

AN ONTOLOGY FOR A MODEL OF MANUFACTURING SCHEDULING PROBLEMS TO BE SOLVED ON THE WEB

Maria Leonilde Rocha Varela⁽¹⁾, Sílvio do Carmo Silva⁽²⁾
School of Engineering, University of Minho, Portugal
leonilde@dps.uminho.pt⁽¹⁾, scarmo@dps.uminho.pt⁽²⁾

In this paper we describe an ontology for a model of manufacturing scheduling problems to be solved on the web, through several different scheduling methods made available by a web scheduling decision support system.

This approach aims enabling to share scheduling knowledge and methods, in a globally distributed context, in order to facilitate solving these problems, either in the industry or in the academic and research context.

The proposed ontology is described in terms of basic objects, the scheduling problem attributes, in the form of problems characterization parameters; problem classes or types of scheduling problems and relations among problem classes; and problems instances.

Finally, a sample of methods is presented for solving some well known problems arising in the scope of manufacturing scheduling through the web.

1. INTRODUCTION

The scheduling activity in an organization seeks to optimize the use of available production means or resources, ensuring short time to complete jobs and, in addition, to satisfy other important organization objectives. Thus, it can highly contribute to good service to customers and to high profitability of an organization.

Manufacturing scheduling may be defined as the activity of allocating tasks to production resources, or vice versa, over time. The result of this is usually expressed in a production schedule.

With this work we make a contribution for the better resolution process of scheduling problems by means of a decision support system on the web, based on a proposed ontology for a model of manufacturing scheduling problems description. Therefore, the system requires, first of all, the specification and identification of each problem to be solved and, then, the access to resolution methods, which are available for solving them. When there are different methods available we can obtain alternative solutions, which should be evaluated against specified criteria or objectives to be reached. Thus, we are able to properly solve a problem or closely related problems, 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 web.

The main objective of this paper consists on presenting the proposed ontology for a model of manufacturing scheduling problems, which consists on a classification framework for scheduling problems. This classification framework is represented through

XML (eXtended Markup Language) on the web scheduling decision support system due to some important advantages, for instance allowing to specify scheduling problems and easily identifying methods for their resolution and to establish the necessary communication for the execution of the implemented scheduling methods through the web by using a web service (Papazoglou *et al.*, 2003), (Varela *et al.*, 2002a,b; 2003).

This paper is organized as follows. In chapter 2 describes the proposed ontology for a model of manufacturing scheduling problems. Therefore, in section 2.1 we present a brief definition of manufacturing scheduling problems. In section 2.2 the scheduling problem attributes are presented through the main underlying characterization parameters. Section 2.3 presents and describes some well known scheduling problem classes and relations among them and in section 2.4 refers to scheduling problem instances, where an example is given. In section 3 we briefly present the web-based scheduling decision support with a list of available methods for solving some closely related and well known manufacturing scheduling problems. Finally, in section 4 we present some conclusions.

2. SCHEDULING PROBLEMS ONTOLOGY

2.1. A brief definition of manufacturing scheduling problems

The manufacturing scheduling activity may be defined as the activity of allocating production resources to tasks or vice versa, during a certain time period. Therefore, a proper production schedule enables a company to make good use of available resources and efficiently achieving operational objectives (Conway *et al.*, 1967).

The process of scheduling problem solving is essentially concerned with finding the sequence in which jobs should be processed. Sometimes, however, we may need also to know the scheduled start and finishing times of every job operation on each machine. This information completely defines a schedule or scheduling plan.

Good schedules strongly contribute to increase companies' success. Among other ways, this is achieved through deadlines satisfaction for the accepted orders, low flow times, few ongoing jobs in the system, high resource utilization and low production costs. These objectives can be better satisfied through the execution of the most suitable scheduling methods available for solving each particular problem under consideration and for doing that we need to clearly specify the problem to be solved.

2.2. Characterization parameters

Scheduling problems have a set of characteristics that need specification. Some important processing requirements that frequently have to be taken into account for processing jobs have to do with resources other than machines, i.e. operators, tools, handling devices buffers and others.

This parameters can be expressed through a " $\alpha | \beta | \gamma$ " nomenclature. Our proposed nomenclature is based on some well known nomenclatures described in the scheduling domain (Varela *et al.*, 2002a) and consists on an expanded form that can be represented by " $\alpha 1, \alpha 2 | \beta 1, \dots, \beta 18 | \gamma$ ".

The first class of factors, which we call the α class, characterizes the production environment, i.e. the system and machines available. Another, the β class, deals mainly with characterization of jobs, resources and processing requirements.

Finally, the third γ class specifies the performance measure or evaluation criterion.

All the classification parameters are summarized in Tables 1, 2 and 3 below.

Table 1. Manufacturing environment characterization parameters

Class	Parameter	Designation	Value	Default value
α	α_1	Production systems types: Geral flexible system with multi-processor jobs/ operations, GFM: Single-stage GFM (GFM/fl). Single-stage GFM with parallel processors (GFM/fl/P), with: identical parallel processors (GFM/fl/PI); uniform parallel processors (GFM/fl/PU) or unrelated parallel processors (GFM/fl/PN). Multiple-stage GFM (GFM/fm): Jobshop or pure jobshop GFM (GFM/fm/J or JP) Flowshop or pure flowshop (GFM/fm/F or FP) Openshop or pure openshop (GFM/fm/O or OP). General flexible system, GF: Single-stage GF: Single-stage single processor GF (GF/fl) or Single-stage parallel processors GF (GF/fl/P): Identical parallel processors GF (GF/fl/PI) Uniform parallel processors GF (GF/fl/PU) Unrelated parallel processors GF (GF/fl/PN). Multiple-stage GF: Jobshop or pure jobshop (GF/fm/J or JP) Flowshop or pure flowshop (GF/fm/F or FP) Openshop or pure openshop (GF/fm/O or OP). General system with multi-processor jobs, GM: Single-stage GM: Single-stage single processor (GM/fl). Single-stage parallel processors GM (GM/fl/P): Identical parallel processors GM (GM/fl/PI) Uniform parallel processors GM (GM/fl/PU) Unrelated parallel processors GM (GM/fl/PN). Multiple-stage GM: Jobshop or pure jobshop (GM/fm/J or JP) Flowshop or pure flowshop (GM/fm/F or FP) Openshop or pure openshop (GM/fm/O or OP). General system, G: Single-stage single processor G (G/fl) Single-stage parallel processors G (G/fl/P). Identical parallel processors G (G/fl/PI) Uniform parallel processors G (G/fl/PU) Unrelated parallel processors G (G/fl/PN). Multiple-stage G: Jobshop or pure jobshop (G/fm/J or JP) Flowshop or pure flowshop (G/fm/F or FP) Openshop or pure openshop (G/fm/O or OP).	GFM, GFM/fl, GFM/fm, GFM/fl/P, (PI, PU, PN GFM/fm/J, GFM/fm/JP, GFM/fm/F, GFM/fm/FP, GFM/fm/O, GFM/fm/OP. GF, GF/fl, GF/fm, GF/fl/P, (PI, PU, PN) GF/fm/J, GF/fm/JP, GF/fm/F, GF/fm/FP, GF/fm/O, GF/fm/OP. GM, GM/fl, GM/fm, GM/fl/P, (PI, PU, PN) GM/fm/J, GM/fm/JP, GM/fm/F, GM/fm/FP, GM/fm/O, GM/fm/OP. G, G/fl, G/fm, G/fl/P, (PI, PU, PN) G/fm/J, G/fm/JP, G/fm/F, G/fm/FP, G/fm/O, G/fm/OP.	1
	α_2	Number of processors.	\emptyset , variable, m	\emptyset

Regarding the α class of parameters, the main manufacturing environment characterization factors are expressed by “G”, representing general shops, where jobs may have no precedence relations among its operations. “F” stands for flexible manufacturing environments, meaning that the system may have two or more alternative processors for executing a given operation on some job. When jobs may require two or more processors simultaneously this is expressed by the multi-processor factor “M”. A single stage manufacturing system, where each job includes only one processor, i.e., $n_j=1 \quad \forall j: j=1, \dots, n$, is represented by factor “fl” and a multi-fase manufacturing system, where each job may have two or more operations: $n_j \geq 2 \quad \forall j: i=1, \dots, n$, is expressed by “fm”.

A single stage manufacturing system may include two or more parallel processors, and this is represented, generally, by “P”. If the parallel processors are unrelated processors then the corresponding characterization factor is “PN”, otherwise, in case of uniform processors this is represented by “PU” and identical processors are expressed by “PI”.

There are additional factors corresponding to well known manufacturing systems like jobshops (“J”); pure jobshops (“JP”); flowshops (“F”); pure flowshops (“FP”); openshops (“O”) and pure openshops (“OP”), which may or may not appear combined with the previously introduced factors “G”, “F” and “M”, forming a variety of manufacturing environments that may occur in several different manufacturing scenarios arising in industry or in academic context.

Table 2. Jobs characterization parameters

Class	Parameter	Designation	Value	Default value
β	β_1	Jobs preemptions	\emptyset , pmtn, free-pmtn, comp-pmtn	\emptyset
	β_2	Jobs precedences	\emptyset , prec, tree:intree, outtree, chain, sp-graph	\emptyset
	β_3	Jobs ready-times	\emptyset , r_j , r_{ij} , r_{jl} , r_{ijl}	\emptyset
	β_4	Jobs processing times restrictions	\emptyset , $t_j=1$, $t_{ij}=1$, $t_j=p$, $t_{ij}=p$, $t_{i(j)} \in \{a,b\}$, $t_{\min} \leq t_{i(j)} \leq t_{\max}, \dots$	\emptyset
	β_5	Jobs due dates	\emptyset , d_i , d_j , d_{ij} , D	\emptyset
	β_6	Batch processing	\emptyset , batch	\emptyset
	β_7	Families processing	\emptyset , fam	\emptyset
	β_8	Complex jogs processing	\emptyset , compj	\emptyset
	β_9	Jobs quantity	n , n_j , n_l	n
	β_{10}	Jobs priorities	\emptyset , w_j , w_{ji} , w_{jl}	\emptyset
	β_{11}	Multi-processor tasks	\emptyset , mpt_j , mpt_{ij} , mpt_{jl}	\emptyset

Table 3. Machines and auxiliary resources characterization parameters

Class	Parameter	Designation	Value	Default value
β	β_{12}	Machines eligibility	\emptyset , M_k	\emptyset
	β_{13}	Machines availability	\emptyset , $avail_k$	\emptyset
	β_{14}	Auxiliary resources	\emptyset , aux_k	\emptyset
	β_{15}	Critical resources	\emptyset , crt_k	\emptyset
	β_{16}	Machines setup	\emptyset , s_{jk} , s_{ijk} , s_{lk} , s_k	\emptyset
	β_{17}	Intermediate buffers	\emptyset , $buffer_k$, no-wait	\emptyset
	β_{18}	Multi-purpose machines	\emptyset , mpm_k	\emptyset

Regarding the performance measurement parameters, we think that objectives can be put together into one parameter (γ) and the proposed manufacturing classification nomenclature allows users to add their own domain specific performance measures to be considered on each particular case.

Typical examples of such measures are the maximum flow time (Fmax), the makespan (Cmax) and the mean and maximum lateness of jobs (Lmean, Lmax), among many others, namely: $\gamma \in \{Fmax, Cmax, \Sigma Cj, \Sigma (wjCj), Lmean, Lmax, \Sigma Tj, \Sigma (wjTj), \Sigma Ej, \Sigma (wjEj), \Sigma NTj, \Sigma (wjNTj), \dots\}$.

2.3. Problem classes and relations among them

The problem characteristics previously described in section 2.2 may be combined in different ways, resulting in many distinct scheduling problem classes. Table 4 presents a sample of problem classes and references about some well known methods for solving them.

Table 4. Sample of problem classes and solving methods

Problem class	Method reference	Observations
G/fm/F,2 n Cmax	Johnson (1954)	Maximal polynomially solvable Without preemption
G/fm/F,2 n, rj Cmax	Lenstra et al (1977)	Minimal NP-hard Without preemption
G/fm/F,2 n, rj; no-wait Cmax	Roeck (1984)	Maximal polynomially solvable With no wait
G/fm/F,3 n, pmtn Cmax	Gonzalez & Sahni (1978) Cho & Sahni (1981)	Maximal polynomially solvable With preemption
G/fm/F,m n, pji=1; prec Cmax	Leung et al (1984) Timkovsky (1998)	Minimal NP-hard Without preemption
GM/F n=3 Cmax	Kraemer (1995)	Minimal NP-hard With multiprocessor task
GF/fm/F,3 n Cmax	Garey et al (1976)	Minimal NP-hard With multipurpose machines
GF/fm/F,m rj; tji=1 Cmax	Brucker et al (1997)	Maximal polynomially solvable With multipurpose machines

For example of use of this notation is “G/fm/F,2|n|Cmax” which reads as: “Scheduling of non-preemptable and independent tasks of arbitrary processing time lengths, arriving to the system at time zero, on a general flowshop, with 2 machines, in order to minimize the maximum flow time or makespan.

Regarding Table 4 we can realize that the problems are related in terms of problem classes and subclasses. For example, class G/fm/F,2|n|Cmax previously described consists on a subclass of the general class G/fm/F,m|n|Cmax, since this last one refers to a flowshop including any number of machines besides the first one refers only to a flowshop integrating only two machines.

The proposed specification nomenclature provides a general framework of scheduling problems and related concepts, and consequently a way of describing each particular problem enabling to easily associate each problem to appropriate solving methods and specifying each problem instance to be solved. Problems are associated to methods by matching problem classification parameters, according to the proposed nomenclature, to scheduling methods inputs and additional information about given problem instances has to be specified for solving them when using the developed scheduling decision-support system (Varela *et al.*, 2002b, 2003).

2.4. Problem instances

One of the most critical success factors for implementing shop floor scheduling systems is the possibility of dealing with various constraints on each kind of production process (Wu, 1999).

In defining job release orders it is necessary to express a hierarchy of elements and attributes about the problems, such as item, quantity, location or destination, due date, processing time, and so on, as previously represented on section 2.2.

Therefore, in manufacturing scheduling problems a job represents an action that has certain time duration. During that time, the job changes status of inventories of corresponding items, occupying or loading some particular resources. The jobs need some resources and produce some outputs. Production resource means workstations, machines, equipments, tools, labors, and so on.

An example of a scheduling problem instance consists on processing a set of ten non-preemptable and independent jobs, with unit processing time lengths, arriving at time zero, on a general flowshop, with 5 machines, in order to minimize total completion time of all jobs or makespan (G/fm/F,5|10,p=1|Cmax).

3. SOLVING SCHEDULING PROBLEMS ON THE WEB

The manufacturing scheduling problems can be solved through a web decision support system developed based on the proposed ontology for specifying these problems, previously presented.

Figure 1 shows the automatically generated interface, based on XML and related technology for specifying scheduling knowledge about problems and solving methods (Varela *et al.*, 2002a,b; 2003). This figure shows a list of related scheduling problem classes that include characteristics specified according to the proposed problem classification nomenclature. From this list the user may select one or more of those closely related problem classes. This multiple choice is important when the user does not know exactly which class of problems he/ she wants to solve or even to enlarge the set of possible solving methods to be found for solving a problem.

Problem types for the general problem characteristics: [system_type=F, jobs=n, measure=Cmax]				
Class	Complexity	Context	Problem characteristics	Select
F2 n Cmax	Maximal Polynomially Solvable	≥	[(system_type, F), (machines, 2), (jobs, n), (measure, Cmax)]	<input checked="" type="checkbox"/>
F2 j,r Cmax	Minimal NP-hard	≥	[(system_type, F), (jobs, n), (arrivals, rj), (machines, 2), (measure, Cmax)]	<input type="checkbox"/>
F2 j,r,n,no-wait	Maximal Polynomially Solvable	≥	[(system_type, F), (machines, 2), (jobs, n), (arrivals, rj), (buffers, no-wait), (measure, Cmax)]	<input type="checkbox"/>
F3 j,r Cmax	Minimal NP-hard	≥	[(system_type, F), (machines, 3), (jobs, n), (arrivals, rj), (measure, Cmax)]	<input type="checkbox"/>
F3 n Cmax	Minimal NP-hard	≥	[(system_type, F), (machines, 3), (jobs, n), (measure, Cmax)]	<input checked="" type="checkbox"/>
Fm n Cmax	Minimal NP-hard	≥	[(system_type, F), (machines, m), (jobs, n), (measure, Cmax)]	<input checked="" type="checkbox"/>
Fm prec,pj=1	Minimal NP-hard	≥	[(system_type, F), (machines, m), (jobs, n), (precedences, prec), (times, pj=1), (measure, Cmax)]	<input type="checkbox"/>

Figure 1. List of closely related problem classes

Once the user specifies the problem classes he/ she wants to consider another automatically generated interface of the developed scheduling web system presents a list of suitable methods for solving them, as presented in Figure 2.

Available methods for solving the problem classes: [class=F2 n Cmax, class=F3 n Cmax, class=Fm n Cmax]							
ID	Prob. Class	Method name	Implementation location	Reference	Complexity	Protocol	Select
1001	Fm n Cmax	BranchBoundIS	http://localhost:6002/RPC2	Ignall and Schrage, 1965	NP-hard	XML-RPC	⊕
32	F2 n Cmax	Johnson	http://localhost:6002/RPC2	Johnson, 1954	Polynomial	XML-RPC	⊖

Figure 2. List of methods for solving scheduling problems

The user may select one of the given methods for solving a specific problem instance, for example, the branch-and-bound method proposed by Ignall and Schrage (Conway *et al.*, 1967), which is a NP-hard method for the general G/fm/F,m|n|Cmax (or simply, Fm|n|Cmax) class of scheduling problems and which was implemented in the developed web scheduling system through a web service based on the XML-RPC communication protocol (Terziyan and Zharko), (Varela *et al.*, 2003), (Wu, 1999).

4. CONCLUSIONS

In production 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 ontology for solving manufacturing scheduling problems using a web based decision-support system developed based on a proposed ontology for expressing manufacturing scheduling problems.

The main importance of the proposed ontology consists on enabling to properly represent scheduling problems in order to seek for appropriate available methods for solving them.

As we referred previously it becomes necessary to use the proposed ontology and underlying nomenclature for clearly specifying the main scheduling problem characteristics to consider when we are interested in solving a given problem instance.

This nomenclature was developed based on some important ones put forward by several different authors in the scheduling domain and was extended in order to include some additional problems characteristics, which are essential for properly specifying real problem arising from diverse industrial and academic scenarios.

Besides the proposed nomenclature may appear very extended, in practice a problem classified according to this framework rarely becomes very complex, because a problem class consists on a kind of model which only has to consider the main problem characteristics to be representative for properly solving a given problem.

5. REFERENCES

1. Conway, R. W.; Maxwell, W. L.; Miller, L. W. *Theory of Scheduling*. England: Addison-Wesley Publishing Company, Inc., 1967.
2. Papazoglou, M.P.; Krämer, B.J.; Yang, J. "Leveraging Web-Services and Peer-to-Peer Networks". In *Proceedings of Advanced Information Systems Engineering, 15th International Conference, CaiSE, Klagenfurt, Austria, June 16-18, 2003*, pp. 485-501.
3. 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>).
4. 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, 2002a*; 61-70.
5. 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, 2002b*, pp. 61-70.
6. Varela, M. L. R.; Aparicio, J. N.; Silva, S. C. "A Scheduling Web Service based on XML-RPC". In *Proceedings of the 1st Multidisciplinary International Conference on Scheduling: Theory and Applications, MISTA'03, Nottingham, UK. ASAP, The University of Nottingham. 2003*, pg. 540-551.
7. Wu, J. *Distributed System Design*. New York: CRC Press, 1999.