Multi-agent System Approach for the Strategic Planning in Ramp-up Production of Small Lots

Paulo Leitão, José Barbosa
Polytechnic Institute of Bragança
Campus Sta Apolónia, Apartado 1134
5301-857 Bragança, Portugal
{pleitao, jbarbosa}@ipb.pt

Pavel Vrba
Czech Technical University in Prague
Karlovo namesti 13
121 35 Prague 2, Czech Republic
pavel.vrba@fel.cvut.cz

Petr Skobelev, Alexander Tsarev, Daria Kazanskaia
Smart Solutions, Ltd.
Moskovskoye shosse 17, office 1201
443013 Samara, Russia
at@anarun.net

***Abstract*— This paper discusses the methodology for the development of a strategic planning tool, and particularly the architectural solution to be adopted, within an integrated ICT solution for the improvement of planning and scheduling systems for the manufacturing of small production lots of complex products. After the analysis of different possible architectural solutions, the paper proposes a hybrid solution that combines existing solvers with the multi-agent system principles. The proposed approach introduces several benefits, namely in terms of flexibility, robustness, and achievement of exploratory alternative solutions, playing interactive what-if simulation supporting decision-makers to take strategic decisions.**

Keywords-Strategic planning, multi-agent systems, ramp-up production, optimization methods.

I. Introduction

The production and ramp-up of complex and highly customized products, such as airplanes or shipyards, is characterized by higher uncertainty and instability of the processes, especially in case of small lot sizes. Frequently occurring errors prove that product design, production lines, suppliers, IT or logistics are not ready at the very first products of the series. Late requests for change by customers, and immature high technology products and processes are additional risks for this state of production. As example, the technology of the production is usually not well defined and can change often until it becomes reliable. It subsequently means that also people involved into the production are not familiar with the processes that change all the time, can spend more time on operations, make more mistakes, and require more guidance or step-by-step instructions through the processes. It is also important to monitor every step and catch all deviations as soon as possible to analyse them and shorten the technology adaptation and learning cycle. Supply channels are also not sufficiently reliable during the ramp-up phase and possible delays on all stages should be considered.

Besides the need for higher adaptability in short-term cases, ramp-up problems induce a higher variety of possible situations in long-term to be considered. The results achieved in long-term are strongly depend on the events that happen often in ramp-up. This means that ramp-up requires more what-if analysis to prepare production for different situations and integration of operational results into the strategic planning.

In case of small-lot production, the ramp-up phase becomes a natural part of the production process because the technology becomes mature only for a short time until the new lot of customized product is produced.

In such challenging scenarios, novel ICT-based approaches are required to develop mitigation strategies to respond faster to unexpected events. Therefore the knowledge base has to be enriched for real-time decision support, to detect early warning and to accelerate learning. ARUM (Adaptive Production Management), a collaborative project within the EC “Factory of the Future” initiative, intends to address these topics by introducing an innovative architecture based on novel ICT solutions to handle new challenges in production and ramp-up of complex and highly customized products, namely those in small-lot production. In particular, the focus is in the development of systems and tools for decision support in planning and operation using a new generation of service orientated enterprise information platforms, a service orientated bus integrating service-based architecture (SOA) and knowledge-based multi-agent systems (MAS).

The designed architecture considers an intelligent Enterprise Service Bus (iESB), enhanced by several functional modules, e.g., service registry enriched by dashboard features, ontology services and various gateways, to interconnect several tools aiming to provide functionalities for planning and scheduling activities. One of these tools is the strategic planning that allows the capacity planning for the mid- and long-term horizon (from several weeks to months), supporting decision-makers with useful information about alternative

978-1-4799-0652-9/13 $31.00 © 2013 IEEE
DOI 10.1109/SMC.2013.807

solutions [1]. This tool performs the planning based on the same or coarse specification of the technological processes as used in the operational scheduling, and also use the results of operational scheduling as a starting point for long-term planning. The strategic planner tool allow building alternative plans based on different alterations of the initial situation and different possible event flows happening in the horizon. This relates accordingly to the "what-if" and "simulation" modes of use of the strategic planner. This tool should consider the possibility to modify the resource pool, e.g. hire or reallocate staff, or create new shifts, analysing its economic efficiency.

The objective of this paper is to analyse the different possibilities to design the strategic planning tool within the ARUM context and define the main architectural guidelines for the adopted solution considering the problem requirements.

The rest of the paper is organized as follows. Section II overviews the ARUM vision and the architecture of the iESB platform and Section III discusses the strategic planning objectives and requirements. Section IV analyses the different possibilities to implement the strategic planning tool and Section V details the architectural solution combining existing solvers and multi-agent systems principles. At last, Section VI rounds up the paper with the conclusions.

II. The ARUM Vision and the iESB Platform

The requirements imposed to the ARUM project will be addressed by the intensive use of two emergent paradigms and technologies: multi-agent systems and Enterprise Service Bus (ESB). In ARUM, an ESB is used as backbone for supporting the interoperability among the developed tools. In this case, two ESBs will be considered, namely the open source JBoss ESB [2] and the proprietary TIE Smart Bridge [3]. To achieve a full interoperability across the entire ARUM solution, the ESB is enriched with a set of services and functionalities that will support the tools lifecycle from creation time until they are unplugged from the system, resulting on an intelligent enterprise service bus (iESB). Namely, it comprises the following main modules, as illustrated in Figure 1:

- *Registry service*, allowing the registration and the search of services offered by the tools plugged into the system. The registry service will be extended in such a way it allows for registering the dialog for administration of the service to be displayed within the administrator dashboard.
- *Ontology service*, allowing the access and manipulation of the ARUM ontology.
- *Data transformation*, allowing the conversion of a data format into other type.
- *KPI service*, aiming to make easier the collection of data for the calculation of performance indicators.
- *Security service*, aiming to add security to the exchanged messages inside the ESB.
- *ESB interlink*, allowing a safe, stable and transparent connection across different ESBs.
- *Gateways*, allowing the tools to access data from legacy systems, such as SAP, AS400, MES and SCADA.

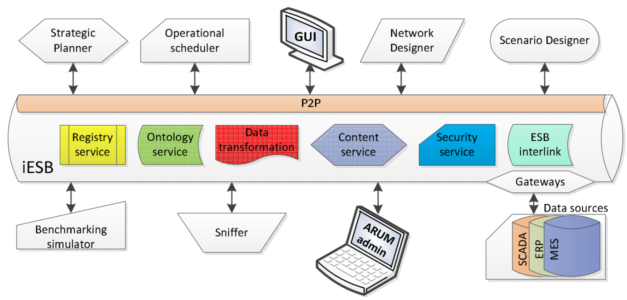

Figure 1 Architecture of the ARUM's iESB platform

Besides these core services, the iESB will also have a simulator for benchmarking purposes and a sniffer to gather and display all the messages exchanged among the bus. Finally, the iESB can be monitored and parameterized, in a graphical way, using the ARUM administrator dashboard.

A set of tools are placed in a different architectural layer, supporting all the ramp-up phases, from process and resource modelling to the real resource allocation, passing by a high level strategic planning.

In this process flow, the first tool to consider is the Network Designer (ND) that allows the manufacturing engineer to specify the assembly process and, in a generic way, the template to be instantiated à posteriori by the Scenario Designer (SD) tool. Briefly, the ND gets the process information from the ARUM ontology and the manufacturing engineer can fill in and fix some parameters. After this first phase, the station manager uses the SD to deeply specify the scene layout. By scene we mean the set-up of the particular work station in terms of available resources, such as tools, machines, parts and workers. The ND and SD tools can be considered as "preparation tools" in the sense they create the scenarios for the other tools to work on, namely the Operational Scheduler (OS) and the Strategic Planner (SP).

The OS is responsible for the low level scheduling of work orders to resources in a specific station. This tool will be developed using MAS technology and its architecture will get inspiration from swarms found in nature (e.g., fish schooling and bird flocking) [4]. Briefly, since it is out of scope of this paper, the OS will distribute the scheduling processing along the shop-floor. One of the main features of the OS is the adaptive event driven behaviour, which means that the tool rebuilds the schedule only in a part affected by the event. This allows to maintain large and detailed schedules in real-time without significant loose in quality. Although technically the horizon of operational scheduling is not limited, it makes sense to keep it shorter to have a shorter event processing time. Thus, OS focuses on schedules and decisions to be made in the scope of few days to few weeks.

Contrary to the OS purpose, the SP has a larger time wide concern with less detailed view on the processes. The main purpose of the SP is to aid the company decision-makers taking strategic decisions. The need for this tool is better explained using an example. Let's suppose that one company is producing two products with a given year production capacity. During the normal production, a rush order from a very important client appears, that exceeds the factory capacity. At this point, a strategic decision is imposed to the management board giving the following options: rejecting the order, which can imply losing the client; or accepting the order with the implication of cancelling or delaying the other on-going orders together with paying significant penalties or create additional shifts or reallocation of staff.

Regardless the decision to take, several constraints must be considered, such as penalties in case of contract breakdown, suppliers capacity, shop-floor layout and manpower restrictions, just to name a few. This set of constraints can increase exponentially, making the strategic decision very hard and erroneous.

III. STRATEGIC PLANNER TOOL

The main focus of this work is the analysis of the strategic planner tool within the ARUM's context.

A. *Objectives and Problem Description*

The SP tool solves the capacity planning problem of given orders for a long-term cases using a simplified technology specification, considering the costs of delays, resource reallocation, increase of resource pool, additional employee training or re-allocation, and partial execution of orders. SP should consider the skills and other resources required by different steps of order execution, and the interdependence of these steps. Additionally, SP considers capacities of factories, lead times, demand information (i.e. current orders to be fulfilled) and costs of extra shifts.

SP should automatically propose an effective decision on the sequence of execution of parts of orders (e.g. first to produce 80% of order A, then 100% of B, then 20% of A), and the steps (high-level processes) of execution of different orders (e.g. make preassembly for all orders in parallel in advance, then put all resources to finalize order A, then B). SP should propose a solution of how the existing resource pool (skills pool) should be distributed between orders considering the deadlines and cost of delays (e.g. put 90% of the electrical staff on order A and 10% on B for 2 months to be in time with A, then put everybody on B). SP should propose an efficient solution on the increase of resource pool considering the costs (e.g. train 3 people to perform electrical operations in two months, hire 1 person in one month and relocate 3 people now for 5 months).

Besides this, the SP should support an interactive way of revising the plan caused by users changing the costs definitions, supply constraints, adding more orders or proposing manual solutions on resource pools to be checked. This interactive mode can include what-if games and analysis of different event flows that can happen in production.

These alternative solutions are evaluated according to different criteria, namely volume penalties (imposed by supplier of complex parts that is in fact in the same situation of high peak demand) or on the contrary volume discounts (in case of simple parts), shipment costs, warehouse costs (considering limited capacity of own warehouse), penalties for delaying the order (given by contract), customer priorities, human resources re-allocation costs, moving parts of the production costs (e.g. from one factory to another one), etc.

B. *Requirements and Assumptions*

The SP tool has some assumptions that should be considered in the architectural design:

- The required response time depends on the mode in which SP is used. It is not usually critical and is usually triggered 3-4 times a year.
- In interactive decision making mode, the tool should return a result in few minutes in the worst case to let decision-makers to review the proposed solution and try to change the constraints several times to find a better balance.
- A mature and robust solver should be considered since the problem to be solved comprises a huge volume of data.
- An UI (User Interface) for the data management should be considered, e.g. to insert simulation data or visualize alternative solutions.
- The MAS technology should be considered to automatize the process and to provide more flexibility and efficiency to the adopted solution, e.g. allowing for achieving faster alternative planning solutions.
- The simulation of "what-if" scenarios should be considered.

The final decision about the architectural decision considers several functional and technical aspects, such as the time to reach solutions, quality of the solution and flexibility introduced.

IV. POSSIBLE ARCHITECTURAL SOLUTIONS

A pertinent question in the design of the strategic planner tool is the selection of the architectural solution that best fits the described objectives and requirements. In this section, two usual possibilities are analysed, namely considering only an existing solver and an agent-based solver.

A. *Solver Engine*

The first solution studied in this work is the most classical one, and presents a solver as the mean to reach possible solutions for given constraints, as illustrated in Figure 2. Briefly, the decision-maker uses the UI to parameterize the mathematical formulation of the problem, including a set of constraints and criteria, which will be later used by the solver to be optimized. When the parameterization is finished, the UI maps the problem formulation and criteria into the solver and sends a signal to start the solver execution.

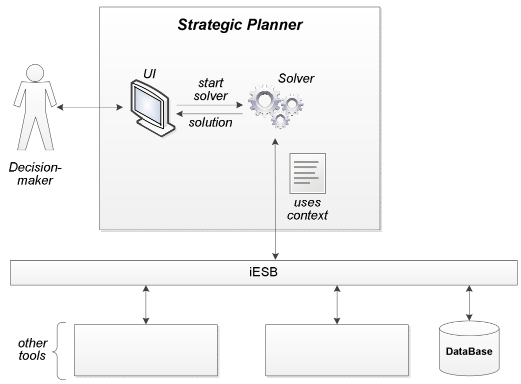

Figure 2 Architecture for a solver based solution

The solver runs the optimization method for the specific problem formulation using the current context, available in the production database and accessed by a gateway provided by the iESB platform. For this purpose, the solver collects data related to the current state of the system, e.g., the current scheduled tasks (e.g. resources allocation, production costs, delivery dates from suppliers and penalties in case of contract break) and the constraints to produce the desired manufacturing orders, both internal and external. Examples of internal constraints can be the skills of resources and operators, and shop-floor layout limitations; examples of external constraints can be the limited production capacity of the supplier.

After the optimization completion, the solver will reply back with the achieved solution, which will be displayed in the UI. The factory manager analyses the solution and can start a new iteration by re-parameterizing the problem, for example adjusting the constraints or criteria used by the solver.

A central component of such approach is the solver, which is software to solve a mathematical problem determining the optimum solution. Solvers implement different optimization algorithms, being possible to choose the algorithm according to the problem type, e.g. linear problems, quadratic problems and nonlinear problems. Several solvers are currently available, such as IBM ILOG CPLEX Optimization Studio, Xpress Optimization Suite, MOSEK, Gurobi Optimizer, KNITRO, GLPK (GNU Linear Programming Kit), LP_solve, Choco or CLP (Coin-or Linear Programming).

Several comparative studies are available in the literature, such as the one described in [5], which compares the performance of solvers using the same machine and the same problem (in this case using a case study focusing on solving linear problems). In the referred work, 87 mixed integer linear optimization problems from MIPLIB library [6] were considered. This study concludes that the solvers that obtain the best behaviour in the terms of speed are ILOG, Xpress and Gurobi, which need less than 120 seconds to solve the mixed integer linear optimization problem. The study also concludes that these solvers are the ones that solve more problems, i.e. more than 83% of all tested problems [5]. Having this in mind, a possible solver to be embedded in this architectural solution, could be IBM ILOG, which provides a set of well-known optimization methods and the advantage of being mature and well-known (widely used in companies and universities).

Two main conclusions can be drawn from this approach. The first one is that using existing and mature solvers, optimal solutions will be achieved allowing to reach the optimized strategic planning. The second conclusion is that for the case of several rounds of re-parameterization, the decision-maker must manually introduce the new parameters making this solution to be hardly managed and very time consuming (sequential achievement of solutions).

B. Multi-agent Based Solver

A second solution is to use the MAS paradigm to build the solver for the strategic planner tool. MAS [7] is a paradigm and a technology that advocates the creation of a set of distributed entities, called agents, to solve complex problems. Agents are autonomous and intelligent entities that have cooperation capabilities that enable the achievement of the system goals in a more robust, flexible and agile manner.

Several approaches can be found in literature that already use the MAS principles for solving optimization problems. One of the main focus areas of such usage is found in the transportation domain. An example can be found in [8] where a MAS solver for a vehicle routing problem is proposed. Other application, but in a different context, is the real time resource allocation using MAS concepts, described in [9] and [13]. Besides these references, some research addresses on techniques to be deployed into MAS to solve optimization problems. Regarding this subject, e.g. [10] overviews some techniques for space partitioning, agent's allocation (to sub-regions of the space divided) and dynamic creation of agents.

In such architecture, illustrated in Figure 3, the main difference relies on the solver part, which is now built up using the MAS principles. For this purpose, a set of agents representing the system components (e.g. orders and resources), interact with each other, trying to optimize the goals transmitted by the UI agent. Several algorithm methods can be considered to be implemented by the agent-based solution, such as GA (Genetic Algorithms) and PSO (Particle Swarm Optimization).

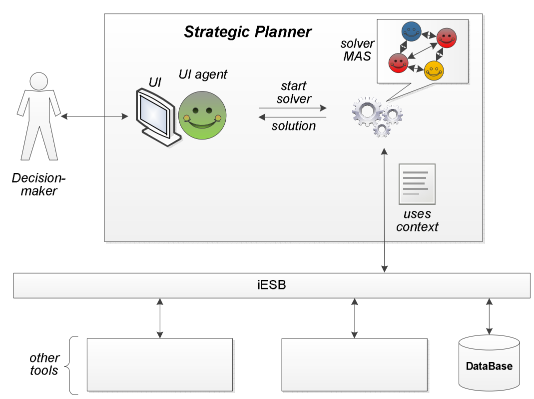

Figure 3 Architecture for a pure MAS based solution

Also the UI is now modified, being now an intelligent agent, named UI Agent, which is responsible for the interface with the decision-maker and responsible to set up and adjust the problem formulation and scenarios in an automatic manner according to the solutions sent by the agent-based solver. In other words, the UI agent can trigger the agent-based solver several times using automatic re-parameterization in case of receiving not satisfactory solutions (see Figure 4).

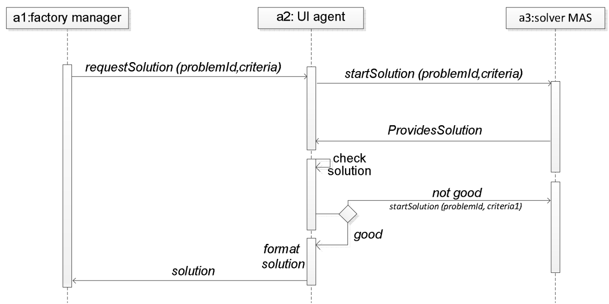

Figure 4 Pure MAS solver interaction pattern

This solution introduces flexibility in the sense that the UI agent can adjust some parameters and calculate a new alternative solution. On the other hand, the solver solution is more optimized for the strategic planner since the objective of the strategic planner is not so much about offering reactivity, which is a feature offered by the MAS. Other potential advantage of the MAS based solver is the possibility of adaptive re-scheduling that increases the reaction time in case of small changes in input data. This let the SP to check more options in a given time frame and is necessary to support interactive mode, when a user introduces small changes in the data to be checked (e.g. one order more, higher relocation cost or one electrician less).

However, this solution presents several disadvantages, namely the need to develop a solver from scratch, the missing maturity of the approach, the not optimal solution reached by this distributed approach, and the missing scalability of MAS technology for a huge size of the problem search. All these disadvantages make the architectural approach as a "shot in the dark" and may constitute a risk for the successful deployment into the factory plant.

V. Hibrid Multi-agent System Solution

Considering the best features of each described solution, the architecture adopted in this work combines an existing solver, which provides maturity, robustness and stability for the optimization algorithm, with a MAS infrastructure that offers the required flexibility to address the complex ramp-up problem. The idea is not to develop a new solver using the MAS principles, but instead complement an existing solver with MAS principles to explore the achievement of alternatives solutions for the strategic planner. Some works considering these ideas are reported in the literature, namely, [11] and [12].

Having these considerations in mind, the multi-agent system solution comprises the following classes of agents, as illustrated in Figure 5:

- *UI agent*, which is responsible for the interface with the user, providing the capability to insert simulation data and the visualization of alternative solutions.
- *Solver agent*, which is responsible to apply the optimization algorithm (i.e. by running an existing solver) to find a solution for the problem. Usually, only one solver agent will exist in the system, however a situation can be envisaged when more than one solver agent will be deployed to test different solvers on the same scenario and subsequently select the one providing best results.
- *Problem agent*, which possesses the description of the problem to be solved, i.e. contains the mathematical formulation of the problem. It is possible to consider more than one problem agent, each one representing a specific problem formulation.
- *Exploratory agent*, which is responsible to explore a solution for the existing problem by asking the solver agent(s). Usually, several exploratory agents are simultaneously running in the system, each one exploring solutions for different scenarios and criteria, such as minimizing the labour cost, inventory cost or tardiness cost.

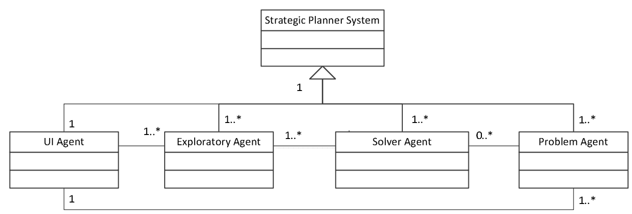

Figure 5 Static diagram of the agent classes for the strategic planner

The global system behaviour, reflecting the functioning of the strategic planner tool, emerges from the interactions between the agents according to proper cooperation processes, as illustrated in Figure 6.

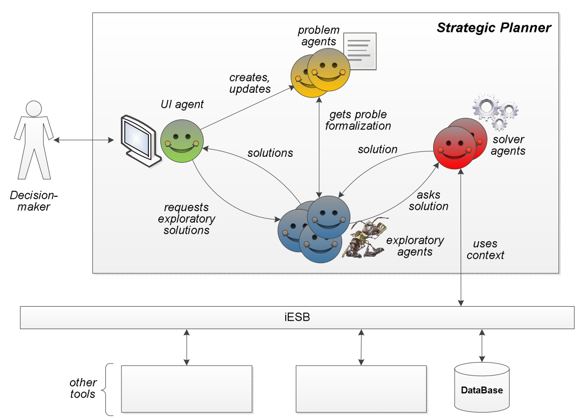

Figure 6 Architecture for a hybrid solution combining MAS with a solver

Briefly, the decision-maker uses the UI agent to select the problem to be explored, by the problem agents registered in the DF (Directory Facilitator) service of the MAS platform. The UI agent also allows to create and modify existing problem by interacting with the registered problem agents.

The use of problem agents to encapsulate the mathematical formulation of the problems, simplifies and makes more flexible the mechanisms, since the problem itself is not hard-coded and new problems can be easily added on the fly without having to stop, re-program, compile and start the system again. Note that the creation of a new problem description implies the launching of a new problem agent.

To execute the requested strategic planning, the UI agent launches several exploratory agents, each one responsible to get a solution for the problem considering specific scenarios and criteria (and also different solvers). These exploratory agents will apply learning techniques to adapt the optimization, constraints and criteria parameters to get alternative solutions (considering different scenarios). For this purpose, each exploratory agent needs to firstly get the problem description by asking the correspondent problem agent.

The exploratory agent consults the DF service to know the existing solver agents, and ask to each one a solution to the problem, passing the identification of the problem and the criteria to be considered, as illustrated in Figure 7.

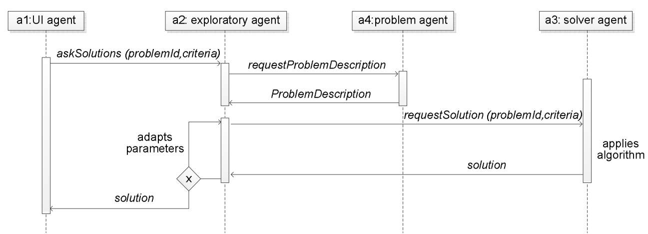

Figure 7 Interaction pattern regarding the exploration of a problem solution

The solver agent, using the problem description and the set of criteria passed by the exploratory agent, triggers the solver to run the optimization method, e.g. by invoking the IBM ILOG or the LP_SOLVER. Note that the solver will access the production data by using a proper gateway to the database and using the ontology services provided by the iESB infra-structure. At the end, the solver agent sends the achieved solution to the exploratory agent.

Each exploratory agent can combine solutions from different solver agents and also to slightly modify/relax the criteria and iteratively ask new solutions to solver agents (note this requires the introduction of learning mechanisms to allow a better behaviour of exploratory agents). When the exploratory agent decides that achieved solutions are suitable, it sends the results to the UI agent that compiles all received exploratory solutions and presents to the factory manager for evaluation.

This approach allows the achievement of several benefits, namely flexibility, robustness, and the dynamic achievement of exploratory alternative solutions. Such intelligent agent-based solution plays interactive what-if games and simulations, supporting decision-makers to take strategic decision based on significant and effective alternative solutions.

VI. CONCLUSIONS

This paper analyses the methodology for the development of a strategic planning tool, and particularly the architectural solution to be adopted, within an integrated ICT solution for the improvement of planning and scheduling systems for the manufacturing of small production lots of complex products. The proposed hybrid solution combines the reliability of existing solvers and the interactivity of adaptive fully multi-agent based solution, introducing several benefits in terms of flexibility, robustness, and achievement of exploratory alternative solutions at strategic level.

With the proposed approach, the strategic planner tool can be used to review plans in background ensuring in-depth analysis of options using conventional solvers and to notify decision-makers in case of significant change in KPIs. The same tool can be used for interactive what-if games and simulations, when it gives the decision-maker a fast solution for different alterations of the problem to be analysed.

Future work is related to the implementation of the strategic planning tool, considering an industrial use case.

ACKNOWLEDGMENT

The research leading to these results has received funding from the European Union Seventh Framework Programme FP7/2007- 2013 under grant agreement n° 314056.

REFERENCES

[1] D. Garvin, "Manufacturing Strategic Planning", California Management Review 35, n. 4 (summer 1993) pp. 85-106.

[2] JBOSS, http://www.jboss.org, (available on 15th May 2013).

[3] TIE Smart Bridge, http://businessintegration.tiekinetix.com/nl/contact/smartbridge-for-suppliers, (available on 15th May 2013).

[4] P. Leitão, J. Barbosa, D. Trentesaux, "Bio-inspired Multi-Agent Systems for Re-configurable Manufacturing Systems", *Engineering Applications of Artificial Intelligence*, vol. 25, n. 5, pp. 934-944, 2012.

[5] B. Meindl, M. Templ, "Analysis of Commercial and Free and Open Source Solvers for Linear Optimization Problems", Eurostat and Statistics Netherlands within the project ESSnet on common tools and harmonised methodology for SDC in the ESS, 2012.

[6] MIPLIB, "Mixed Integer Problem Library. Software", 2012 (available at http://http://miplib.zib.de/).

[7] M. Wooldridge, "An Introduction to Multi-Agent Systems", John Wiley & Sons, 2002.

[8] J. Vokrínek, A. Komenda, and M. Pechoucek, "Agents Towards Vehicle Routing Problems", Proceedings of the 9th Int. Conf. on Autonomous Agents and Multiagent Systems (AAMAS 2010), pp.773-780, 2010.

[9] P. Skobelev, "Multi-Agent Systems for Real Time Resource Allocation, Scheduling, Optimization and Controlling: Industrial Application", Proceedings of the 10th International Conference on Industrial Applications of Holonic and Multi-Agent Systems (HoloMAS'11), Springer Verlag, France, Toulouse, pp. 5-14, 2011.

[10] D. Villanueva, R. Le Riche, G. Picard, and R. Haftka, "Dynamic Partitioning for Balancing Exploitation and Exploration in Constrained Optimization: A Multi-Agent Approach", Proceedings of the 12th AIAA Aviation Technology, Integration, and Operations (ATIO) Conf. and 14th AIAA/ISSMO Multidisciplinary Analysis and Optimization Conf., 2012.

[11] L. Florin, "Design of a Multiagent System for Solving Search Problems", Journal of Engineering Studies and Research, vol. 16, n. 3, pp. 51-64, 2010.

[12] L. Qin, Q. Li, "A New Construction of Job-Shop Scheduling System Integrating ILOG and MAS", Journal of Software, vol. 7, n. 2, pp. 269-276, 2012.

[13] P. Vrba, M. Radakovič, M., Obitko, V. Mařík, "Semantic Extension of Agent-Based Control: The Packing Cell Case Study", Mařík, V., Strasser, T. Zoitl, A. (Eds.): Holonic and Multi-Agent Systems for Manufacturing, LNAI 5696, Springer-Verlag Berlin Heidelberg 2009, pp. 47-60.