PPSSNET : A PRODUCTION SIMULATION SERVER WITH DATABASE INTERFACE AND INTERNET-CAPABLE DATA ACQUISITION

Jonas L. Maia
jlmaia@comp.ufscar.br

Marcel B. Desco
mdesco@comp.ufscar.br

Orides Morandin Jr.
orides@dc.ufscar.br

Edílson R. R. Kato
kato@dc.ufscar.br

Artificial Intelligence and Automation Laboratory, Computing Department
Federal University of São Carlos
Rod. Washington Luiz, Km 253, ZIP 13565-905, São Carlos, SP, BRAZIL

Resumo Ultimamente, o mercado consumidor tem exigido maior responsividade de decisões de controle e programação da produção. Tais condições levaram a melhorias nos processos de manufatura, através do uso de ferramentas e técnicas modernas como simulação de eventos discretos. Neste contexto, a proposta original do PPSS foi realizada, como uma solução de suporte às decisões de reprogramação. Este artigo visa apresentar a expansão PPSSNet, a qual se comporta como um servidor de simulação da produção, com características como interfaceamento com bancos de dados comerciais e habilidade de trocar informações através da Internet ou outras fontes de dados e mídias de comunicação. Um exemplo de aplicação é apresentado e analisado, validando a proposta de expansão.

Palavras Chaves: Simulação, Planejamento e Controle da Produção, Sistemas Flexíveis de Manufatura, Internet e Redes Corporativas.

Abstract: Lately, consumer market has demanded higher responsiveness from production control and scheduling decisions. Such conditions have led to improvements in the manufacturing processes, by using modern tools and techniques like discrete-event simulation. In such context, the original PPSS approach was proposed as a support solution to production rescheduling decisions. This paper aims at introducing the PPSSNet expansion, which behaves like a production simulation server, with features like commercial database interfacing and ability to exchange information via the Internet or other data sources and communication media. A deployment example is presented and analyzed, validating the expansion approach.

Keywords: Simulation, Production Planning and Control, Flexible Manufacturing Systems, Internet and Corporate Networks.

1 INTRODUCTION

In the latest years, changes in the behavior of the consumer market and in the production systems have demanded higher responsiveness from production control and scheduling decisions[4][5][8][9]. Meanwhile, discrete-event simulation has been used to solve typical manufacturing problems like finding production bottleneck, validating plant layout and analyzing long-term production data[2][7][10][11][12]. However, computer simulation was not being deployed in short-term production issues. In order to fill this gap, our research group proposed in previous works an approach called PPSS (Production Planning System based on Simulation), a simulation-based tool designed to provide solutions to the above-mentioned demand for faster decisions regarding production rescheduling[1][3][13].

Concomitantly, we have observed an enormous growth of computer networks, including the Internet, Corporate Networks and other data communication architectures and technologies. Communication is becoming more and more reliable, with data transfer reaching much higher rates, besides the possibility of connectivity everywhere introduced by the wireless networks. All these developments have made possible the creation of new applications, as well as the further development of existing ones[1][2][5].

The work here presented aims at introducing a PPSS expansion, called PPSSNet. The main purpose of this approach is making the PPSS capable of acting like a Production Simulation Server, i.e., an application that executes in only one computer but can be accessed by many other client computers via the Internet and, as a consequence, via the private network of the company. This centralized approach is in consonance with the tendency of centralizing information, which is taking place in most of the companies around the world. Solutions like ERP (Enterprise Resource Planning) make information available to different levels of users, with different needs and interests; in spite of all this data being stored in a unique database, what improves consistency and reduces contradictory information. Following the same idea, the PPSS uses a unique database to keep up-to-date information about plant conditions (e.g. product stocks, work in process, machine status, etc.) and about the company strategies. Also, distinct user views are created by different Web-Pages and authentication systems, what makes the people involved capable of entering/accessing data related to their needs and interests.

With these distinct user views, a plant operator can manually enter data about a machine being taken down for preventive maintenance, while a salesman accesses the system from his
laptop and evaluates the possibility of meeting a delivery deadline when closing a deal. Once that the salesman has decided that the company can deliver the products in due time, the industrial engineer also uses the approach to evaluate the best machine scheduling to manufacture the just-sold products. All these activities are enabled by the centralized information and by the simulation engine, which takes these data as inputs, creates different simulation scenarios, carries out the simulation, and analyzes the output data, based on a set of pre-defined rules.

This way, the PPSSNet can account for a quality and time improvement of rescheduling decisions in manufacturing systems, as well as opening horizons for those who want to use the World Wide Web to have trustworthy plant analysis and delivery forecasts, demanding no knowledge on simulation software.

The paper is organized as follows: first, the PPSS approach is revisited, showing its main capabilities and characteristics. Then, the proposed extension is introduced, its features, modeling and technical details. In order to validate the work, a case study is carried out, showing the deployment of the PPSSNet in a plant built in our research laboratory. Finally, some conclusions are drawn corroborating the functionality of the presented solution.

2 THE ORIGINAL PPSS APPROACH

The Production Planning System based on Simulation (PPSS), previously proposed by our research group, assesses the performance of a set of possible actions or different possible arrangements using a commercial simulation software fed with information about the manufacturing plant status.

Whenever a rescheduling decision must be made, the PPSS provides the possibility of selecting a group of simulation models (previously designed to portray all relevant situations, either considering the current plant status or historical data), and evaluate the performance and the impacts of each one.

The simulation core model must be fed with information concerning the plant conditions (e.g. product stock, raw material, work in process, resources, etc.) supplied by the monitoring system and also with information related to the company strategies. It’s imperative that all information be always up-to-date, so that PPSS can provide the simulation software with the exact plant conditions at the decision-making moment.

Such information could originally be gathered from a monitoring system based on a supervisory system, a PLC (Programmable Logic Controller) coupled to sensors, or from data collectors used by plant operators. The expansion proposed in this paper aims at increasing the number of information sources to other systems like ERPs (Enterprise Resource Planning), other corporate databases and to the Internet/Corporate Networks.

An overview of the information flow between the different PPSS modules is shown in Picture 1.

According to the information inputs, some scenarios need to be picked up, so that the simulation rehearsals are carried out. Then, the scenarios that achieved a previously specified goal (e.g. best resource utilization, minimum stock levels, shortest delivery time) are classified according to their performance and the best one can be chosen.

The whole PPSS usage procedure is shown in Picture 2. This flowchart summarizes the main steps that must be followed in order to achieve the expected results.

The last paragraphs were primarily concerned with introducing the overall PPSS approach, as well as describing how it is used indeed. However, our research group has also worked on evolving and adapting an application methodology for it, so that it can be efficiently applied in any corporation, reducing deployment difficulties and assuring trustworthy simulation results. The main steps of this methodology can be summarized as: [3]

Step #1 – Problem Definition
- General system characterization by the customer;
- Establishment of objectives, goals and constraints.

Step #2 – Conceptual Model Formulation
- Formal techniques are used to accomplish a previous modeling of the system;
- Allows preliminary behavior analysis.
Step #3 – Data Preparation
- Identification of data relevant to the simulation, based on the problem definition;
- Data collection;
- Statistical analysis, to verify data representativity and the most realistic manner of inserting them into the simulation.

Step #4 – Computational Model Construction
- Develop the simulation model that represents the factory to be analyzed.

Step #5 – Verification
- Check the computational model logic, in order to avoid inconsistencies.

Step #6 – Validation
- Determine if the model actually reproduces the real system;
- Verify the degree of sensitivity of the output data, as well as their behavior.

Step #7 – Experiment Project
- Determine simulation scenarios, duration of simulation runs and the number of replications.

Step #8 – Simulation Runs
- Simulation of different scenarios is carried out.

Step #9 – Interpretation and Analysis
- The results generated by the previous step are interpreted and analyzed in order to support the rescheduling decision.

Step #10 – Reports and Documentation
- The results of the ninth step are summarized and organized in reports and documents.

From this methodology, steps 1 through 6 are expected to be accomplished during the PPSS deployment phase, while the others are done every time a rescheduling question arises. It’s also worth noting that the third step takes place during both phases, once that some data can be considered static in the simulation model, but some data changes dynamically according to the plant status.

3 THE PPSSNET EXPANSION

The PPSS presented in the previous section was rather standalone, once that it could only execute and access data located in the computer where it was hosted. The expansion here proposed aims at expanding the approach capabilities, making it able to act as a Production Simulation Server, a service that executes in one computer but that can be accessed by many other client computers via the Internet or Corporate Networks.

The expansion tries to take advantage of the current state of development of network technologies, as well as being in consonance with the tendency of centralizing information that is taking place in most of the companies. As already mentioned before, solutions like ERP (Enterprise Resource Planning) make information available to different levels of users, with different needs and interests; in spite of all this data being stored in a unique database, what improves consistency and reduces contradictory information. The PPSSNet follows the same idea, using a unique database to keep up-to-date information about plant conditions (e.g. product stocks, work in process, machine status, etc.) and about the company strategies.

Picture 3 shows an overview of the PPSSNet. It’s basically composed of:
- Web Interface with distinct views, each view showing just information relevant to a certain group of users;
- Web Front-end, consisting of the servers needed to make the Web-interfaces available. It includes basically an HTTP Server, and a Server-side Application Server, like ASP (Active Server Pages) or JSP (Java Server Pages);
- The Database-Simulation Model Interfacing, responsible for enabling the communication between the simulation environment and the commercial database used;
- The PPSS Core, which is primarily the original PPSS approach, only with some capabilities included;
- Triggering Mechanism, which starts the simulation process whenever a request arrives.

Below follows a more careful explanation of the PPSSNet new components believed to be more complex, namely the Interface with Distinct Views and the Database-Simulation Model Interfacing.

3.1 Database – Simulation Model Interfacing

Despite the understanding of the need for centralized information, one special issue must be addressed in order to make the approach capable of interacting with commercial databases. This issue is interfacing. Most of the simulation tools available are not capable of interacting with databases, neither directly, nor using ODBC (Open Database Connectivity) or other solutions. It’s commonplace for the commercial simulation tools available just being able to deal with raw text files and, less frequently, with network socket connections. This way, interfacing between database and simulation environment plays a key role when trying to acquire dynamical data and input it into the simulation model.
Also, one must be able to collect the simulation results not only from the simulation environment, but also from the same database that provided information to the simulation process. It brings data persistence to the results obtained, allows calculations and projections on such data, besides providing different manners of representing them to the user. [6] Both information flows, database to simulation environment and back, are dealt with by the interfacing system, what gives it a bidirectional capability.

In our proposal, the interfacing solution comprises three different parties. One is the database-side party, which is responsible for extracting/inserting data from/into the database tables, processing them and performing any calculations or adjustments necessary. The other is the information “means of transportation”, the pipe that will carry the data from the database-side peer to the simulation-side one and back. The last one is the simulation-side party, responsible for acquiring the information from the pipe in which data flow, processing it, loading it into the simulation environment and executing any start-up processes necessary for the model to run. It’s also up to the simulation-side peer, to collect the simulation results and insert into the information pipe back to the database peer, so that these data can be inserted into the base.

Picture 4 exemplifies the interaction between the interfacing parties from the approach.

Among the interfacing parties, the database-side peer and the information pipe are of relatively low complexity, once that the latter is mostly a text file with a pre-defined format and the former is a system with ODBC connection, which processes SQL commands and other calculations and inserts the results into the information pipe. However, the simulation-side peer must deal with more complex issues like reading information from the pipe, depacking this information into pieces of data, inserting these pieces into the simulation environment variables and carrying out the start-up processes. Also, it’s worth noting that the simulation core processes must have their codes changed because they won’t deal with real data anymore, but with data references implemented by indexed variables. Clearly, before simulation runs, all these data references must be loaded with actual data, and that’s one of the roles played by this side of the solution.

In Picture 5, we have shown a sketch of how simulation process code need to be organized in this new approach. At first in the execution timeline, are the input interfacing processes, in order to load the necessary data into the variable references. Then, in order to initialize the simulation model environment, come the start-up processes. Afterwards, we have the core simulation processes, which implement the product manufacturing operations, warehousing, transportation, resource maintenance, and so on. Finally, there are the output interfacing processes, which collect the results and send them to the database.

3.2 Web Interface and Distinct Views

Besides exchanging information with databases, the PPSSNet must also exchange information via the Internet. This information may either regard the plant status, or be a request to evaluate a set of scenarios in order to support a rescheduling decision, for example. In order to fit those different kinds of information, distinct user views are created by different Web-Pages and authentication systems, what makes the people involved capable of entering/accessing data related to their needs and interests.

With these distinct user views, a plant operator can manually enter data about a machine being taken down for preventive maintenance, while a salesman accesses the system from his laptop and evaluates the possibility of meeting a delivery deadline when closing a deal. Once that the salesman has decided that the company can deliver the products in due time, the industrial engineer also uses the approach to evaluate the best machine scheduling to manufacture the just-sold products.

Each of the above mentioned activities comprises a different subset of data. For example, the plant operator must access a Web-page with the plant resources and an option to inform that a certain resource is out-of-order. The salesman just needs to input the amount of products to be manufactured and access an estimation of the delivery date. The industrial engineer, in turn, must be able to define the goals that he expects the plant to achieve and obtain comparative data from re-scheduling possibilities. Picture 6 illustrates those different views and data subsets.

![Picture 6: Views and Data subsets](image)

![Picture 5: Simulation process code organization](image)
All these activities are enabled by the centralized information and by the simulation engine, which takes these data as inputs, creates different simulation scenarios, carries out the simulation, and analyzes the output data, based on a set of pre-defined rules.

4 A DEPLOYMENT EXAMPLE

In order to validate the approach, a case-study was carried out aiming to, using the proposed PPSSNet expansion (its Internet interfacing and service server availability), evaluate the impact of a resource maintenance on the possibility of meeting a delivery deadline before closing a deal. This analysis was made using the experimental plant being built in the Artificial Intelligence and Automation Lab.

4.1 The modeled plant

The simulation model was built to represent a plant composed of 6 manufacturing machines (being the first four ones for tooling and the last two ones for finishing), 3 AGVs (Automated Guided Vehicles), 2 input and output queues, and an AGV maintenance station. This plant layout represents the third phase of a flexible manufacturing system that’s being installed in the Artificial Intelligence and Automation Laboratory, at the Federal University of São Carlos. Picture 7 shows the graphical simulation model for the mentioned plant.

4.2 The manufactured products

The factory manufactures three different kinds of products. For every product, a set of tooling machines and of finishing machines are defined. Product “1” must pass by two tooling process and by one finishing process. The first tooling process is done either at the machine #1 or at the #2 (with a 40% - 60% probability for each one, respectively). The second tooling is done using either the machine #3 or the #4 (66% - 34%, respectively). Finally, the finishing action takes place at machine #5 or #6 (25% - 75%).

Following the same idea, picture 8 shows the Petri-Net representation of the production process for every product.

Each transition in the Petri Net above represents an operation performed by a machine. Depending on the product being manufactured and on the resource being used, each of those operations demands a specific period of time, as represented in table 1.

<table>
<thead>
<tr>
<th>Product</th>
<th>Mch #1</th>
<th>Mch #2</th>
<th>Mch #3</th>
<th>Mch #4</th>
<th>Mch #5</th>
<th>Mch #6</th>
</tr>
</thead>
<tbody>
<tr>
<td>Product 1</td>
<td>3.9</td>
<td>1.5</td>
<td>1.3</td>
<td>2.9</td>
<td>3.6</td>
<td>1.75</td>
</tr>
<tr>
<td>Product 2</td>
<td>1.3</td>
<td>3.15</td>
<td>-</td>
<td>2.0</td>
<td>1.6</td>
<td>3.85</td>
</tr>
<tr>
<td>Product 3</td>
<td>1.7</td>
<td>3.75</td>
<td>4.1</td>
<td>1.5</td>
<td>4.0</td>
<td>1.8</td>
</tr>
</tbody>
</table>

In order to represent the actual market demand, a production mix of 45%, 30% and 25% was adopted for products “1”, “2” and “3” respectively.

4.3 Using the system via the Internet

This section specifically addresses the usage of the proposed expansion, demonstrating the already mentioned different views created, in order to assess the possibility of meeting a product delivery deadline.

In this example, three views were defined. The operator one, which inputs into the system information about the status of the resources; the salesman one which makes a request to estimate the total amount of time needed to manufacture the order of products, and the industrial engineer one, who needs to make a
decision about the scheduling of the manufacturing plant, solving questions like which production-line input sequence to use, and so on.

Picture 9 shows the HTML pages generated for each view. They are dynamically generated by JSP (Java Server Pages) via a Java-based server-side application that creates the page contents, collects data and transfers data from/to the approach database.

In the operator area, he informs that resource #3 is out of order and expected to remain inoperative for 12 hours. The salesman inputs the total amount of products expected to be sold (4000 products, being 1800 products “1”, 1200 products “2” and 1000 products “3”). After inputting such information, with one click he triggers the simulation process and after a short processing time, he is informed that in approximately 8 days the company is able to deliver the products.

At the same time, the industrial engineer is informed that, in this case, the best production-line input sequence is product “2” first, then product “1” and finally product “3”; as well as that the best batch size is 100.

5 CONCLUSIONS
Concluding, the solution provides the company with an integrated simulation approach that is accessible anywhere. In this sense, the PPSSNet can account for a quality and time improvement of rescheduling decisions in manufacturing systems, as well as opening horizons for those who want to use the World Wide Web to have trustworthy plant analysis and delivery forecasts, demanding no knowledge on simulation software.

It is also important to highlight that, by inserting production simulation results into a database, the PPSSNet copes with data persistency, making them available for future analysis, for instance, by means of BI (Business Intelligence) solutions.

Further research is being carried out in order to enhance the features of the PPSS/PPSSNet. For example, collaborative manufacturing issues are being addressed by the model, improving its capabilities regarding to production distributed on various manufacturing sites.

6 BIBLIOGRAPHY