OFFER REQUEST MANAGEMENT IN SERVICE ORIENTED HOLONIC MANUFACTURING*

Cristina MORARIU[1], Octavian MORARIU[2], Theodor BORANGIU[3]

This paper explores the offer request management domain in manufacturing context considering two main directions: costs and time. The goal is to enable manufacturing enterprises to respond in real time to customer requests with offers that are computed in an accurate manner. Traditionally the offer was based on empirical information and often leads to financial losses. The main reason for this is the lack of real time information about the current production line available capacity. Another important aspect when creating the customer offer is the ability to decide on what operations can be externalized to third party suppliers and what operations can be done in-house, in regards to time and costs. This paper argues that the advances in standardization in the manufacturing area, with SOA orientation being the most notable one, help in centralizing the real time production information and allow systems to compute in real time the optimum offer for each customer order. The offer creation problem has two sub-problems: first the producer must decide on which service to externalize and the second one is to compute the overall batch production time and costs. To solve this problem, we propose a genetic algorithm implementation that considers both hard conditions like resource and material availability and soft conditions, like profit, costs and time in which the batch order can be completed. The algorithm is targeted to solve the offer problem in the context of holonic manufacturing systems architectures. The original contribution in this paper consists in the novel architecture proposed for the Offer Request Management module together with the genetic algorithm based implementation. Experimental results are based on a sequence of customer orders for T shaped and H shaped product assemblies and show the differences between the approach based on empirical rules and the genetic algorithm approach proposed in this paper.

Key Words: Service Oriented Enterprise Architecture, COM, mixed production planning and scheduling, Web services, EIS, multi-agent system, HMES

---

[1] PhD., Faculty of Automatic Control and Computers, University POLITEHNICA of Bucharest, e-mail: cristina.morariu@cimr.pub.ro
[2] PhD, Faculty of Automatic Control and Computers, University POLITEHNICA of Bucharest, e-mail: octavian.morariu@cimr.pub.ro
[3] Prof. , Faculty of Automatic Control and Computers, University POLITEHNICA of Bucharest, e-mail: theodor.borangiu@cimr.pub.ro
* *The results presented in this article were obtained withthe support of Ministry of Labour, Family and Social Protection through Sectorial Operational Programme Human Resources Development 2007-2013, Contract no. POSDRU/107/1.5/S/76909.*

1. Introduction

Manufacturing systems have evolved from simple production lines that were able to produce a limited set of products, to complex environments that can produce an almost infinite number of different products by combining various operations and procedures. This evolution is driven by the changes in the market environment, where customers demand new products with more specific requirements, to meet their business goals. Also, there is an increased need for enabling the production lines to react to unexpected situations, like resource break-down and rush orders with minimum impact. This aspect reduces the overall production costs and offers more flexibility for companies to respond to market changes. These intelligent production lines are de-signed based on the holonic concept [1], where a holon is defined as an entity that is simultaneously a whole and a part. Starting from this definition, manufacturing systems are represented as hierarchies of holons - each one related to components of the material flow (products), the production flow (orders) and the processing flow (resources), and encapsulating specific functionality of the component at each level of the enterprise processes in which the component is involved.

The PROSA reference architecture [2] for holonic manufacturing is adopted, extended with the auto-supply functionality, the basic holons being defined as order / supply holon, resource holon and product holon. The order holons are generated in centralized mode by mixed batch planning (sequencing products) and product scheduling (sequencing operations and allocating resources) which is performed by an expertise (or staff) holon. Each holon contains a physical part (the computing or manufacturing component it represents) and an informational part (the agent describing the component's behavior). Current research is focused on service orientation of holonic manufacturing systems and is based on new concepts such as Service Oriented Architecture (SOA) [3], Enterprise Service Bus (ESB), Web Services, Manufacturing Service Bus (MSB), Distributed Intelligence (DI) and product-driven automation, Service-oriented Multi-Agent Systems (SoMAS), resource service access model (RSAM) [4, 5, 6, 7].

At the aggregate level of a manufacturing enterprise, SOA is the standard for business process modeling and management [8].The integration of the shop floor processes in the enterprise business processes requires service orientation to fill in existing technological gaps and solve legacy problems. An end to end integration, from the initial offer request to the manufacturing execution system and supply chain, provides enterprises with the ability to gain agile control of all activities, allowing flexibility and constant improvement. Other advantages of service orientation for manufacturing systems include central and integrated audit track allowing enterprises to comply with specific laws that apply to the products [9, 10].

The primary goal of Service Oriented Architecture is to align the business world with the one of information technology - the latter being partitioned on two layers: (1) the business layer [management of customer orders]; (2) the shop floor layer [execution of customer orders]. SOA is the bridge that creates a symbiotic and synergistic relationship between the two layers. SOA is an IT system model providing flexibility to the enterprise in the way business applications are created. SOA is not just focused on application integration, but also on application construction from existing IT assets. This type of architecture allows for the creation of composite business applications from independent, self-describing, and inter-changeable code modules called services. These services are available for use on a services bus, and they can be arranged together into a business process or composite application using process choreography. Thus, the major components of SOA are: services; services bus; process choreography - composite applications; message transformation, mediation and routing; services registry [11].

The Service Oriented Enterprise Architecture (SOEA) for the generic business layer of a manufacturing enterprise is shown in Fig. 1. The set of business processes and business services that a given business user will consume (consumers of processes and services) are: Offer Request-, Production Order-, Rush Order-, Customer and Supplier Relationship-Management.

SOA framework for production management and control integrates four areas, see Fig. 2: (1) Offer Request Management; (2) Management of Client Orders; (3) Order- & Supply- Holon (OH, SH) Management; (4) OH Execution & Tracking. In the first area the Expertize Holon (EH) is responsible for generating offers in response to requests, based on: product knowledge (embedded in Product Holons - PH), resource capabilities (from Resource Holons), supply constraints (by using web services in Extended Information Systems) and activities planning (CAPP).

Once received from the customer, offer requests are interpreted, processed and mapped into aggregate production orders (APO) at the ERP level. APO is the input to the Global Production Scheduler (GPS) which generates the lists of Supply- (SH) and Order Holons (OH).

The Customer Offer Request management processes are treated as compositions of subordinate business sub-processes and services. They are detailed in service components - converted into a detailed set of definition metadata that will describe the service to the information system. The Service Oriented Architecture concept is used to face the interoperability problems in the autonomous, reconfigurable architecture implemented as a total integrated Holonic Manufacturing Management and Control System (TIHMS).

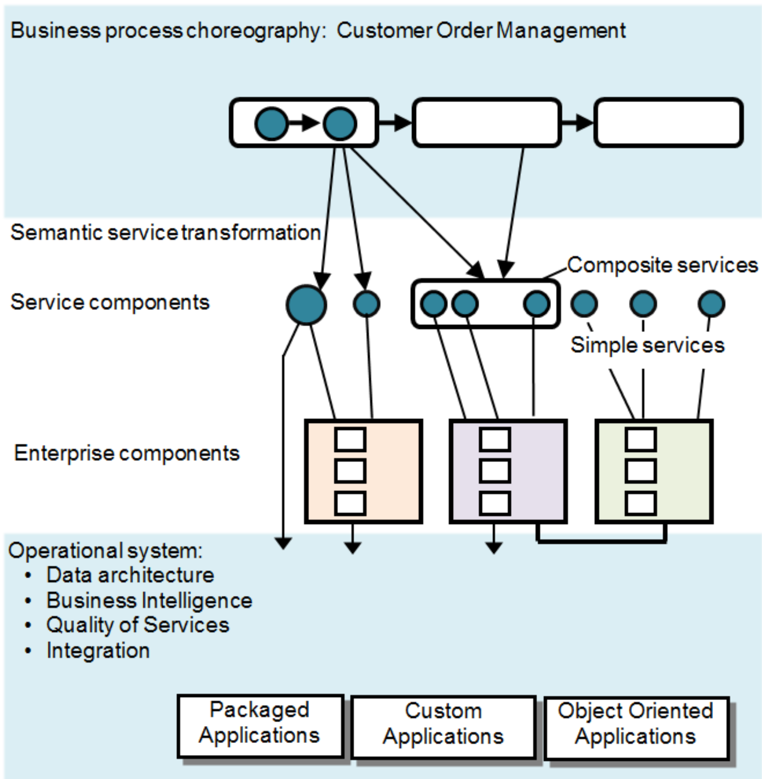

Fig.1. Service Oriented Enterprise Architecture for the business layer of manufacturing enterprises

TIHMS integrates the Customer Order Management System (COM) and the distributed Manufacturing Execution System (dMES).

The following section of this paper presents the main standards that are emerging in manufacturing domain. Section 3 discusses the aspects related to externalization of several activities and the costs associated together with the internal production costs. This analysis serves as input for the genetic algorithm execution that computes the offer for the customer. Section 4 presents the main factors that are part of the custom offer creation and proposes a genetic algorithm that would assure the computation of the best possible offer in the context of the availability, profit, costs and time. The original contribution in this paper consists in the novel architecture proposed for the Offer Request Management module, together with the genetic algorithm based implementation. Section 5 presents the experimental results obtained on a series of order for T and H shaped product assemblies, when considering the traditional rule based approach against the genetic algorithm proposed in this paper.

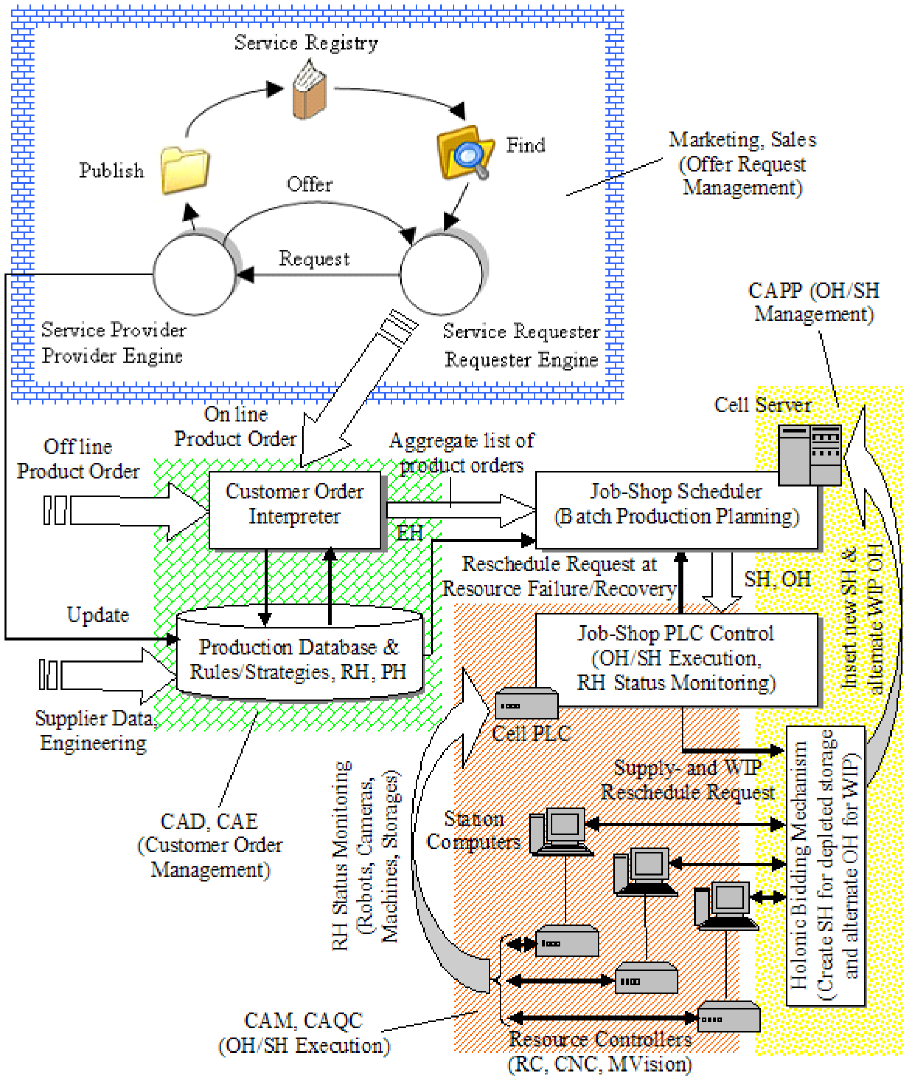

Fig.2. SOA integrates COM with dMES in a total integrated holonic approach

Finally, the conclusions drawn from this research are presented, outlining some best practices in regards to the genetic algorithm approach, specifically the fitness function tuning and possible solution crossover configurations.

2. Emerging Standards in SOA Oriented Manufacturing

Almost all industries by now strive to achieve SOA architecture, either by starting from scratch or by slowly migrating the legacy applications and more importantly legacy processes towards this goal. So, how are the manufacturing enterprises reacting to this trend? First of all the manufacturing enterprises have to move in this direction also. Probably the most important reason is that the market has extended and be-came global. This means that a new kind of integration is required with third party partners that are not even known in advance or that are subject to change. The most important standards from a manufacturing perspective are presented in the following section, together with their scope, starting from shop floor level to the integration with external partners, further divided in document format standards and protocol standards.

2.1 Document Format Standards

OAG-BOD: is a document format standard developed by Open Applications Group Integration Specification (OAGIS) with the goal of encapsulating the data from and to higher level applications on ERP layer and external partners. For example, customer orders are transformed in a standard document (OAG Business Object Document - BOD) and passed to the MES. This allows a decoupling between MES and higher levels, allowing flexibility in regards to the ERP implementation (i.e. ERP system can be replaced with no impact on MES integration). For more information refer to: www.oagi.org.

ISA-95: was developed in order to provide the interface used to exchange data be-tween enterprise systems (ERP, Supply Chain, High Level Scheduling) and manufacturing operations systems (Work Item Scheduling, Plant Data Collection and Analysis, WIP Tracking, Operation Execution). For more information refer to: www.isa-95.com.

ISA-88: was developed by the WBF and is the worldwide recognized standard for batch processing. The standard is implemented by BatchML, which provides a set of XML type and element definitions that can be used for batch, product recipe and equipment data. The latest version of BatchML at this time is V2. For more information refer to: www.isa-88.com.

SCOR: Supply Chain Operations Reference-model has been developed by Supply-Chain Council (SCC) as the cross-industry standard for supply-chain management. The SCOR model scope spreads over all business processes involved in satisfying a customer's request. The five major processes covered are: Plan, Source, Make, Deliver, and Return. For more information refer to: www.supply-chain.org.

OSA-EAI/CBM (MIMOSA): Open System Architecture for Enterprise Application Integration (OSA-EAI) architecture is a specification published by the Machinery Information Management Open Systems Alliance (MIMOSA) organization and is focused on asset management. MIMOSA publishes XML specifications for Enterprise Application Integration (EAI) and Condition-based Maintenance (CBM). For more information refer to: www.mimosa.org.

STEP: Standard for the Exchange of Product Model Data is an ISO standard (ISO 10303) describing exchange of digital product information between computer-aided design (CAD), computer-aided engineering (CAE), and computer-aided manufacturing (CAM) systems. For more information refer to: www.steptools.com.

ebXML: Electronic Business using eXtensible Markup Language set of specifications, that assure a standard method to exchange business messages, conduct trading relationships, communicate data in common terms and define and register business processes. ebXML was started in 1999 as an initiative of OASIS and the United Na-tions/ECE agency CEFACT. For more information refer to: www.ebxml.org.

2.2 Messaging Protocol Standards

FDT: Field Device Tool standard is needed to ensure the consistent management of a plant-wide control and automation technology, focused on life-cycle management. A goal of the standard is to create universal and central plant-wide tooling for the life-cycle management of heterogeneous fieldbus environments, multi-vendor devices and sub-systems in process and manufacturing. For more information refer to: www.fdtgroup.org.

EDDL: Electronic Device Description Language technology is used by major manufacturers to describe the information that is accessible in digital devices. Electronic device descriptions are available for millions of devices that are currently installed in the process industry. The technology is used by the major process control systems and maintenance tool suppliers to support device diagnostics and calibration. For more information refer to: www.eddl.org.

SOAP: Simple Object Access Protocol is a protocol specification for exchanging structured information in the implementation of Web Services in computer networks. It relies on Extensible Markup Language (XML) for its message format, and usually relies on other Application Layer protocols, most notably Hypertext Transfer Protocol (HTTP) and Simple Mail Transfer Protocol (SMTP), for message negotiation and transmission. SOAP can form the foundation layer of a web services protocol stack, providing a basic messaging framework upon which web services can be built. For more information refer to: www.w3.org/TR/soap

3. External Costs and Internal Costs

When considering customer offer creation the first problem to solve is to decide what operations and raw materials should be outsourced, or in other words acquired from external suppliers and what can be manufactured in-house. The decision is taken based on two factors: internal availability and costs. Externalization introduces a new problem for manufacturing enterprises in regards to the integration mechanisms of the software infrastructures. Fig. 3 below illustrates this concept.

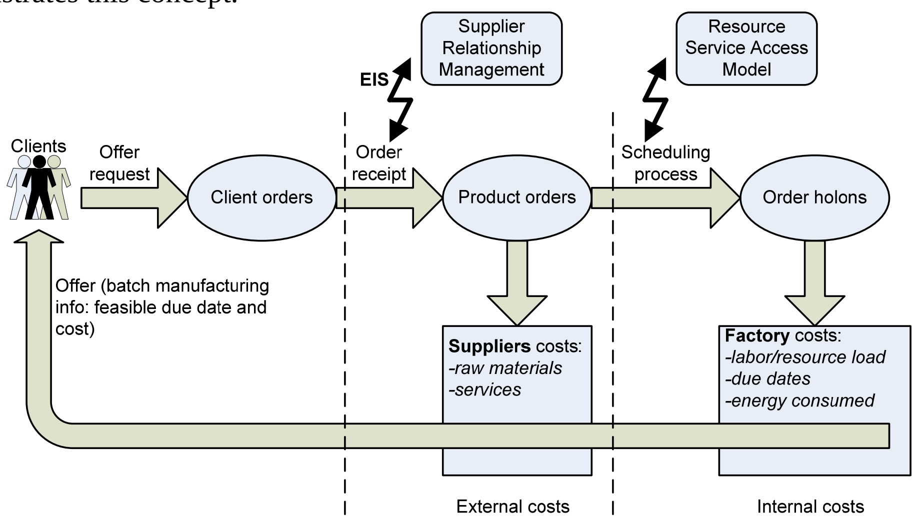

Fig.3. Computing the total cost for manufacturing a batch of products

The next section proposes a mechanism for achieving such an integration based on SOA paradigms and supported by SOA application stack.

3.1 Integration with external suppliers

Integration with external suppliers at offer management layer is done according to four main coordinates: service, raw materials, time and costs. The first two coordinates are focused on the nature of the interaction required, which can be a service, raw materials acquisition or a combination of the two. The last two refer to the practical aspects of the customer offer, specifically on the costs associated and the time in which the supplier is able to provide required deliverables. From the manufacturing enterprise perspective, these factors are accountable for the external costs associated with the customer offer.

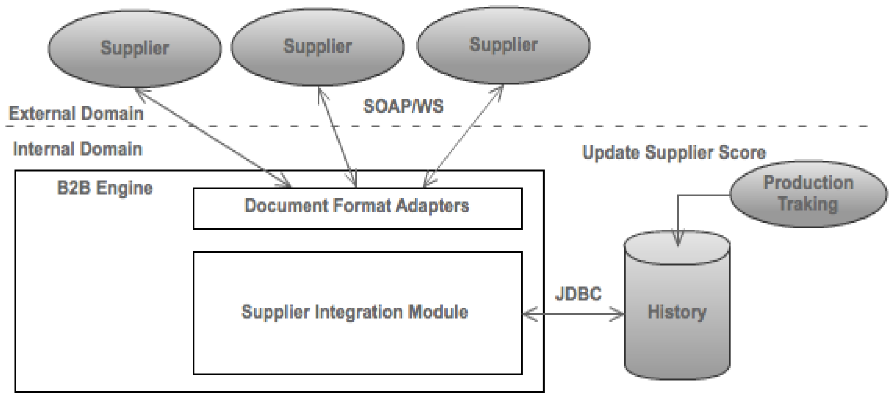

Fig. 4. B2B based architecture of the Supplier Integration Module (SIM)

The Supplier Integration Module (SIM) architecture is based on a Business to Business (B2B) engine, part of the SOA product stack. B2B engines offer support for federated integration based on bidirectional web service information flow, consisting in authentication and authorization services, encryption and data privacy, document format conversion and data encapsulation. The B2B integration mechanism has been adopted by many enterprises, not only form manufacturing domain, when integrating with third parties and OEM's over the last decade.

One of the most important problems addressed by the SIM is evaluating the supplier selection criteria. In other words the SIM module must evaluate all competing suppliers in terms of time and costs for the required service or raw materials and decide on the best approach for minimizing the external costs while keeping the potential risks associated at a minimum. Traditionally this was implemented as a simple auction type algorithm with several sequential steps and based in conjunction with set of predefined thresholds the best offer was selected. This works fine in a scenario where the suppliers are equally trustful and the number of suppliers is relatively small. However, in the current global economy this is not the case anymore and so an advanced mechanism must be used, that considers the associated risk and tracks the trustfulness of suppliers, so it can be used as a factor in the decision making process. For this reason the SIM module has a decision history repository where each supplier is stored along with a risk factor. The risk factor is the percentage of failed transactions against the total number of transactions with that supplier. The risk factor is updated in the repository by the production tracking module after the product batch has been completed. If the services or raw materials from the external suppliers were adequate and available on time then the transaction is considered successful. The SIM module evaluates the external suppliers in descending order based on the risk factor. The formula to compute the decision factor is:

$$D = \frac{1}{Time} \times T + \frac{1}{Costs} \times C + R_{factor}$$

where T is a constant representing the time factor impact and C is a constant representing the cost factor impact. The decision factor D, gives a rating for the each specific supplier in the context of the current request (time and costs factors). The first N suppliers from the repository ordered by risk factor are considered sequentially and D is computed for each. The first one that is achieving a result between the acceptable thresholds is selected. Technically the interaction is implemented using web services and a publish-subscribe mechanism for services and raw materials offered. The main services are:

- **inquireRawMaterialsAvailability** - takes as parameters the material type definition in the form of Product Holons (PH) and the quantity required. The supplier responds with the availability to provide the required materials, the time in which the materials can be delivered and the total costs.
- **inquireServiceAvailability** - takes as parameters the service definition. The service provider in this case responds with the time and costs associated with the delivery of the service.

The SIM module has an incoming web service endpoint which is consumed by the offer management module to submit the client order. Internally the SIM module de-composes the client order in materials and services required and for each one that requires external suppliers, it will search for all available suppliers that have published and filter them against the internal repository. It will then invoke one of the above web services for each selected supplier. Once the selection process is complete and the cost for each item is known, the external costs are computed as a sum of all the externally sourced materials and services. Another important aspect of the SIM module selection process are the two factors considered in the computation of the decision: T and C. These are dynamically computed based on the client offer time and costs constraints. The usage of an internal risk factor based on previous transaction history reduces the risk in the supplier selection process and allows the system to improve the decision accuracy over time.

From a technological perspective, B2B platforms represent the most appropriate middleware technology for implementing the SIM module, as it eliminates the complexity required for automated application to application communication in federated environments, while aligning with SOA design paradigms and emerging manufacturing standards for data and material flow.

Software providers offer mature solutions for B2B such as IBM B2B Sterling Integrator [12] solutions or Microsoft BizTalk Server [13].

3.2 Internal costs and availability evaluation

Internal costs are estimated by analyzing the information resulting from product batch scheduling process. Another output of this estimation is the possible completion time for the requested batch of products. All the above information concerning the internal costs results from the lifecycle of a client order [14] as follows:

-Resource usage as a result of the batch scheduling [15];

-Product batch make-span which, added to the time the manufacturing cell becomes available, results in a feasible due date. Moreover, if previous orders have flexible due dates (the due date is longer than the make-span), the current batch can be treated like a rush order and introduced in between the previous orders using the EDF technique [16]. Similarly to the SIM module presented for external suppliers in the previous section, the offer request management module is invoking several web services provided by the production scheduling and tracking module to gather the real time data, as follows:

- inquireSchedulingETA - takes as parameters the product batch in the form of Order Holons (OH) and computes the possible ETA and the costs. This computation is done by the production scheduler considering the other orders that are currently in production and the possibility of handling this as a rush order.
- inquireRawMaterialsAvailability - takes as parameters the material type definition in the form of Product Holons (PH) and the quantity required. The production scheduler responds with the availability of the required materials on the production line.

3.3 Offer Request Decomposition Algorithm

Once an offer request is received by the system, the first step is to break it down in terms of raw materials required, product batch size, services and operations required to complete the batch. The data structure in which the offer request is compiled represents the Order Holon. The steps involved are detailed below:

receiveOfferRequest
    for each product in the batch
        fetch Product Holon from knowledge base
       identify required materials from PH
       identify required operations from PH
       identify required services from PH

```
    populate Order Holon
  end for
  for each Order Holon
    fetch materials, operations, services
      can be outsourced ? flag OH item accordingly
  end for
  computeBestOfferResponse(OH)
  return offer response
end receiveOfferRequest
```

The above pseudo code section explains the basic sequence of events that are required to handle a customer order and compute an offer request and has two parts: first a transformation part that structures the data from the customer offer request in an Order Holon and augments it with the possibility to use external suppliers for each operation, service or raw material acquisition. The second part is represented by the **computeBestOfferResponse(OH)** invocation. This method encapsulates the genetic algorithm used to generate the best possible offer based on the data structured in the OH. The offer computation can be reduced to adding up the external and internal costs. However, computing the best combination of outsourcing and internal manufacturing can be reduced to a resource and material scheduling problem, with conflicting conditions. For job-shop scheduling, similar approaches were considered in literature [15, 17], where the resource allocation for a flexible job-shop is considered an NP-hard problem. For this reason heuristic solutions are preferred to analytical ones. Several generic tools are available that can be configured to resolve such a specific problem. One such example is IBM ILOG Optimization Suite [18], a product composed of a set of tools for developing custom optimization applications. From manufacturing perspective the Constraint Programming (CP) engine of ILOG can be used for detailed production planning and resource scheduling problems.

However, for offer request computation, a meta-heuristic approach is better suited considering two important factors: the dynamic nature of the offer requests (including the actual products required, the state of the production line, the delivery time re-quested), and the variability introduced in the system by external suppliers, which are not known in advance considering the SOA based B2B integration. To resolve this problem we introduce a genetic algorithm prototype that computes the near-optimum offer response considering the above described conflicting conditions.

4. Genetic Algorithm Prototype for Customer Offer Computation

The customer offer computation is a NP-Complete problem, so the solutions generated would be sub-optimal. There are several hard conditions that need to be respected in order for a customer offer to be accepted as a viable solution such as raw materials availability from third party suppliers or from internal stocks, resource availability for all required operations and services and scheduling possibility within the required due date. As these conditions are sometimes conflicting a genetic algorithm would be able to generate only sub-optimal solutions for the customer offer problem. Genetic algorithms start with a randomly generated population of solutions and by applying operations as selection, crossover and mutation on individuals, create new generations evaluating the fitness of each individual of the population in the process. When the fitness level in the population reaches a satisfying value, a set of solutions are obtained. In the design of the genetic algorithm prototype, there are five objects considered that form the Order Holon, as shown in Figure 5:

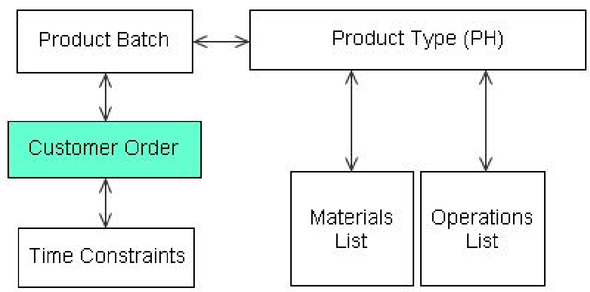

Fig.5. Customer Order Holon (OH) internal structure

The OH is centered on the customer order object that has two internal references: the product batch, which contains information about what type of products are being ordered and the time constraints for the order. The Product Batch object references the product type, or Product Holon (PH), for each of the products in the batch. The PH contains the list of required materials and the list of required operations for the product manufacturing. Where the order is decomposed and third parties are consulted about the possibility to provide raw materials or to perform operations and services, the materials list and operations lists are augmented with the outsource possibility and costs associated. The genetic algorithm proposed would generate a near optimal solution consisting in deciding which raw materials and operations will be outsourced and which will be completed in-house. This information together with the total costs computed as a simple sum, will be stored in the OH and returned as a response to the customer offer request. The individual considered by the genetic algorithm can be represented as an array of *boolean* values as shown in Fig. 6.

The fitness function computes the fitness value for each individual in the population. The fitness is computed by evaluating a set of conditions against the computed product batch solution instance as follows:

Condition1 (Hard Condition) Availability: iterates all the materials and operations and checks if each is available in the selected configuration (outsourced or in-house). If all of them are available then 10 points are awarded to the individual.

Condition2 (Hard Condition) Scheduling Possible: Checks if all operations in the current configuration, can be performed at job-shop level. The production line scheduler is invoked in order to validate the solution from this perspective. If this condition is fulfilled 10 points are awarded.

Condition3 (Hard Condition) Resources Available: Iterates all operations that are performed in-house and checks if there are resources available to perform them. The random generation of the solutions won't assure that some operations for which resources are not available internally are always outsourced. The reason behind this approach is to assure a certain degree of tolerance in unexpected resource breakdown scenarios, either from third party providers or internal. If this condition is fulfilled 15 points are awarded.

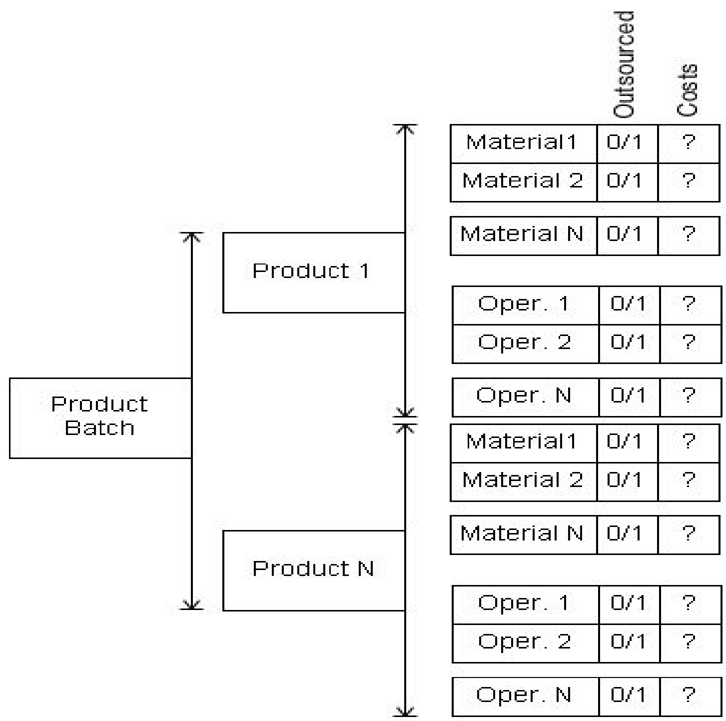

Fig.6. Solution Data Structure

Condition4 (Hard Condition) Time Constraint: Checks if the production line scheduler can schedule the order in such a way that it will complete in the time interval required by the customer. If this condition is fulfilled 15 points are awarded.

Condition5 (Soft Condition) Cost Condition: Iterates the entire population of solutions and computes the total cost for each individual. The population is sorted by cost and the first 10 solutions with the lower costs are awarded 5 points.

Condition6 (Soft Condition) Internal Resource Utilization: This condition is used to assure the internal resource utilization is as high as possible. It works in conjunction with condition 5 because normally the internal prices for the majority of the operations are lower than the external alternative. For each generated solution, the internal utilization is summed up and the population is ordered by this criteria. The first 20 solutions are awarded 3 points.

The score computed based on above evaluation is divided to the maximum score to obtain a fitness value between 0 and 1. Genetic algorithms have typically three operations: selection, crossover and mutation [19, 20, 21, 22]. The selection operation consists in computing the fitness value for each individual in the population and sorting the population based on the results. Then, the best 75% individuals are selected for crossover operation. The crossover operation represents the combination of two Product Batch instances that produce an offspring. The crossover operation is implemented by generating a random number X (crossover point) between 1 and N, where N is the number of products in the batch. The offspring will inherit the schedule of the first parent from Product 1 to Product X and the schedule of the second parent from ProductX+1 to ProductN. The mutation operation is applied to a randomly chosen subset of individuals in each generation, and consists in rescheduling of two of the Products from the data structure. The new schedule is generated by randomly assigning a new set of values for the Product instance.

The genetic algorithm itself has the following structure:

Step1: generateInitialPopulation()
Step2: while (best individual fitness < min_fitness){
Step3: do_crossover(best 75% individuals)
Step4: calculate_fitness(offsprings)
Step5: remove_worst(worst 35% individuals)
Step6: calculate_best_individual_fitness
Step7: }

As we can see in the above pseudo code, the genetic algorithm sorts the individuals after the crossover operation, based on the fitness and removes the worst individuals from the population. This assures the evolution from one generation to another.

5. Experimental Results

The customer offer request management concept presented in this paper was evaluated in the context of a pilot holonic manufacturing system of the CIMR Centre within University Politehnica of Bucharest, assembling T and H shaped mechanical products from raw components using 6 individual workstations, as presented in Fig. 7. In order to simulate both internal and external sourced raw materials and operations, we considered 4 workstations as internal and 2 as external. At the same time, two external suppliers, implemented as dummy web services were used. The costs associated with both raw materials from external suppliers and the 2 external workstations were implemented to have random costs behavior between two limits. This was done in order to simulate the real market behavior.

Fig. 7. Holonic Manufacturing System (CIMR Centre)

The input data was also a randomly generated list of customer offers requests, assuring that the random offer requests would be in the proximity of the manufacturing system maximal capacity. The intent is to force the algorithm to select also solutions that involve external operations while keeping the internal resource utilization at maximum. The simulation spans over 3 days of production, considering that all offers are accepted by the customers and executed on the production line. The inflow of customer orders is 1 per hour, with random product batch sizes.

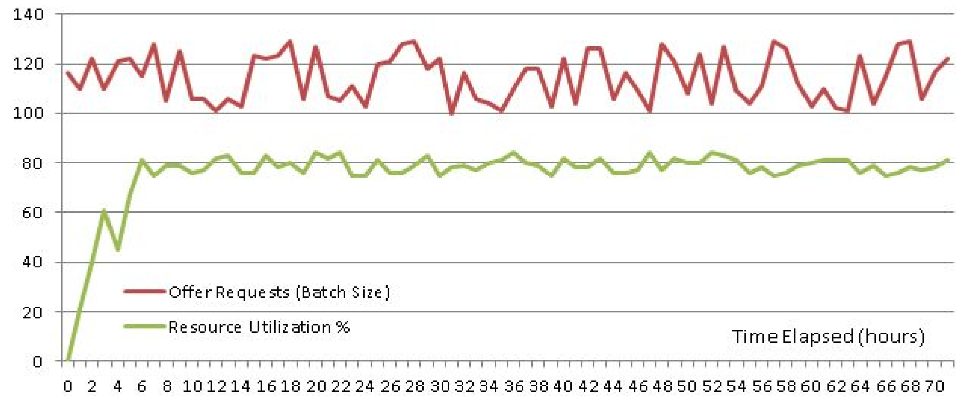

Fig. 8. Offer request inflow and internal resource utilization % over 72 hours

Fig. 8 illustrates the offer request inflow and the internal resource utilization percent obtained. We see an increase once the orders start accumulating and stabilization at around 80% for internal resource utilization for this test. This happens because the make-span considered for the product batch is at around 2 hours.

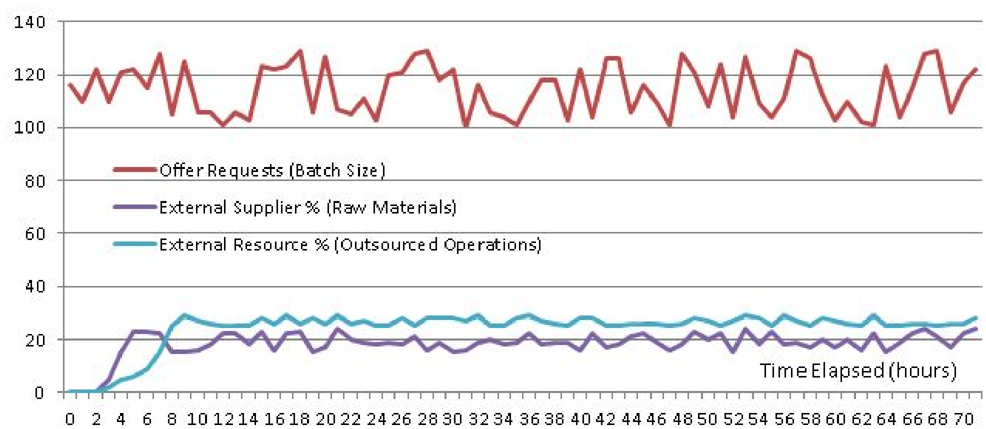

Fig. 9. External supplier usage % for raw materials and operations

Fig. 9 illustrates the percentage of outsourced operations (to the two workstations considered as external suppliers) and raw materials. We can observe that the system starts utilizing the two external workstations a bit faster than the two external parts suppliers. This is caused by the fact that initial stocks are enough for the first 3 offers and because the costs are configured to be lower, the genetic algorithm will favor the solutions that use stock raw materials. Finally Fig. 10 illustrates the preference given by the SIM (Supplier Integration Module)

module proposed to each of the two suppliers considered in this test. To evaluate the risk factor based selection criteria, a miss -delivery of raw materials has been simulated at supplier 2 after 24 hours of execution. The results clearly show that from that point on, the SIM module favors the first supplier. Requests to the second supplier are only sent when the first supplier is out of stock.

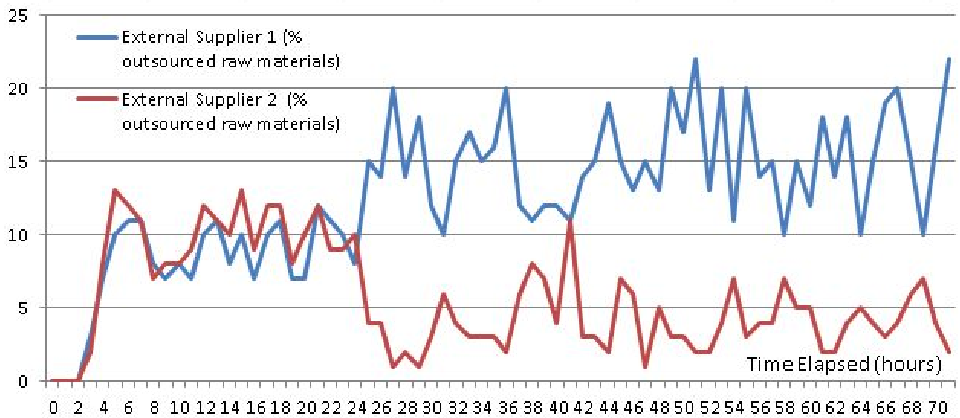

Fig.10. External supplier usage with risk factor determination

The risk factor selection criteria approach is however conflicting with the cost aspect, as there might be a situation where the supplier with a higher risk factor offers a better price for the raw materials. However, in long term, this approach is efficient in filtering out the external suppliers that can have a bad influence in the production process.

5. Conclusions

The genetic algorithm proposed in this paper represents a valid solution for handling customer offer computation in an automated and optimum fashion. Depending on the size of the problem domain the algorithm can be tuned to generate more accurate solutions in terms of fitness function, crossover rate or mutation percentage. The population size and the maximum number of generations also play an important role in the results generated. In practice when using a large population size, the fitness score for the solution is higher after a fixed number of generations. However, it has the drawback of longer execution time with CPU intensive behavior. For the above tests a population size of 100 with 40 generations was used with acceptable results in terms of algorithm execution time. Further work is focused on integrating the risk factor selection criteria in the genetic algorithm as an additional soft condition along with the price. It is expected that by doing so it will serve as a feedback factor and will allow a bad supplier to reduce the associated risk factor due to price and reliability conditions.

R E F E R E N C E S

[1] *Koestler, A.*, The Ghost in the Machine. Arkana Books. London, 1969
[2] *Van Brussel, H.V., Wyns, J., Valckenaers, P., Bongaerts, L., Peeters, P.*, Reference architecture for holonic manufacturing systems: PROSA. Computers in Indus-try, Special Issue on Intelligent Manufacturing Systems 37 (3), 255-276, 1998
[3] *Demirkan H., Kaufmann, R., Vayghan, J., Fill, H.-G., Karagiannis, D., Maglio, P.*, Service-oriented technology and management: Perspectives on research and practice for the coming decade, Int. J. on Electronic Commerce Research and Applications, Vol. 7, Elsevier, doi: 10.1016/j.elerap.2008.07.002, pp. 356-376, 2008
[4] MESA, SOA in Manufacturing Guidebook, White Paper #27, MESA Interna-tional, IBM & Capgemini, 2008
[5] *Borangiu, Th., Raileanu S., Anton F., Parlea M., Tahon C., Berger T., Trentesaux, D.*, Product-driven automation in a service oriented manufacturing cell, Proceedings of the Int. Conf. on Industrial Engineering and Systems Management, IESM'11, Metz, May 25-27, ISBN 978-2-9600532-3-4, 2011
[6] *Borangiu, Th.*, IBM Service Oriented Technologies and Management for Smarter enterprise, Proceedings of the 15th IEEE Int. Conf. on System Theory, Control and Computing ICSTCC (ed. M. Voicu), Sinaia, October 14-16, ISBN 978-973-621-322-9, IEEE Catalog no. CFP 1136P-PRT, pp. 97-102, 2011
[7] *Barata, J.*, The Cobasa Architecture as an Answer to Shop Floor Agility, in Manufacturing the Future, Concepts, Technologies and Vision, (ed. V. Kordic), ARS Germany, pp.908-923, 2006
[8] Real-World SOA: SOA Platform Case Studies, Forrester Research, Inc., 2005
[9] Service-oriented architecture, IBM Systems Journal, 44/, <www.ibm.research.com/journal/sj46-3.html>, IBM Corp, 2005
[10] *Vrba, P.*, Automation Agents for Manufacturing, Keynote presentation at the 2nd ECOLEAD Summer School, Prague, Rockwell Automation Inc., 2007
[11] *Brown W. A., Laird R. G., Gee C., Mitra T.*, SOA Governance: Achieving and Sustaining Business and IT Agility, IBM Press, ISBN: 978-0-13-714746-5, 2008
[12] *Ballentine J., Braghirolli C.*, End-to-end Integration with IBM Sterling B2B Integration and Managed File Transfer solutions, IBM Press, ISBN 0738436925
[13] Microsoft BizTalk, http://www.microsoft.com/biztalk/en/us/b2b.aspx, accessed December 2012
[14] *Borangiu Th., Gilbert P., Ivanescu N., Rosu, A.*, An Implementing framework for holonic manufacturing control with multiple robot-vision stations. Journal of Engineering Applications of Artificial Intelligence, Elsevier, ISSN 0952-1976, 2009
[15] *Barták, R.*, Mixing planning and scheduling to model process environments, PACLP, Manchester, UK, 2000
[16] *Borangiu, Th.*, A Service Oriented Architecture for Holonic Manufacturing Control, in Rudas I. (ed) Towards Intelligent Engineering and Information Technology. Vol. 243 of Springer Book Series Studies in Computational Intelligence, Springer Berlin/Heidelberg, 2009
[17] *Kaschel, J., Teich, T., Kobermik, G., Meier, B.*, Algorithms for the Job Shop Scheduling Problem - a comparison of different methods, citese-erx.ist.psu.edu/viewdoc/download?doi=10.1.1.89.154&rep=rep1&type=pdf, 2008
[18] ILOG, http://www-01.ibm.com/software/websphere/ilog/, 2012

[19] *Ferguson, D., Stockton, M.,* Service-oriented architecture: Programming model and product architecture, IBM Systems Journal, vol. 44, no.4, 2005
[20] *Erl, T.,* Service-Oriented Architecture: A Field Guide to Integrating XML and Web Services. Upper Saddle River, Prentice Hall, 2004
[21] *Catts A., Clair J. St.,* Business Process Management Enabled by SOA, IBM Redbooks IBM Form No. REDP-4495-00, 2009
[22] *Erich, G., Helm, R., Johnson, R., Vlissides, J.,* Design Patterns: Elements of Reusable Object-Oriented Software. Addison-Wesley. ISBN 0-201-63361-2, 1994