A Scheduling Web Service based on XML-RPC

Maria Leonilde R. Varela*, Joaquim N. Aparício†, and Sílvio do Carmo Silva+

*Dept. of Production and Systems, University of Minho, Campus Azurém, 4800-058 Guimarães, Portugal, phone: +351 253 510341, email: leonilde@dps.uminho.pt

†Dept. of Computer Science, New University of Lisbon, Quinta da Torre, 2829-516 Monte de Caparica, Portugal, phone: +351 21 2948536, email: jna@di.fct.unl.pt

+Dept. of Production and Systems, University of Minho, Campus Gualtar, 4710-057 Braga, Portugal, phone: +351 253 604745, email: scarmo@dps.uminho.pt

Abstract

In this paper we make a XML-based modeling and communication contribution to production scheduling activity. Scheduling problems and resolution methods are modeled in XML[1]. This information modeling is the basis of a web service oriented to scheduling problem solving, which is under development. The resolution of scheduling problems is carried out with the aid of a web system that uses the XML-RPC protocol for the execution of methods, local or remotely available through the Internet.

Keywords: Production scheduling, XML Modeling, Web Service and XML-RPC protocol.

Introduction

The scheduling activity in a production environment seeks to optimize the use of available production means or resources, ensuring short time to complete jobs and to satisfy other organizational optimizing criteria.
Production Scheduling may be defined as the activity, of allocating tasks or jobs to production resources, or vice versa, over time, for achieving good operating performance. The resulting schedule or scheduling plan can be more or less detailed, in accordance with the intended objectives and the planning time horizon. Thus, there are cases where we are only interested in obtaining the sequence in which the jobs should be processed in certain machines of a production system, and other cases where we are interested in knowing the planned start and finishing times of each job operation on each machine.

The effective and efficient resolution of scheduling problems begins with the identification of suitable scheduling methods to solve them. Sometimes we may encounter methods that find optimum solutions. Frequently, however, for real world problems, this is not the case, due to the problems' complexity. So we might have to draw upon available methods, which are likely to find good solutions but not necessarily optimum ones.
When there are alternative methods to solve a problem we can obtain alternative solutions, which should be evaluated against specified criteria or objectives to be reached. Thus, we are able to solve a problem, through the execution of one or more scheduling methods and, subsequently, select the best solution provided by them. These methods can either be local or remotely accessible through the Internet.

In our work we seek to improve the resolution process for scheduling problems by means of a web system. This system requires the specification of each problem to be considered and the access to resolution methods, which are available for solving them. For problems specification we propose a problem classification framework. Based on this, the XML language is used as a specification language for scheduling data modeling and processing on the Internet. This kind of data modeling allows, for instance, identifying scheduling problems, and methods for its resolution, as well as the communication necessary for the execution of implemented scheduling methods through the web.

This paper is organized as follows. The next section describes the nature of scheduling problems and the classification model proposed. References to some well known scheduling problem classes and corresponding resolution methods are shown. This section also presents a brief summary of the main search techniques used by scheduling methods. Next, the XML-based specification for scheduling concepts modeling is briefly described Subsequently, the XML-RPC, an XML-based Remote Procedure Call protocol, is presented and its use is exemplified through an example for the invocation of a scheduling method, for solving a certain problem instance. Finally, some concluding remarks are presented and planned future work is summarized.

Scheduling Problems and Resolution Methods

Scheduling problems belong to a much broader class of combinatorial optimization problems, which, in many cases, are hard to solve, i. e. are NP-hard problems (Ceponkus 1999, Jordan 1996, Blazewicz 1996, Brucker 1995). In presence of a NP-hard problem we may try to relax some constraints imposed on the original problem and then solve the relaxed problem. The solution of the latter may be a good approximation to the solution of the original one. Many times we do not have a choice and have to draw upon what we may, generally, call heuristic methods. These include both, those, which we know how near their solutions may be from optimum ones, known

[1] eXtended Markup Language.

as approximation methods, and also a variety of heuristic methods, which are likely to achieve good solutions (French 1982).

In order to execute the scheduling process it is necessary to clearly specify the problem to be solved. 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 classification and representation. A framework for achieving this was developed by Varela (1999), summarized in Table 1, based on published work by Conway (1967), Graham et al (1979), Brucker (1995), Blazewicz (1996), and Jordan (1996), as well as on other information presented by (Morton 1993) and by other authors namely (Artiba 1997) and (Pinedo 1995). This framework allows identifying the underlying characteristics of each problem to be solved, and is used as a basis for the XML-based problem specification model put forward with this work.

The referred framework for problem representation includes three classes of notation parameters, each one representing a set of problem characteristics, Table 1. The first class of characteristics, the α class, is related with the environment where the production is carried out. It specifies the production system type and the number of machines that exist in that system. Another class allows specifying the interrelated characteristics and constraints of jobs and production resources, which is denoted by the class β ($\beta_1 \ldots \beta_{12}$) of parameters. 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 characteristics. For the optimization criterion, the third and last class of the framework, we use the parameter γ. This allows specifying the schedule evaluation or performance measure, which can be either a single criterion measure or a multi-criteria one.

An example of use of this notation is "F,3|n|Fmax", which reads as: "Scheduling of non-preemptable and independent tasks of arbitrary processing time lengths, arriving to the system at time 0, on a pure flow shop, with 3 machines, to minimize the maximum flow time.

Good schedules strongly contribute to business success. This may mean meeting due dates, achieving short delivery times for 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 available for solving each particular problem.

Class	Factor	Description	Value
α	α1	Manufacturing system type	O, *P*, *Q*, *R*, *X*, *O*, *J*, *F*, PMPM, ...
	α2	Number of machines	O, *k*
β	β1	Job/ operation preemption	O, pmtn
	β2	Precedence constraints	prec, chain, tree, sp-graph, …
	β3	Ready times	O, r_j
	β4	Restrictions on processing times	p_j=1, p_{ji}=1, p_j=p, $p_{inf} \leq p_j \leq p_{sup}$, …
	β5	Due dates (deadlines)	O, d_j
	β6	Batches/ families processing	O, batch
	β7	Number of jobs or of tasks in a job (job shop case)	O, n_j
	β8	Job/ task priorities	O, w_j
	β9	Dynamic machine availability	O, avail
	β10	Additional/ auxiliary resources	O, aux
	β11	Buffers	O, no-wait
	β12	Setup (changeover)	O, setup*
γ	γ	Performance measure	C_{max}, F_{max}, $\sum C_j$, $\sum w_j C_j$, L_{max}, $\sum T_j$, …

Table 1 – Scheduling problems characteristics.

Resolution Methods

It is rather obvious that the time we often can devote for solving particular scheduling problems is usually short. Therefore, only low order polynomial time approaches are likely to be acceptable to solve real world problems, usually complex. Thus, the examination of the complexity of those problems should be the basis for further analysis to problem solving. Fortunately, not all NP-hard problems are equally hard from a practical perspective. Some NP-hard problems can be solved pseudo-polynomially using approximation methods that provide feasible solutions, which, although normally sub-optimum, are of good quality. Examples of this kind of methods are based on dynamic programming or branch and bound techniques. Other approaches to obtain good or at least satisfactory solutions, in acceptable time, are based on the nowadays widely used meta-heuristics, based on local or neighborhood search techniques, such as Genetic Algorithms (GA), Simulated Annealing (SA) and Tabu Search (TS). These are also known as extended neighborhood search techniques. We can still mention promising scheduling approaches based on bottleneck approaches, neural networks, Petri-nets and computer simulation.

Heuristic methods tend to provide good results in the available time to make decisions, reason why it is important to incorporate them in the web scheduling system here presented, as we are doing.

Table 2 shows a small sample of makespan optimization flow shop scheduling problems, for which methods are referenced, collected from Brucker (1995) and which may appear in real world production systems.

Problem class	Method reference	Observations
F2 | | Cmax	Johnson (1954)	Maximal polynomially solvable Without preemption
F2 | rj | Cmax	Lenstra et al (1977)	Minimal NP-hard Without preemption
F2 | rj; no-wait | Cmax	Roeck (1984)	Maximal polynomially solvable With no wait
F3 | pmtn | Cmax	Gonzalez & Sahni (1978) Cho & Sahni (1981)	Maximal polynomially solvable With preemption
F3 | | Cmax	Garey et al (1976)	Minimal NP-hard Without preemption
F | pji=1; prec | Cmax	Leung et al (1984), Timkovsky (1998)	Minimal NP-hard Without preemption
FMPT | n=3 | Cmax	Kraemer (1995)	Minimal NP-hard With multiprocessor task
FMPT, m | rj; pji=1|Cmax	Brucker & Kraemer (1996)	Maximal polynomially solvable With multiprocessor task
FMPM, m | rj;pji=1|Cmax	Brucker et al (1997)	Maximal polynomially solvable With multipurpose machines
FMPM | prec; pji=1| Cmax	Ullman (1975)	Minimal NP-hard With multipurpose machines

Table 2 – Scheduling methods assigned to problems.

Information like this is used by and available for retrieval through the web scheduling system. The system is able to quickly suggest methods for solving problems that occur in real world manufacturing environments and solve them through the execution of an appropriate method implementation, local or remotely available through the Internet. This draws upon the web system knowledge base for scheduling problem and methods.

Scheduling Concepts Modeling using XML

Since 1995 great happenings have changed the world of information technology, especially the emergence of new Internet technologies. The eXtensible Markup Language (XML) is one of those new technologies that has been having a wide acceptance and is causing a great impact on Internet real world applications, since its release by the World Wide Web Consortium (W3C) in 1998 (Pardi 1999). XML enables to describe structures and meanings of data, with a simple syntax, and is an ideal candidate format for exchanging and processing data through the Internet. Other advantages of XML based representation are its openness, simplicity and scalability (Abiteboul et al 2000). These were important reasons for choosing XML to develop our web application. For details about XML and related technologies (DTD, XSL, XML Schemas, Namespaces, etc.) see, for example, Abiteboul (2000) or Ceponkus (1999).

The web applications can use XML for data storage and processing, for showing multiple views of the data and for representing complex data structures. Therefore, XML may guarantee the future utilization of data formats and the exchange of data structures, so that the web documents and the platforms become more robust for systems integration (Pardi 1999).

Some interesting XML applications, which are more or less related with this work, are PDML (Product Data Markup Language), RDF (Resource Description Format) and STEPml (Pardi 1999). Other XML specifications devoted to manufacturing processes are JDF (Job Definition Format), PSL (Process Specification Language), PIX-ML (Product Information Exchange), PIF (Process Interchange Format) and XML-based workflow (Abiteboul et al 2000).
There are many other web-based technologies available for data storage and transferring, but we think that it is more adequate and easier to develop a new system using these new techniques rather than using conventional ones, such as EDI (Electronic Data Interchange). XML based data exchange is becoming very popular in global manufacturing, and this will cause connectivity becoming more and more convenient and necessary.

Following the lines already presented in (Varela 1999, Varela 2002a, Varela 2002b), problems are classified and modeled by a DTD - Document Type Definition. Elements introduced in the referred DTD are expected to become part of a common namespace. Elements on the problem DTD file precisely characterize a scheduling problem, meaning that in order to interact with the system a problem must be described according to that grammar.

In order to match problem instances to resolution methods we must be able to identify those problems and retrieve appropriate methods for its resolution. Therefore, the scheduling methods and its implementations are also described by a given DTD. Many scheduling methods may be more or less adequate to solve a given class of problems so, in the methods knowledge base; the system records the scheduling method(s) that can be used for solving a certain problem class.
The methods' implementations description must include, among other things, the address to the running method or program and its signature, which, in turn, includes the definition of the parameters that are necessary for its execution (input) and its output format.
The result from running a method implementation on the given problem instance can then be delivered to the client as an XML file and/or can be transformed into some more expressive output, like a Gantt chart.
The web system under development is being implemented as a web service using the XML-RPC protocol (Laurent et al. 2001), (http://www.w3.org, http://www.xmlrpc.com).

Methods Invocation through XML-RPC

As mentioned above the main purpose of this work is to provide a framework to improve the resolution of scheduling problems based on XML modeling and related technologies.
The main element of the web system structure is an interface located in the Centralized Problems/Methods Knowledge Base (CPM-KB) unit, for introduction, validation, and transformation of manufacturing

scheduling data. This interface is mainly controlled by DTD and XSL (eXtended Stylesheet Language) documents stored in a database. The scheduling information is also stored in XML documents and these documents are verified using DTDs, before being inserted in the XML database. The XML and related documents may either be located on the server or on the client side. In this work the documents are stored on the server (e.g. XML, DTD, XSL and other documents) in order to achieve easy and efficient data transferring.

Figure 1 illustrates a general outline of the system architecture.

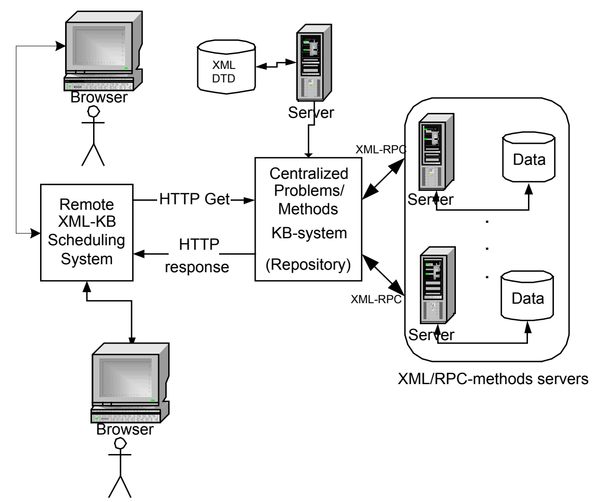

Figure 1 – Web-system architecture.

The term Web Services has emerged as a general category for loosely coupled, dynamically connected web-based services and are a set of tools that let us build distributed applications on top of existing web infrastructures.

These services use XML to encode both the message wrapper and the content of the message body. As a result, the integration is completely independent of operating system, language or other middleware product used by each component participating in the service. The only fundamental requirement is that each component has the ability to process XML documents and that each node connected in a distributed system supports HTTP as a default transport layer.

These Web Services are most commonly used to invoke remote application services or methods using a Remote Procedure Call (RPC) interaction implemented using only XML messages (Laurent 2001).

The XML-RPC protocol is the sequence and structure of requests and responses required to invoke communications on a remote machine. Several other protocols that could also be used exist, namely SOAP (Simple Object Access Protocol), UDDI (Universal Description, Discovery, and Integration of business for the web), WSDL (Web Services Description Language), or other well known, like CORBA, RMI or DCOM. Nevertheless, XML-RPC is among the simplest and most foolproof web service approaches, and makes it easy for computers to call procedures on other computers (Laurent et al. 2001). The XML provides a vocabulary for describing remote procedure calls, which are then transmitted between computers using the Hyper Text Transfer Protocol (HTTP). XML-RPC clients make procedure requests to XML-RPC servers, which return results to the XML-RPC clients. XML-RPC clients use the same HTTP facilities as web browser clients, and XML-RPC servers use the same HTTP facilities as web servers.

XML-RPC requires a minimal number of HTTP headers to be sent along with the XML method request. Listing 1 shows an example that joins the headers and XML payload to form a complete XML-RPC request to a call to the method ExactBranchBound (Ignall, 1965), which returns a solution to an instance of the F,3|n|Fmax problem class.

```
POST /rpchandler HTTP/1.0
User-Agent: AcmeXMLRPC/1.0
Host:localhost:5001
Content-Type: text/xml
Content-Length: 832
<?xml version="1.0"?>
<methodCall>
<methodName>ExactBranchBound</methodName>
<params>
  <param><value><int>4</int></value></param>
  <param><value><int>3</int></value></param>
  <param><value>
   <array><data>
      <value><string>J1</string></value>
      <value><string>M1</string></value>
      <value><double>3</double></value>…
    </data></array>
</value></param></params></methodCall>
```

Listing 1 – A complete XML-RPC request.

Upon receiving an XML-RPC request, an XML-RPC server must deliver a response to the client. The response may take one of two forms: the result of processing the method or a fault report, indicating that something has gone wrong in handling the request from the client. As with an XML-RPC request, the response consists of HTTP headers and an XML payload.

Figure 2 schematizes our web service framework, based on this protocol.

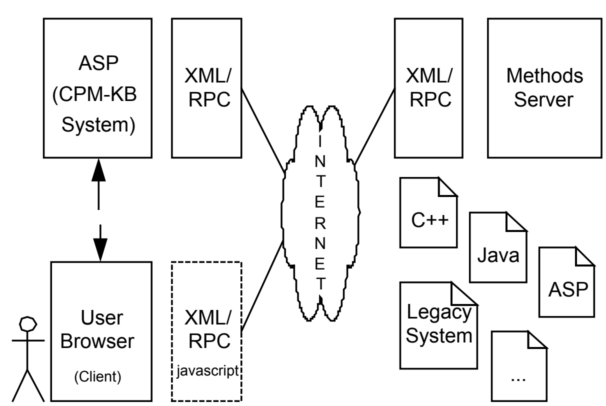

Figure 2 – Web-service based on XML-RPC.

System Functionalities

This web system encompasses several functionalities, which include knowledge insertion, about scheduling problems and resolution methods, and corresponding 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 documents, adequate for each specific visualization request. Another important functionality is the execution of scheduling methods, given the scheduling problem definition. The selection of one or more specific methods is made through a system's searching process on the knowledge base of scheduling methods (CPM-KB). The system also enables problem results presentation and storage.

Gantt charts are automatically generated by the system. This is easily achieved because the problem output data is expressed in XML documents that enable an easy way of outputs conversion into different desired output forms. This facilitates comparing solutions obtained from the execution of several method implementations. Other alternatives for displaying the same data are available.

Conclusions

In manufacturing enterprises, it is important nowadays, as a competitive strategy, to explore and use software applications, now becoming available through the Internet and Intranets, for solving scheduling problems.

This paper proposes an XML-based specification framework for production scheduling concepts modeling, together with a web-based production scheduling system. Some of the important functions include the ability to represent scheduling problems and the identification of appropriate methods to solve them.

In order to make possible flexible communication among different scheduling applications XML-based data modeling is used. The XML-based scheduling data specification contributes to the improvement of the scheduling processes by allowing an easy selection of several alternative methods available for problem solving, as well as an easy maintenance of the knowledge base itself. This primarily includes both scheduling problems and solution methods, which are available through the Internet. The framework enables the registration of new problem classes, resolution methods and implementations, as well as post classification and matching among them.

The specification format is adequate for the exchange of scheduling data, since it enables to handle with loosely coupled systems and with complex hierarchical data.

The XML based specification can be generated and visualized by computers in appropriate and different ways. An important issue is that the data representation model is general, accommodating a large variety of production scheduling problems, which may occur in different types of manufacturing environments.

Furthermore, the web scheduling system under development facilitates the resolution of scheduling problems, through the execution of local or remote scheduling methods, available on different computers through the Internet, in order to greatly contribute to assist de scheduling decision-making process, by allowing different solutions comparison, obtained by the execution of different methods for a same problem and to choose the solution, which shows more suitable to solve each particular problem that occurs in the identified manufacturing environment under consideration.

References

Artiba, A., Elmaghraby, S. eds. 1997. The Planning and Scheduling of Production Systems, UK: Chapman & Hall.

Abiteboul, S., et al. eds. 2000. Data on the Web - From Relations to Semistructured Data and XML, USA: Morgan Kaufmann Publishers.

Blazewicz, J., et al. eds. 1996. Scheduling Computer and Manufacturing Processes, Germany: Springer-Verlag.

Brucker, P. Eds. 1995. Scheduling Algorithms, Germany: Springer-Verlag.

Ceponkus, A., Hoodbhoy, F. eds. 1999. Applied XML, USA: Wiley Computer Publishing.

Conway, R. W., Maxwell, W. L., Miller, L. W. eds. 1967. Theory of Scheduling, England: Addison-Wesley Publishing Company, Inc.

French, S. eds. 1982. Sequencing and Scheduling – An Introduction to Mathematics of the Job-Shop. John Wiley and Sons, Inc.

Graham, R. L., Lawler, E. L., Lenstra, J. K., Rinnooy Kan, A. H. G. eds. 1979. Optimization and Approximation in Deterministic Sequencing and Scheduling: A survey, Annals of Discrete mathematics.

Ignall, E., Schrage L. eds 1965. Application of the Branch-and-Bound Technique to Some Flow-Shop Problems. Operations Research 13 (3).

Jordan, C. eds. 1996. Batching and Scheduling, Germany: Springer-Verlag.

Laurent, S., et al. eds. 2001. Programming Web Services with XML-RPC, O'Reilly & Associates, Inc.

Morton, T., Pentico, D. eds. 1993. Heuristic Scheduling Systems, USA: John Wiley & Sons Inc.

Pardi, W. eds. 1999. XML - Enabling Next-generation Web Applications, USA: Microsoft Press.

Pinedo, M. eds. 1995. Scheduling Theory, Algorithms and Systems, USA: Prentice-Hall Inc.

Varela, L., Aparício, J., Silva, S. 2002, An XML Knowledge Base System for Scheduling Problems. In Proceedings of the Innovative Internet Computing System Conference, 61-70. Kuhlungsborn, Germany: Springer-Verlag in the Lecture Notes in Computer Science series.

Varela, L., Aparício, J., Silva, S. 2002, Scheduling Problems Modeling with XML, In Proceedings of the 4th International Meeting for Research in Logistics, 897-909. Lisbon, Portugal.

Varela, M. L. 1999. Automatic Scheduling Algorithms Selection, Msc. Diss., Dept. of Production and Systems, University of Minho, Portugal.