

Maria Leonilde R. Varela<sup>1</sup>, Joaquim N. Aparício<sup>2</sup>, Sílvio do Carmo Silva<sup>3</sup>

<sup>1</sup>University of Minho, School of Engineering, Dept. of Production and Systems

Email: [leonilde@dps.uminho.pt](mailto:leonilde@dps.uminho.pt)

<sup>2</sup>New University of Lisbon, Faculty of Science and Technology, Dept. of Computer Science

Email: [jna@di.fct.unl.pt](mailto:jna@di.fct.unl.pt)

<sup>3</sup>University of Minho, School of Engineering, Dept. of Production and Systems

Email: [scarmo@dps.uminho.pt](mailto:scarmo@dps.uminho.pt)

PORTUGAL

*This paper describes a distributed knowledge base for manufacturing scheduling. A great variety of scheduling problems can occur in manufacturing. For solving different problems, usually different methods are required. The distributed knowledge base enables sharing information about scheduling problems and corresponding solving methods in a widened search space. These methods can be remotely available and accessible through the Internet. Running several methods enables obtaining alternative results for a given problem, consequently, contributing for a better scheduling decision-making. An important aspect is that end users and scheduling methods' providers alike can easily and continuously update this knowledge base.*

## 1. INTRODUCTION

Competitive organizations are operating today in global and worldwide markets. Therefore, the competitiveness of enterprises and the quality of working life, in today's knowledge-based economy, are directly tied to the ability of effective creation and share of knowledge both, within and across organizations.

Manufacturing scheduling is a complex task that involves a wide range of knowledge. Scheduling problems are often complicated by large numbers of constraints, interrelating tasks, resources and events external to a manufacturing system. Moreover, slight differences in the manufacturing environment originate distinct problems, which even though being closely related, require different solving methods to be applied.

The effective and efficient resolution of those problems begins with the identification of suitable scheduling methods for solving them. When there are alternative methods to solve a problem alternative solutions can be obtained, which should be evaluated against specified criteria or objectives to be reached. Thus, users are able to properly solve a problem, through the execution of one or more

scheduling methods, local or remotely available and accessible through the Internet, and, subsequently, select the most suited solution obtained.

This work attempts to offer new possibilities for carrying out manufacturing scheduling, following the approach of solving problems through a web-based decision support system. The system follows a peer-to-peer computing model, which permits sharing scheduling knowledge by means of a distributed knowledge base (DKB). This distributed scheduling repository enables accessing knowledge arising from an extended range of contributors and, therefore, providing a widened search space. This infrastructure is based on the principles of virtual organizations (VO) [1, 2].

The system permits the characterization of each problem to be solved and, then, the access to corresponding solving methods. For problem identification, a problem classification model that includes a set of parameters is used. This model enables specifying problem classes to which real problem instances belong [6, 7] and for which, hopefully, suitable solving methods can be found in the DKB. The data representation model for scheduling problems and related concepts is based on XML (extensible markup language), which is used as a specification language for scheduling data representation and processing on the Internet [6, 7].

This paper is organized as follows. The next section briefly describes the nature of scheduling problems and the underlying classification model. Section 3 presents the web system's distributed knowledge base (DKB) for supporting the scheduling decision making process, by any end-user who wishes to solve a problem, and describes the underlying peer-to-peer framework. Moreover, a document type definition (DTD) and the corresponding XML document about scheduling methods specification are shown in order to better explain the DKB updating process. Finally, in section 4, some conclusions are reached.

## 2. MANUFACTURING SCHEDULING PROBLEMS

Manufacturing scheduling focuses on the efficient allocation of one or more resources to tasks over time. It is an important activity to be performed for a company to achieve competitive production, which usually means to deliver products on time and to use resources efficiently. "Good" orderings to perform a series of given tasks have to be found, whereby specific objectives shall be optimized.

In order to perform the scheduling process it becomes necessary to clearly specify the problem to be solved. Manufacturing scheduling problems have a set of characteristics that must be clearly and unequivocally defined.

Due to the existence of a great variety of scheduling problems, there is a need for a formal and systematic manner of problem representation that can serve as a basis for their classification. A framework for achieving this was developed by Varela et al. [6, 7], based on existing notations available in the literature. This framework allows identifying the characteristics of each problem to be solved and it is used as a basis for an XML-based problem specification model developed [6, 7].

The referred framework for problem representation includes three classes of notation parameters for each corresponding class of problem characteristics, in the form of  $\alpha|\beta|\gamma$ . The first class of characteristics, the  $\alpha$  class, is related to the

environment where the production is carried out. It specifies the production system type ( $\alpha_1$ ) and, eventually, the number of machines that exist in the system ( $\alpha_2$ ). The second class allows specifying the interrelated characteristics and constraints of jobs and production resources, which are expressed by the  $\beta$  ( $\beta_1 \dots \beta_{14}$ ) parameters, and also the performance criterion, which is the third class ( $\gamma$ ). Some important processing constraints are imposed by the need for auxiliary resources, like robots and transportation devices and/or the existence of buffers, among others factors. The evaluation criterion, the third class of parameters, may include any kind of performance measure, namely multi-criteria measures [6, 7].

An example of use of this notation is "F2|n|Cmax" which reads as: "Scheduling a set of  $n$  independent jobs, on a pure flow shop (F), with 2 machines, in order to minimize the maximum completion time or makespan (Cmax). Due to the absence of some characteristics in this problem characterization, it is assumed that they are defined by default. Thus, for example about the possibility of job preemption (pmtn), the jobs are non-preemptable, similarly no job arrival times are specified ( $r_j$ ), which means that we are in presence of a static scheduling problem, with all jobs being ready to be processed at the same time, let us say at time zero. Moreover, the jobs are independent, as no precedence relations were defined (simple precedences, prec, or other type of precedences) and they have arbitrary processing time lengths, because no processing time restrictions are specified.

Good schedules strongly contribute to the company's success. This may mean meeting deadlines for the accepted orders, low flow times, few ongoing jobs in the system, low inventory levels, high resource utilization and, certainly, low production costs. All these objectives can be better satisfied through the execution of the most suitable scheduling methods made available through a distributed knowledge base, which enables searching for appropriate methods to solve each particular problem.

### **3. DISTRIBUTED KNOWLEDGE BASE**

The last tendencies show that computing environments are characterized by increasing heterogeneity, distribution and cooperation, where distributed knowledge bases play an important role [9].

Knowledge usage in computer systems directly depends on knowledge representative schemes. The standardization of extensible markup language (<http://www.w3.org/XML/>) on the Internet gives new opportunities in such direction. XML provides general markup facilities that are useful for data interchange. The web system described in this paper is based on XML modeling and related technologies.

The system is able to quickly assign methods to problems that occur in real world manufacturing environments and solve them through the execution of one or more appropriate implemented methods that are local or remotely available and accessible through the Internet.

The selection of one or more specific scheduling methods for solving a given problem is made through a searching process on the distributed knowledge base. The matching process, between problems and methods is performed by a built-in prolog search engine, which was developed using the SWI-Prolog V.5.2.1. free software tool available at <http://www.swi-prolog.org/>.

Figure 1 shows a general outline of the system’s architecture.

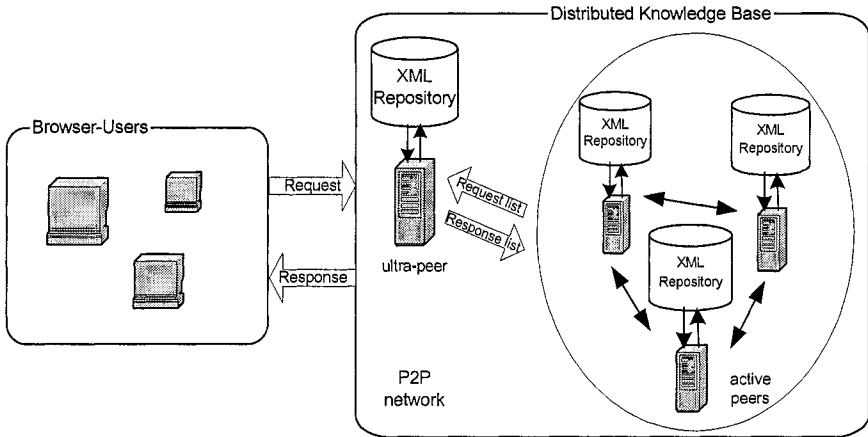


Figure 1 – Web system architecture

Figure 2 illustrates the main system processes, which also include knowledge insertion, about scheduling problems and solving methods, and correspondent information searching. Users can make requests for visualizing scheduling problem classes and methods’ information or even browse information about other concepts presented by the system. The data can be shown in different views, using existing XSL (extensible stylesheet language) documents, adequate for each specific visualization request.

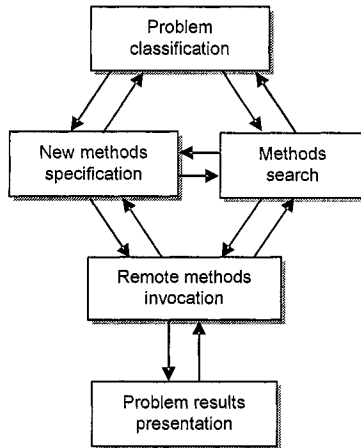


Figure 2 – Main system processes

The system has been designed and implemented as a web service (<http://www.w3.org>) using the XML-RPC (extensible markup language – remote procedure call) protocol [8] and will be available soon through the <http://www.dps.uminho.pt/> web site. In a web service a certain method accepts as input a problem definition and returns a result in some particular form. Different methods’ implementations may provide results in different forms, and the system

must have a description of them in order to format them according to the problem output to be returned to the client as the very last step of the service.

The system enables different ways of problem results presentation and storage. The result from running a method implementation on the given problem instance can be delivered to the client as an XML file and/ or can be transformed into some more expressive output, like a Gantt chart or other data representation formats, like tables.

### **3.1 Peer-to-peer network**

In networked environments, distributed knowledge bases and intelligent brokers, for information retrieval from specialized servers and knowledge repositories distributed across the Internet, enable to establish high-quality problem solving, through knowledge and resource sharing. In this context, emerging peer-to-peer technology and appropriate networks, suite well to the increasingly decentralized nature of modern companies and their industrial and business processes, whether it is a single enterprise or a group of companies [4]. The P2P framework provides the capabilities that allow users, or peers, to directly interact with each other [4, 5].

The web application described in this paper follows a P2P computing model. A set of peers, contributing with a local knowledge base component, composes a DKB as a P2P network. The P2P network has the capability of allowing a direct-interaction between the peers, which turns the computing environment decentralized, namely in terms of storage, computations, messaging, security and distribution. One of the greatest benefits of this P2P network, in the context of this work, is to easily support the concept of community. Consequently, it is possible for peers to organize themselves into groups that can collaborate with each other in order to achieve certain goals. One of the main goals aimed at in this work is the collaborative improvement of the resolution of manufacturing scheduling problems. This is possible through the DKB for manufacturing scheduling by accessing several different scheduling approaches. This is achieved by providing a mechanism that allows the members of the P2P network to share their scheduling knowledge and scheduling methods.

As previously referred the DKB scheduling system is based on the principles of VO. In this VO, each peer contributes with a local knowledge base (KB) in the P2P network. Each one can then be seen as a VO member or partner interested on solving scheduling problems. Whenever a member stores knowledge in its KB component, he/she is automatically contributing to the enrichment of the whole distributed knowledge repository, which is available to all the members of the VO.

Some peers of the organization can also act as ultra-peers. These special peers have the additional functionality of owning the list of the peers that belong to the VO. Such list contains information about the VO members and a flag that indicates their current state, which can be active or non-active. An ultra-peer also serves the purpose of configuring the P2P network as an open system, allowing any external user to join the organization, or as a closed system in the sense that only the nodes belonging to a certain company or domain can join the organization.

Each active peer is continuously listening for requests from other peers or from browser-like users. When a request reaches a peer, it firstly asks to one of the ultra-peers for the list of other active peers. Next, it propagates the request to all the peers of that list. Once the replies have been returned from the contributing active peers,

the compiled results are presented to the user in order to fulfill the request, as previously illustrated in Figure 1.

At any time, external users can join the VO and configure themselves as active peers. This can be easily done, by just installing a set of common components that compose the interface for accessing the network and its DKB. When a new (ultra)peer joins the VO it sends a request with its address to the ultra-peer(s) that represent the root domain of the P2P network, which is guaranteed to be always available. The ultra-peer(s) register the new (ultra)peer address, which is dynamically broadcasted to the remaining ultra-peer(s) of the current list. The VO members can join and stay connected or disconnect and leave the P2P network whenever they want, which configures very dynamic features to this VO model.

### 3.2 Updating the knowledge base

The web system enables introduction, validation, and transformation of manufacturing scheduling data. These processes are mainly controlled by DTD and XSL documents stored in the distributed knowledge base and all the scheduling information is stored in XML documents, which are validated according to associated DTDs, before being put in the corresponding knowledge base [6, 7].

At each peer the knowledge base can be continuously improved with new problem descriptions and available solving methods.

In the Internet many implementations may exist for a given method. From the point of view of the web system two implementations of the same method may differ if, for example, they differ on its outputs. Unfortunately, not all implementations work in the same way. Therefore, for the system to be able to match problem instances to resolution methods and to retrieve and use implemented methods available, in a programmatic way, they must also be described within the system. This description must include, among other things, the uniform resource locator to the running method and its signature, which, in turn, includes the definition of the parameters that are necessary for its invocation (inputs) and its output format.

The scheduling methods and their implementation details are described by a given DTD. Listing 1 illustrates this DTD for specifying the methods' information, such as their signatures, which are subsequently used to invoke the methods as web services and for other relevant information retrieval.

```
<!ELEMENT methods (method)*>
<!ELEMENT method
(id,name,url?,problem_class,method_class?,reference,complexity?,protocol?,signature?,gant?)>
<!ELEMENT id (#PCDATA)><!ELEMENT name (#PCDATA)>
<!ELEMENT url (#PCDATA)><!ELEMENT problem_class (#PCDATA)>
<!ELEMENT method_class (#PCDATA)><!ELEMENT reference (#PCDATA)>
<!ELEMENT complexity (#PCDATA)><!ELEMENT protocol (#PCDATA)>
<!ELEMENT signature (input,output)>
<!ELEMENT input (param | array | matrix)+><!ELEMENT param (#PCDATA)>
<!ATTLIST param name CDATA #REQUIRED type CDATA #REQUIRED control (submit)
#IMPLIED>
<!ELEMENT array (item+)>
<!ATTLIST array from CDATA #FIXED "1" to CDATA #REQUIRED control (submit) #IMPLIED>
<!ELEMENT item (#PCDATA)>
<!ATTLIST item name CDATA #REQUIRED type CDATA #REQUIRED>
<!ELEMENT matrix (item+)>
```

```
<!ATTLIST matrix lines CDATA #REQUIRED columns CDATA #REQUIRED control (submit)
#IMPLIED>
```

```
<!ELEMENT output (param | array | matrix)+><!ELEMENT gantt (#PCDATA)>
```

#### Listing 1 – DTD document sample about methods specification

Many scheduling methods may be more or less adequate to solve a given class of problems. In the methods distributed knowledge base the system records the scheduling method(s) that can be used for solving a certain problem class. Searching for the adequate methods, for a given problem, is performed by matching the problem details with the methods' characteristics, a process performed by the built-in prolog engine of each peer. The methods are usually available in the knowledge base of the peers belonging to the VO but they can also be found in other sites not belonging to the community.

Listing 2 shows a sample of the XML document about scheduling methods. It illustrates the information related to the implementation of the Johnson's Rule [3] for solving problem instances belonging to the F2|n|Cmax class described in section 2. This document is validated against the corresponding DTD, previously shown in Listing 1, before being put in the corresponding knowledge base component.

```
<?xml version="1.0" encoding="UTF-8"?>
<!DOCTYPE methods SYSTEM "methods.dtd">
<methods>
  <method>
    <id>32</id><name>Johnson</name><url>http://localhost:6002/RPC2</url>
    <prob_class>F2|n|Cmax</prob_class><method_class>Sequencing Rule</method_class>
    <reference>Johnson,1954</reference>
    <complexity>Maximal Polynomially Solvable</complexity><protocol>XML-RPC</protocol>
    <signature>
      <input>
        <param name="n" type="integer"/>
        <param name="m" type="integer" control="submit"/>
        <matrix lines="m" columns="n">
          <item name="job" type="string"/><item name="machine" type="string"/>
          <item name="p" type="double"/>
        </matrix>
      </input>
      <output>
        <param name="Cmax" type="double"/><param name="sequence" type="string"/>...
        <matrix lines="m" columns="n">
          <item name="start" type="double"/><item name="finish" type="double"/>...
        </matrix>
      </output>
    </signature>
    <gantt>yes</gantt>
  </method>...
</methods>
```

#### Listing 2 – XML document sample about methods specification

The inputs include the definition of a parameter  $n$ , for the number of jobs to be processed, a parameter  $m$ , for the number of machines, and a set of three items organized as a matrix structure, which represent the job name, the machine name and the processing time  $p$  of each job on each machine. There is also the definition for the method's output following the same lines. After a method definition has been inserted in a local KB it becomes immediately accessible to any further information retrieval. For the example given, after the insertion of the Johnson's method

definition any methods search that is a match for the F2|n|Cmax problem class will include this method in the search results. The methods' definitions are also used in the automatic generation of interfaces for methods invocation and corresponding inputs introduction and subsequently outputs presentation.

#### 4. CONCLUSIONS

In this paper a web system based on a peer-to-peer (P2P) infrastructure and a distributed knowledge base (DKB) is presented. The DKB is spread through a set of members (peers) forming a virtual organization. These members can store information about methods for solving different kinds of manufacturing scheduling problems, as well as their implementations. Each peer, in a decentralized computing environment, is able to directly interact with each other, as well as with any other external user, in order to enable executing diverse scheduling functions, including the ability to represent different scheduling problems, searching for appropriate solving methods and running methods that are made available and accessible by the DKB, through this P2P network. As running different methods enables obtaining different solutions for a problem, the system contributes for a better decision-making process, enhanced by means of the collaboration among the peers forming the virtual organization.

#### 5. REFERENCES

1. Camarinha-Matos, L. M.; Afsarmanesh, H., The Virtual Enterprise Concept, In: *Infrastructures for Virtual Enterprises, Networking Industrial Enterprises*. Kluwer Academic Publishers, 1999, pp. 3-14.
2. Camarinha-Matos, L. M.; Afsarmanesh, H., Design of a Virtual Community Infrastructure for Elderly Care, In: *Collaborative Business Ecosystems and Virtual Enterprises, PRO-VE'02*, Sesimbra, Portugal. Kluwer Academic Publishers, 2002, pp. 439-450.
3. Conway, R. W.; Maxwell, W. L.; Miller, L. W., *Theory of Scheduling*. England: Addison-Wesley Publishing Company, Inc., 1967.
4. Papazoglou, M.P.; Krämer, B.J.; Yang, J.; Leveraging Web-Services and Peer-to-Peer Networks, In: *Proceedings of Advanced Information Systems Engineering, 15<sup>th</sup> International Conference, CaiSE*, Klagenfurt, Austria. June 16-18, 2003, pp. 485-501.
5. Terziyan, V.; Zharko, A.; Semantic Web and Peer-to-Peer: Integration and Interoperability in Industry, Industrial Ontologies Group, MIT Department, University of Jyväskylä, Finland (<http://www.cs.jyu.fi/ai/vagan/papers.html>).
6. Varela, L. R.; Aparicio, J. N.; Silva, C. S., An XML knowledge base system for scheduling problems, In: *Proceedings of the Innovative Internet Computing System Conference, I2CS'02*, Kuhlungsborn, Germany. Springer-Verlag in the Lecture Notes in Computer Science series, 2002; 61-70.
7. Varela, M. L. R.; Aparicio, J. N.; Silva, S. C., Developing a Web Scheduling System Based on XML Modeling, In: *Knowledge and Technology Integration in Product and Services – Balancing Knowledge and Technology in Product and Service Life Cycle, BASYS'02*, Cancun, Mexico. Kluwer Academic Publishers, 2002, pp. 61-70.
8. Varela, M. L. R.; Aparicio, J. N.; Silva, S. C., A Scheduling Web Service based on XML-RPC, In: *Proceedings of the 1<sup>st</sup> Multidisciplinary International Conference on Scheduling: Theory and Applications, MISTA'03*, Nottingham, UK. ASAP, The University of Nottingham. 2003, p.g. 540-551.
9. Wu, J.; *Distributed System Design*. New York: CRC Press, 1999.