical and non-technical people that contribute to the development of a software project without receiving a material reward. Because these people are not in the same location or even in the same country, the management of the project is done online, on a site dedicated to the project. There is a team that coordinates and integrates the work of each person involved. There are used open source tools for implementing, managing and testing the product. Open source software offers a high quality and a higher response time than the commercial applications.

As presented in [8], among the most important advantages of using open source software there are:

- availability of source code and the right to change it: allow unlimited modification and improvement of the application;
- the right to use the product in any way and redistribute the product together with any amendments;
- high stability - when the source code is public and a large number of people are involved, the product is tested extensively, the problems are detected and solved quickly, everything being visible in the source code;
- no information is hidden in the product; all information is visible and available.

A frequent question, even in the open source community is that whether open source models are more or less productive than proprietary ones. In many cases, proprietary software is profitable if it doesn't compete directly with a similar open source product. Moreover, open source software model has shown that once it enters a niche segment with sufficient energy, it can develop a product able to compete with any other similar proprietary software.

A company that wants to develop a new product should consider launching the product under open source license if in the same field there is already a dominant producer. In this way the dominant producer no longer has all the advantages and many customers are sufficiently motivated, by the low cost or a better product, to test the open source product. At this point, even the large software companies hardly manage to convince users to buy their software if another open source alternative is available.

Starting from the SOA architecture presented in the previous chapter, we carefully selected open source components to cover each technology category. In choosing the appropriate open source product we defined a selection criteria presented in table 1. This is important because even though there is an open source product that satisfies our necessities we must investigate if the product is mature, whether patches and updates are regularly released and how actively it's supported through community forums and websites. Very often we may find code from "dead" projects. Support also plays a significant role in selecting the right product.

Table 1. Selection criteria guidelines for open source products

Criteria	Observations
Distribution costs	Is the product licensed using one of the common open source licenses: GPL, BSD, LGPL, Mozilla Public License? We want to make sure there are no hidden costs, restrictions in usage or modification.
Extensibility	Can the 3rd party component be extended by allowing to add new functionality?
Viability	How well is the product supported in the open source community? Is it a "dead" project or not?

Now that we defined a general set of guidelines for choosing our open source products or components we must define the categories from which we will select them.

The centralized management for the business rules and logic of each application in our architecture is the enterprise decision management or Enterprise Decision Management. Through EDM, business rules are gathered from each application and managed in a centralized way. In Table 2 we selected some open source EDMs:

Table 2. Open source EDM

Product	Comments
JBoss Rules (Drools)	It is a well-known product, dating from 2003. It provides a unified and integrated platform for rules, workflow and event processing.
OpenRules	It has a restrictive license. In order to integrate it into commercial applications the user must purchase a non GPL license.
OpenLexicon	It is not very easy to use and integrate.

Based on the selection criteria defined in table 1, the most suited product for our architecture is JBoss Rules (Drools).

In the next phase we need a product that takes the messages from the business applications and directs or transforms them. When an application needs to communicate with another, it will send a message to the Enterprise Service Bus, ESB. Then the ESB is responsible for determining how to send this message to its destination.

In this category we choose Apache Synapse Enterprise Service Bus because of the following characteristics:

- lightweight and easy to use and configure;
- support for XML, Web Services and other protocols;
- highly mature, originating from 2005;
- very fast, needed in a production system.

We have to choose a Business Process Management or BMP product. We need a simple to use but powerful engine, which is why we selected JBossjBPM open source product.

For choosing the Web Service Mediation open source solution we also analyzed the already selected ESB Apache Synapse. The Apache Synapse features include proxy, caching, load balancing/fail-over capabilities and WS-Security support.

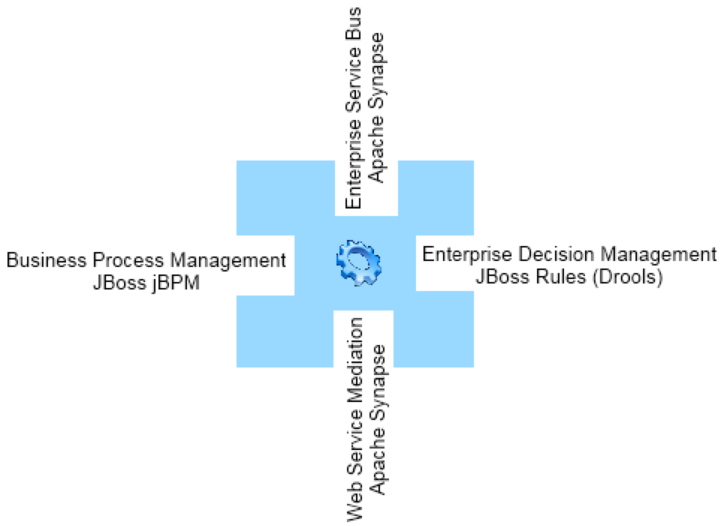


Fig. 3. Open source products integrated in the manufacturing system

Table 3. Open source products integrated in our architecture

Product Category	Selected Product	Home page of product
Business process management	JBossjBPM	www.jboss.org/jbpm
Enterprise decision management	JBoss Rules (Drools)	www.jboss.org/drools
Enterprise service bus	Apache Synapse	http://synapse.apache.org/
Web service mediation	Apache Synapse	http://synapse.apache.org/

In Figure 3 and in Table 3 there are presented the open source components that are integrated in our manufacturing-oriented informatics system.

5 Grid computing benefits in manufacturing-oriented informatics systems

When dealing with complex, uncertain and confusing problems we have to think broadly, systematically and to see the big-picture of it. In [9] it is stated that a complex problem is characterized by: complexity - contains a large number of diverse, dynamic and independent elements; difficulty in measurement - is difficult or practically unfeasible to get good quantitative data; novelty - where a new situation evolving or an innovative design is needed. Nowadays, due to the existing computing power we can model, design and architect systems and solutions for solving our biggest and complex problems. The usage of computing power is direct proportional with the complexity level. For example, for resolving a problem with a low complexity level we can use our personal pc for modeling, statistical analysis and math solving equations. For problems with a medium or high level of complexity we have to use individual super computers specialized in modeling and solving our problems. For a higher level of complexity such as earthquakes simulations, weather predictions, medical modeling, huge statistically analysis, production planning and control and other types of actions that require modeling, analyzing and processing of a large amount of data, we need the power of multiple computers and / or super computers. In this case we have to use the resources of multiple CPU, desktops CPUs, mobile CPUs, servers and super computers CPUs, and GPUs in our actions for finding the solutions and answers we are waiting for. Here comes in play the technique and physiology of the grid computing. CERN defines the Grid to be *"a service for sharing computer power and data storage capacity over the Internet."*

From [10] we find out that grid computing offers a model for solving massive computational problems by making use of the unused CPU cycles of large numbers of disparate, often desktop, computers and mobile devices treated as a virtual cluster embedded in a distributed telecommunications infrastructure. Grid computing has the design goal of solving problems too big for any single supercomputer, whilst retaining the flexibility to work on multiple smaller problems. Its secondary aims are: better exploitation of the available computing power, and catering for the intermittent demands of large computational exercises.
Grid computing involves sharing heterogeneous resources (based on different platforms, hardware/software architectures, and computer languages), located in different places belonging to different administrative domains over a network using open standards. In short, it involves virtualization of computing resources.
Comparing to grid computing, [11] distributed computing normally refers to managing or pooling the hundreds or thousands of computer systems which individually are more limited in their memory and processing power. On the other hand, grid computing has some extra characteristics. It is concerned to efficient utilization of a pool of heterogeneous systems with optimal workload management utilizing an enterprise's entire computational resources (servers, networks, storage, and information) acting together to create one or more large pools of computing resources. There is no limitation of users, departments or originations in grid computing.
The software architecture of a system [12] is the set of structures needed to reason about the system, which comprise software elements, relations among them, and properties of both.
Designed and build for addressing a wide range of problems, [13] computational grids are created to serve different communities with widely varying characteristics and requirements.
Two important drivers for grid architectural solutions are the scale and performance requirements. Computational infrastructure, like other infrastructures, is fractal, or self-similar at different scales. We have networks between countries, organizations, clusters, and computers; between components of a computer; and even within a single component. However, at different scales, we often operate in different physical, economic, and political regimes.
According to Wikipedia [15], the Grids can be categorized with a three stage model of departmental grids, enterprise grids and global grids. In production purposes the department and enterprise grids are more widely used. The global grids are primarily used in science experiments and other simulations as mentioned before. As described in [14] there are several large national and international Grids, for example, Enabling Grids for EsciencE (EGEE), GridPP, Nordic DataGrid Facility, Open Science Grid, and TeraGrid. These grids are capable in providing around five hundred Teraflops of processing power and around ten PetaBytes. They also support high rates of data transfers within production or research networks.
In [14] the author identifies the challenges to implement a production Grid as being: to provide high reliability, high throughput, scalable, multi-user, distributed data centers which operate around the clock and around the world; to provide the security, technologies, infrastructure and architecture to serve an increasingly large and demanding community; and to provide usable services that facilitate the entry of new participants in the use of distributed computational infrastructures.
Trying to respond to all the challenges mentioned above, we present an overview of our research in improving the production planning and processes by designing a new production Grid architecture.
The architecture addresses and not only to those production processes in creating and delivering user custom products, such as: shoes, dresses, gloves, etc. The products are

customized by the user himself based on certain material categories, colors and other features supported and provided by the production system.

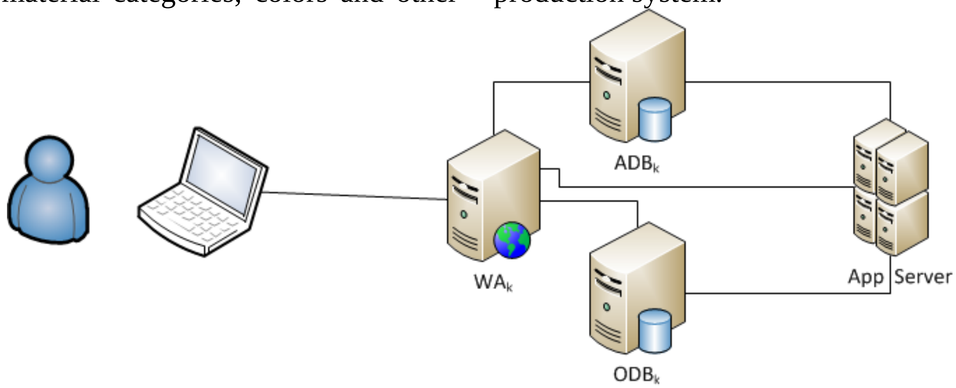


Fig. 4. User Interaction with the production portal / web application

The product manufacturing starts from the moment when the user accesses the manufacturer web site or portal or goes to its office. When accessing the portal the user has the possibility in choosing from a wide range of products the initial prototype based upon he can customize his final version. If he goes to the office he has the option in creating a 3D model of its body (parts). From architectural point of view, the user interacts in a direct mode with the web application server (WA) and indirect with the whole production Grid through the Application Server (AS) as shown in Figure 4.

As a normal web application the manufacturer portal has an application database for storing the user related information (profile, order history, billing history, etc). For retrieving the products information the web application interacts through specialized web services with the AS.

After finishing designing its own product, color, features, season, brand, etc - the user places his order in the system. Here he has to possibilities to either to wait for a real time or asynchronous answer regarding the product availability, time of execution, time of delivery and price. Either way the user is notified by email, text message or phone call about its order. When placing his order, in background, the application stores the order in the Order Database (ODB) from where they are picked by the AS.

The system presented in Figure 4 is the scenario for a certain number of users. For managing huge traffic and orders, the system can be designed in distributed clusters around the world.

After an order is placed in the system, it is taken by an Order Processing (OP) process which starts processing it.

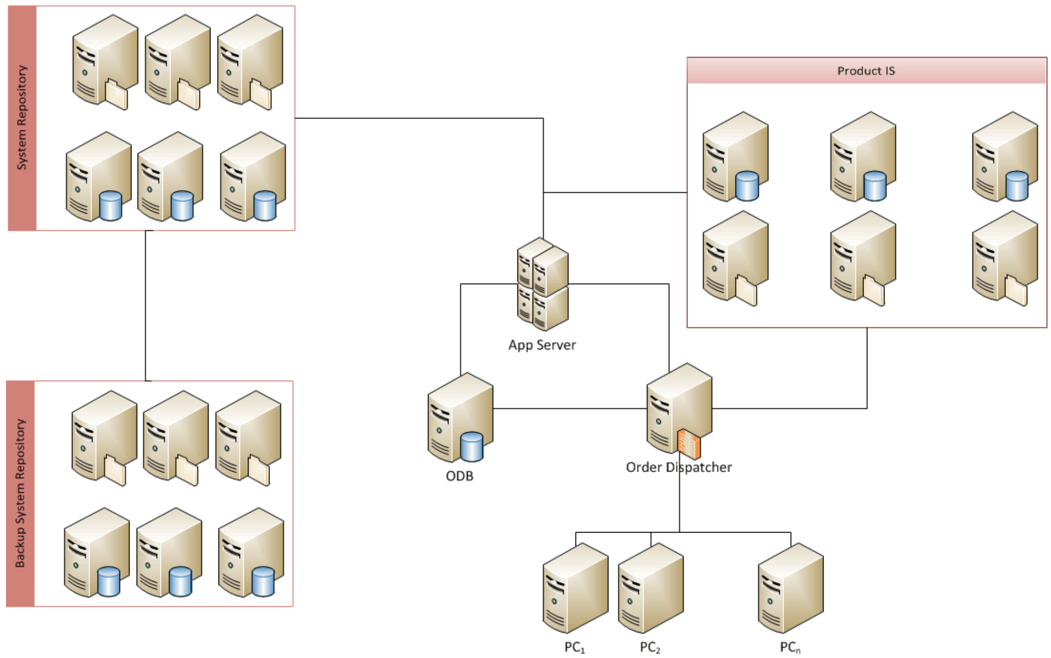


Fig. 5. Production Grid overview

The OP pass the order to an Order Dispatcher (OD) which based on products certain features dispatches the order to a Product Controller (PC).

From architectural point of view, the Grid is designed as shown in figure 5, where the AS is the central point which interacts with system repositories – databases, files, knowledge repositories – and its backups, with product information repositories – which can be widely distributed across the world and through the dispatcher with products controllers.

A product controller is an application, which controllers the actual product manufacturing. It is responsible in indentifying the product components and raw materials and based on its own information systems in contacting the final product component or raw material provider, as shown in figure 6. Architectural specking, the design proposed in figure 6 is a sub-grid of the entire production grid and is composed of: the Product Controller Information System (PC IS) which contains the product manufacturing related information – such as location where components are build or from where raw material can be brought; the Product Controller which as mentioned before is a software application, which we can roughly assumed that is an minimal ERP with certain custom functions in planning, negotiation, discovery and so on; a sub-grid of manufacturer distributed systems (PC Manufacturer L1) having as end-points custom ERP's implementations. When an order reaches a Product Controller another internal process is started. The process starts by identifying the product components and raw materials. Based on requirements, the controller searches within its internal IS the cheapest, near manufacturer taking in account the previous historical data. After the identification and discovery is made, the controller initiate a communication and trading channel for querying the availability and the cost of the component / raw material. If the component / raw material are not available in stocks, then it asks for time estimation and costs in case it waits for it. After querying several PC Manufacturers based on the availability, costs, distance it takes a decision either to wait for those missing components or search for replacements. When all the components / raw materials are found, the controller plans and forecasts the execution time and costs and sends them back to the user. If the user gives his consent and pays the bill the controller place all the orders for all the product components and raw materials and redirects them to the Assembly Line where the final product is made and from where is shipped to the user. The Product Controller internal process is finished when the final product is build and shipped to the user. Until then he continuously queries the manufactures or the assembly line in order to get real time status regarding the product manufacturing.

The AS controls and manages the whole process. Also it gathers all the data from users, dispatchers, controllers, manufactures ERP's and assembly lines for historical and statistical purposes in data warehouses from where knowledge for improving the systems / order process can be extracted. Knowing the fact that users are widely spread all across the world and they can order a product from anywhere, the proposed system can easily respond to any order due the fact that the system itself is distributed in different geographical regions. We choose this production grid architecture in order to cut costs, to optimize the entire planning and manufacturing process and build an easy and transparent way on how a user can achieve custom products and low prices.

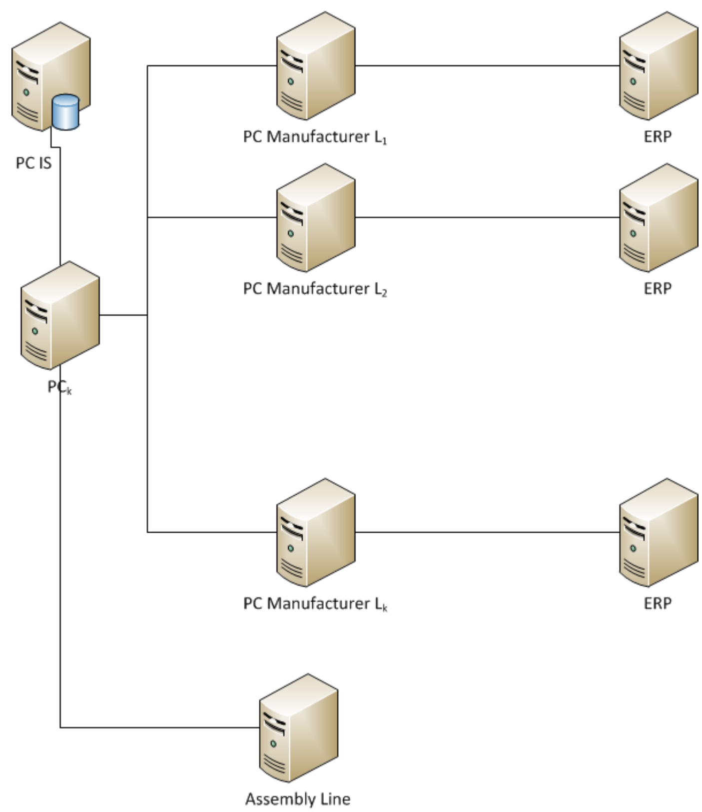


Fig. 6. Product Controller

From manufacturing point of view, the system takes all the advantages and functions exposed by the grid system itself, such as processing power, high transfer rates, storage space, security, failover, etc.

6 Web based project management for manufacturing informatics systems

In the implementation of production oriented systems, project management is of paramount importance. Most implementations of ERPs fail not because of the system itself or because of the corporate culture of the company where it is implemented but because of a poor implementation process. It is this problem that needs to be addressed using new technologies in project management.

Project management includes the organizational structures necessary to carry out projects within the organization, internal organization of the project and correlations between them. It is at the same time an integrated concept of management. Project management involves the conscious application of a coherent set of principles, rules, knowledge, methods, techniques and tools used in planning, organizing, directing and controlling a project. An integrated concept for project management takes into account four basic elements of a project: time, budget, quality, expectations of participants.

A project requires using specific project management concepts (network diagrams, WBS, time, resources, quality, etc..), specific tools and techniques for each phase, but it also requires the participation of individuals and organizations that form the organization structure of the project. Knowing all the people involved is very important because they can positively or negatively influence the project. The number and composition of staff involved in a project will depend on the following factors:

- project size;
- the timeframe of the project (if the project needs to be completed in a short time, the number of people involved will be higher);
- importance of the project (as the project has a greater importance, the number will increase and more project participants will be recruited);
- needs of the project, which implies certain specialist staff.

Based on the number of people involved with a certain project the management needs increase and tools must be used in order to address this need, especially since in recent times project teams are no longer bound by geographical limitations so managing remote people is a new problem that project managers need to deal with.

Project-based activities are central to the concept of virtual organization, which involves working within a limited period of time using distributed teams. Large, complex projects are most often the result of a collective effort, each participant adding value to the project. Even within a single organization distributed virtual projects are now quite common. Increasing prevalence of projects implemented in a distributed way has led to new challenges relating to coordination and control functions that are necessary to ensure proper integration and delivery of expected results. Some recent studies have addressed these issues, including research on how information technologies can help project management.

Project management software involves different processes that help in inter-process communication in distributed and remote teams. Service-oriented architecture (SOA) has the potential to greatly improve efficiency of project management. Project management software increases visibility and control in the application development lifecycle, providing better control over the entire development process, from the planning stage to the development and delivery stages.

The purpose of project management software is to produce a product which is delivered on time, within budget, and capabilities expected by the customer. Project management software will ensure a properly managed project, which has a set of clear goals and objectives, known by all participants and whose success and progress is quantified and controlled. Resources are used effectively and efficiently to produce the desired product.

In such a distributed environment, the need for translating classical concepts of project management to a web-based architecture has become evident in order to facilitate collaboration and management of distributed resources to deliver a consistent product on time and within the budget limits set for the project.

If a company wants to implement a production oriented computer system, first of all it needs the tools to properly plan, select and implement the solution. Considering the advantages described above for using a web-based project management system it should be the company`s first choice in starting the implementation process. Such system would be used to plan the project, divide the project into work blocks, launch a request for proposal for each block, select the provider and then monitor, control and interact with the provider 24/7 no matter their geographical position.

This architecture will provide various services for project management which will be used to collect the information from the different levels of software development and the different people involved in the project and product implementation.

This software will handle all the activities starting from the RFP from the client to planning, completion and delivery of the product by the provider to the client. All the activities will be monitored and will be documented providing information to all parties involved so that the scope of the project is clear as well as the feature sets for each module to be implemented, thus making sure everyone is working in synch and there are no miscommunication issues between client and provider and also between the members of the team. This system also serves as an archive for all the information and events related to the project which makes it the ideal tool for analyzing the project and optimizing work flow processes.

The client will come to know the exact manpower involved in the project at any time and the time spent on each task as well as constantly comparing the actual results with the plan and reallocating resources as needed. This will lead to total transparency within the project between the developer and the client with information being available to both parties within a secure system accessible from anywhere in the world via the internet.

Based on the advantages presented above, the need for a web based project management system is clear when implementing a system in a distributed environment with multiple teams in different geographical locations all working towards a common goal. Such a system would eliminate or at least mitigate risks regarding communication, coordination, poorly understood requirements and poor time management. A company implementing a new ERP system should take advantage of the new global market and not be limited by location when choosing providers solely because of the logistics involved in working with virtual teams since it is clear that new technologies such as service oriented architectures, grid processing and web services are beginning to bridge the gaps between supply and demand in different corners of the world.

7 Conclusions

The integration of ERP systems with manufacturing systems and end-users portals raises new series of limitations because of the differences between each specific system, choosing the sharable information, data format differences, organizational culture related to sharing information with confidential character such as technical documentations or products recipes. These kind of problems can be solved using Service Oriented Architecture and integration between commercial and open source software components. Problems rise regarding the security of information access and regarding the hardware performances and software response but grid architecture prove to be cost saving and able to optimize the entire planning and manufacturing process. A proper project management system will help to deliver the solution on time, within budget, and capabilities expected by the customer.

Acknowledgement

This work was co-financed from the European Social Fund through Sectoral Operational Programme Human Resources Development 2007-2013, project number POSDRU/107/1.5/S/77213 „Ph.D. for a career in interdisciplinary economic research at the European standards”

References

[1] A. Dashchenko, *Reconfigurable Manufacturing Systems and Transformable Factories*. New York, SUA: Springer Publishing House, 2010, pp. 1-759.

[2] M. Groover, *Automation Production Systems and Computer-Integrated Manufacturing*. New Jersey, SUA: Prentice Hall Publishing House, 2008, pp. 1-1042.

[3] I. Lungu, A.R. Bologa, V. Diaconita, A. Bara and I. Botha, *Integration of Informatics Systems*. Buchares, Romania: ASE Publishing House, 2007, pp. 1-297.

[4] L. Hurbean and D. Fotache, *Integrated software solutions for business management-ERP*. Bucharest, Romania: Economics Publishing House, 2007, pp. 1-264.

[5] A. Uta, I. Intorsureanu and R. Mihalca, "Criteria for the selection of ERP software", *Economic Informatics Journal*, vol. XI, no. 2, pp. 63-66, 2007.

[6] J. Davis, *Open Source SOA.* SUA: Manning Publishing House, 2009, pp. 28-57.

[7] R. Kanneganti and P. Chodavarapu, *SOA Security*. SUA: Manning Publishing House, December 2007, pp. 1-512.

[8] M. Georgescu and D. Milodin, "Open Source Tools to Assist in the Development of Software Application", *Open Source Science Journal*, vol. 2, no. 2, 2010.

[9] *The Art Of Complex Problem Solving*, http://www.idiagram.com/CP/cpprocess.html, accessed on 28.05.2011.

[10] *Grid Computing Definition,* http://www.wordiq.com/definition/Grid_computing, accessed on 28.05.2011.

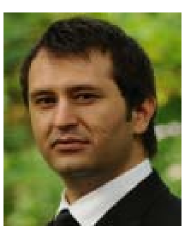
[11] *Difference Between Grid Computing Vs. Distributed Computing,* http://www.jatit.org/distributed-computing/grid-vs-distributed.htm, accessed on 28.05.2011.

[12] Software architecture, http://en.wikipedia.org/wiki/Software_architecture, accessed on 28.05.2011.

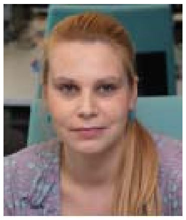
[13] I. Foster and C. Kesselman, *The Grid 2, Second Edition: Blueprint for a New Computing Infrastructure*. SUA: 2003, pp. 1-748.

[14] R. Pordes, "Challenges Facing Production Grids", *Fermi National Accelerator Laboratory,* Batavia, Illinois, US, http://home.fnal.gov/~ruth/GridsInAction.pdf.

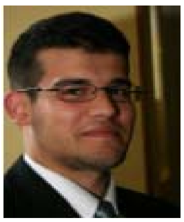
[15] Wikipedia http://en.wikipedia.org/wiki/Grid_computing#Production_grids.



Constantin-Daniel AVRAM has graduated the Faculty of Economic Cybernetics, Statistics and Informatics in 2005. He holds a master degree in Computerized Project Management. He is the author and co-author of several scientific articles in the field of ERP systems, manufacturing-oriented informatics systems, software recognition, risk management, software and projects quality assurance, project management. He is currently attending a PhD Program in Economic Informatics. He has more than six years of experience in ERP implementations. He was involved in large and very large international ERP projects.



Cecilia CIOLOCA has graduated the Faculty of Economic Cybernetics, Statistics and Informatics from the Bucharest Academy of Economic Studies in 2008. She is currently conducting doctoral research in Economic Informatics at the Academy of Economic Studies. She is experienced in software development using open source technologies. She has successfully completed several projects using various programming languages and open source frameworks. She is proficient in using Java Enterprise Edition.



Vasile-Laurentiu CIOVICA has graduated the Faculty of Science, in 2008 gaining a Bachelor of Science degree in Information Technology with a thesis on Translators and Interpreters for Code Generation and Software Optimization. In 2010 he gained a Master of Management degree in the field of Cybernetics, Statistics and Economic Informatics with a thesis on Intelligent Agents. He is currently a PhD student at Academy of Economic Studies in Bucharest. He is the author and co-author of more than 12 scientific articles in the field of software quality and optimization, code generation techniques, collaborative systems, data bases, programming environments and techniques, mobile platforms and economic informatics systems. He is also an active software developer, being the author of few applications. His area of interests includes: software quality, optimization techniques and algorithms, code generation techniques, economic informatics systems, intelligent and collaborative systems, mobile platforms.

Mihai GEORGESCU has graduated the Faculty of Electronics, Telecommunications and Information Technology – Polytechnic University Bucharest in 2005 and also completed a master program at the Faculty of Economic Cybernetics, Statistics and Informatics from the Bucharest Academy of Economic Studies in 2007. He has experience in the development of software products using various programming languages like C/C++, C# both Windows Forms and Web.

Mihai CURTEANU has graduated the Faculty of Cybernetics, Statistics and Economic Informatics in 2008. He obtained a Master's Degree in Project Management in 2010 with a paper on Online Project Management Applications. He is a PhD student of the Doctoral School of Bucharest Academy of Economic Studies in the field of Economic Informatics.