DOI: 10.2478/v10051-008-0024-4

Implementation of the Scheduling Domain Description Model

Alenka Baggia, Robert Leskovar, Miroljub Kljajić

University of Maribor, Faculty of Organizational Sciences, Kidričeva cesta 55a, SI-4000 Kranj, Slovenia
alenka.baggia@fov.uni-mb.si, robert.leskovar@fov.uni-mb.si, miroljub.kljajic@fov.uni-mb.si

This paper presents the problem of a uniform scheduling domain description. It was established that the algorithm used for scheduling is general, disregarding the type of scheduling domain. On the basis of five different scheduling domains, a general description model was developed. The research is focused on the programming application of the resource scheduling model, presented as a UML class diagram. Diverse meta-languages for the model description were considered. Of these XML, an EAV model and object oriented languages have shown to be the most effective. Even though Java is not widely used as a description language, it has proved effective as a meta-language for the description of the extensible scheduling model.

Key words: Scheduling, Domain description, Description Language, Object oriented analysis

1 Introduction

Scheduling problems have been widely researched in recent years (e.g. Brucker 2001; Pinedo 2002; Pinedo 2005). The theory of scheduling is characterized by a virtually unlimited number of problem types (Brucker, 2001). Different types of resources can be scheduled and for each type there are diverse solutions or scheduling algorithms. Pinedo (2005) emphasizes that each type of resource has attributes and parameters that are important for the planning and scheduling process. Since there is no common standard for specifying a scheduling problem and its solution proposed by a researcher, the algorithms cannot be compared and benchmarking is impossible. Artificial intelligence methods, for example genetic algorithms, are used in cases where criteria for schedule evaluation are well defined, but a scheduling algorithm is difficult to develop (Kljajić et. al. 2004). On the base of research into production scheduling (Papler, 2001) and human resource scheduling in a hospital (Baggia, 2004) the need arose for a generally used scheduling model.

In some cases the same scheduling solution can be used to schedule different types of resources, but little research has been done on models that cover diverse scheduling domains. Pinedo (2002) established that it may be of interest in the future to study more specific models that combine machine scheduling with personnel scheduling.

This paper presents the solution of a general scheduling problem and the development of a scheduling domain description language. Each scheduling problem consists of different basic elements and rules, presented as a UML class diagram. The solution of a scheduling problem is not within the scope of this paper. Different programming languages for implementation of the scheduling model are considered and the application of the most appropriate meta-language is presented.

2 The scheduling problem

Different resources can be scheduled depending on the type of problem. Most cases presented in previous research discuss scheduling algorithms for a defined scheduling problem. Colindres (1992) was one of the first researchers to develop a scheduling description language to be used in different areas of scheduling, but only the problems of production scheduling and project scheduling were included in the research.

Scheduling procedures in the description of production scheduling (S_1) and the description of personnel scheduling (S_2) are presented in Table 1. The differences between these procedures lie in the approach and search procedures (Baggia 2005). Procedure S_1 searches for the most appropriate machine M_i for an order N_i in a predefined time frame, while considering any constraints. The evaluation considers all constraints that need to be included when generating a schedule.

Procedure S_2 searches for the most appropriate person for a single activity in a predefined time interval. Constraints, hard or soft, are also considered in this case. When established that persons W_1, W_2 and W_3 are suitable for activity A_1, the search procedure finds a person with the highest criteria function considering the constraints. Person W_i is scheduled for activity A_i.

Table 1: Comparison of two different scheduling problem search procedures.

Production Scheduling problem S_1	**Personnel Scheduling problem S_2**

While (t_1<t_k)
While (order)
While (machine)
Evaluate order
End-while
Find machine(order)
End-while
End-while
	While (t_1<t_k)
	While (activity)
	While (person)
	Evaluate person
	End-while
	Find person(activity)
	End-while
	End-while

The result of both procedures is the same; the difference is in the sequence of steps leading to the solution, where procedure S_1 searches for the machine for a certain order while procedure S_2 searches for the person for a specific job.

The Unified Modelling Language (UML) Use Case diagram was used to describe the general scheduling process. The diagram in Figure 1 shows the general scheduling process, used as the basis for developing a scheduling description model.

If we represent the scheduling problem as a black box, with Input *x* (scheduling requirements) and Output *y* (results of scheduling), the scheduling problem S_n could be written as shown in Table 2. In different scheduling environments only the variable *v* (e.g. machine or person) is changed.

To sum up, the search procedure is general, regardless of the type of scheduling problem; the search procedures are identical:

$$S_1 \equiv S_2 \qquad (1)$$

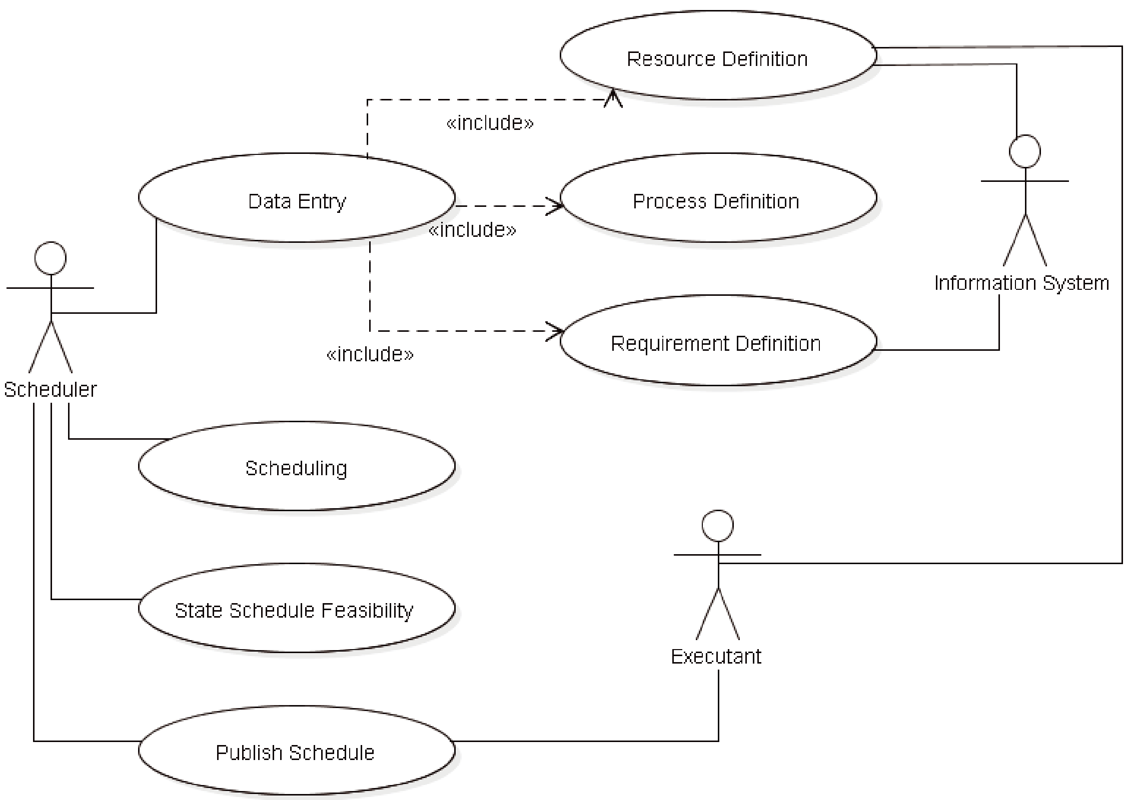

Figure 1: Resource scheduling use case diagram.

Table 2: Scheduling problem search procedure.

Scheduling problem S_n
While (t_1<t_k)
While (x)
While (v)
Evaluate
End-while
y = find v(x)
End-while
End-while

3 Elements of the scheduling problem

Equivalence exists between different scheduling procedures, as shown in Expression (1). The differences between the procedures are presented with variables for which the the algorithm finds the optimal values. The discovery of an analogy in the scheduling procedures led to the attempts to describe the scheduling problem with a general model, regardless of the type of resource that is to be scheduled (Baggia 2004).

Every scheduling domain (SD) can be described by four basic elements: object types, syntax, parameter and algorithm:

$$SD \subseteq O \times P \times S \times A \tag{3}$$

Where
- O is an object type.
- P is a parameter.
- S is a syntax.
- A is an algorithm.

"Object type" represents the components that describe an object, schema or rule. Object types can be elementary or complex. Elementary object types are usually characterized by input, output and simple transparent rules. Complex object types comprise a Cartesian product of elementary object types:

$$o_i \subseteq o_1 \times o_2 \times \ldots \times o_n \tag{4}$$

The syntax describes the relationship rules between individual object types. Different types of relations between object types describe the behaviour of the scheduling system. Each object type is described with parameters or attributes. Parameters define the specifics of an individual scheduling problem. Sometimes, relationships between object types also need to be specified in detail. Therefore, parameters can also be used in the syntax. Object types with their parameters and defined syntax represent the input for the scheduling algorithm. The scheduling algorithm defines the logic to generate a schedule.

4 The resource scheduling model

Prior to selecting the scheduling algorithm, a precise and integral resource scheduling model has to be determined. Scheduling solutions on the market use their own methods to describe the problem. These methods usually depend on the type of the scheduling problem. It is not possible for algorithms to be reused or to exactly determine the quality of the algorithms.

Based on five different scheduling problems, a model was developed to describe any scheduling problem. Cases included the problem of order scheduling in the production industry (Papler, 2001), the human resource scheduling problem in a Slovenian hospital (Baggia, 2004), the scheduling problem in the chemical/pharmaceutical industry (Zentner et al., 1998), the nurse scheduling problem in a Belgian hospital (Vanden Berghe, 2002) and casino personnel scheduling (www.schedulerexpert.com). A detailed description of the model developed and the validation of the model on three real-world cases are included in Baggia (2006).

The most important object presented in the model is a resource. Resources can be described with common attributes disregarding the class they belong to. The model is made extensible through generalization association. All attributes that are not common to different types of resource object describe the subclasses, while common attributes describe the class ***Resource***. In the first iteration of the model, the subclasses ***Person*** and ***Material Resource*** extend the ***Resource*** superclass.

Extensibility using generalization association is also applied to the classes of ***Structure***, ***Process***, ***Unit***, ***Resource Property*** and ***Work***. Besides some common attributes, different attributes are used to describe the specifics of scheduling problems. Subclasses inherit all the attributes and methods of superclasses.

A common set of methods is used to describe the model in the first iteration. Since no scheduling algorithms are used and the scope of the model is to describe the problem, not to solve it, only three types of methods are applied to classes: insert, update and delete.

5 Implementation

Different languages can be used to describe the proposed scheduling model. At the beginning of our project (Baggia, 2005), XML was used to present the scheduling problem because of its simple communication with the database, where all attributes of the scheduling problem not used by scheduling algorithms are stored. It was established in latter phases that an object oriented technique is more appropriate for the problem description as a relational database. A UML Class diagram was used to describe the model of a scheduling problem. The basic possibilities of describing the objects of the extensible model of the scheduling problem with XML, an EAV model and object oriented programming languages are presented.

5.1 XML

The eXtensible Markup Language (XML) is a general purpose markup language presented by the World Wide Web Consortium, whose primary purpose is to facilitate the sharing of data across different systems. It is widely used in combination with databases and the internet.

All the data from previous research into scheduling problems were stored in a database. Due to its compatibility with databases, XML was used at the initial language development phase for the scheduling problem description. XML was used as a meta-language for the description of the scheduling problem. All data that are usually not of key importance when scheduling were stored in the database as additional information.

When using an object oriented approach for the description of scheduling problem, XML was also considered as a meta-language. All objects and their attributes needed for scheduling can be described with XML, while the basic constraints for data insertion can be applied using Document Type Definition (DTD). XML is widely used as a standard for data exchange between different systems. Its greatest advantage lies in its connectivity with different databases as well as programming languages. Specific relations between objects in XML are not easy to describe. One possible solution is to use special XML documents for the description of relations between objects.

According to Kim and Carrington (2000), the UML class diagram offers specific associations such as aggregation, composition, generalization and realization. Most of these associations were used in the class diagram describing the problem of resource scheduling. As presented in Goatly (2001) and SWIFT Standards XML design rules (2001), the rules for transforming UML class diagrams to XML consider associations, but the generalization association cannot be shown at more than one level. Multilevel generalization is dealt with by XMI (OMG, 2005) in a way where attributes, associations and other parts of classes which appear more than once in the inheritance hierarchy are included only inthe subordinate classes. Generalization is one of the key types of associations for the resource scheduling description language. Using generalization the extensibility of the model can be applied. For every extended class in the model, the XML document needs to be updated. The purpose of the extensibility of the model is therefore neglected. It has been established that in this phase of development XML is not appropriate for the description of the scheduling problem. An Entity Attribute Value model can be used to combine the benefits of object-oriented languages and relational databases.

5.2 EAV model

Anhoj (2003) presents the term "Entity-Attribute-Value (EAV) design" for the generic structuring of data in a relational database. In a conventional database design, each parameter of interest is represented in a separate column in a table. As new kinds of data need to be managed, the number of columns and/or tables needs to grow. In the EAV model, data are conceptually stored in a single table with three columns: an Entity (the object being described), an Attribute (an aspect of the object being described), and the value for that attribute (Marenco et al., 2003). In EAV design, one row stores a single fact. In contrast, in a conventional table that has one column per attribute, one row stores a set of facts (Nadkarni, 2005). More tables are needed if one wants to present data in different formats.

An EAV model has some advantages compared to a conventional relational database. The number of attributes in an entity is not limited and no space in a database needs to be reserved for null values. There are also drawbacks in the EAV design. The layout of data needs to be adjusted for the user interface and the handling of queries based on attributes is poor. There are also many drawbacks considering the use of constraints.

As mentioned above, in the case of the scheduling problem, generalization expresses extensibility and is highly important. Generalization can be presented with the EAV model. The storage of data is also convenient, since many different parameters need to be stored when scheduling resources. The key problem is checking consistency. Since the model has to be irrespective of the user interface, constraints are not checked when inserting data. An additional interface should be developed for checking consistency, so the EAV design is not appropriate for the scheduling problem description.

5.3 Object oriented languages

It is not only data types and data structures that are defined in object oriented programming, but also functions that can be applied to the data structures. In object oriented programming, a data structure includes data and functions. As stated by Pinedo (2005) some of the new planning and scheduling systems are based on an object oriented approach and not conventional relational databases.

The main advantage of object oriented programming techniques is that an existing object does not need to be altered when adding a new type of object. The new object inherits the attributes of existing objects. Pinedo (2005) explores the possibility of inheritance and generalization as the main advantage of object oriented techniques. With these attributes, object oriented software is easier to supplement or change.

Wampler (2002) finds that C++ and Java are prevalent on the object oriented languages market. Even though both C++ and Java are object oriented languages, there are some basic differences among them. The discussible problem will determine the language to be used. Due to its simplicity and independence, Java was used in the presented research.

5.4 Java

Java and C++ were both developed to support application programming. Furthermore C++ was also developed to support system programming. In general the syntax of Java is simpler than the syntax of C++. The main difference between the two languages is in the compiling.

One of the reasons for Java's popularity is the World Wide Web and Java's ability to run Web applets directly on any computer or operating system with a Web browser (Wampler, 2002). Another reason is that Java is an excellent programming language. It can not only be used for Web applets, but for programs on almost any computer. In its early days Java was disregarded because of its performance, but this is no longer an issue.

In comparison to the primary proposed XML, Java is not as widely used as XML, but is nevertheless widespread and only a Web browser is needed to read it. In some ways, Java is not really suitable for data storage. It was already established at the beginning of the research that it would be appropriate to use a conventional relational database, but it is not flexible and extensible enough to enable a high quality scheduling problem description as presented in the research.

6 Programming implementation in Java classes

A Java class cannot inherit the behaviour and attributes of more than one superior class. This characteristic of the Java programming language is not crucial for this research since the depth of inheritance enables the extensibility of the description language, not the width of inheritance. The Java class ***Resource*** is presented, with the extended classes ***Person*** and ***Material Resource***. Methods are not presented in the following description.

```
public class Resource
{
  String EndDate;
  String Name;
  int Value;
  String StartDate;
  String ID;
  String CurrentStatus;
  /**
   * Comment here
   * ≅link aggregationByValue
   * ≅label UMLSta1
   * ≅associates <šRSDLExt12.mypackage.State}>
   */
  protected State defines[];
}
```

The **Resource** class is extended with two classes for different types of scheduling problems. The **Material Resource** class inherits the attributes of the **Resource** class and has some attributes that are not significant for the description of human resources.

```
public class MaterialResource extends Resource
{
  String Renewability;
  String MeasureUnit;
  int MinStockLevel;
  String Availability;
  int StartState;
  boolean OrderPossibility;
  int Costs;
  int TargetStock;
  String Stability;
}
```

Some attributes are generally used only to describe personnel. While attributes like the minimal level of stock and target stock are not appropriate for personnel description, some (for example renewability) are obvious. No constraints for the work time description are used for material resources, but attributes linked to work time constraints are an important factor when scheduling human resources, since union and contract constraints must be considered.

```
public class Person extends Resource
{
  int MaxNoWorkHours;
  String ContractType;
  int MaxNoConsecutiveDays;
  String Worktime;
  String Address;
  String FormalTitle;
  int MaxNoConsecutiveWeekends;
  int MaxNoWeekends4Weeks;
  String AcademicTitle;
  int MaxNoAcitivitiesHoliday;
  String Name;
  int MaxNoAcitivities;
  int MaxNoConsecutiveDaysOff;
  int MaxNoAcitivitiesDay;
  String BirthDate;
  int MinNoWorkHours;
  int MaxNoShiftsWeek;
  int MaxNoActivitiesShift;
  int MinNoConsecutiveDays;
  int MinNoConsecutiveDaysOff;
  /**
   * Comment here
   * ≅label UMLOse4
    * ≅associates <{RSDLExt12.mypackage.Delovni-
Nalog}>
   */
  protected WorkOrder creates[];
  /**
   * Comment here
   * ≅label UMLOse2
```

```
 * ≅associates <{RSDLExt12.mypackage.Prerazpo-
reditev}>
 */
protected Rearrangement rearranges[];
/**
 * Comment here
 * ≅label UMLOse1
 * ≅associates <{RSDLExt12.mypackage.Stanje}>
 */
protected State changes[];
}
```

Associations are an important part of the model description. No association is described for the material resource, since no responsibility can be assigned to a material. The ***Person*** class is associated with different classes. A person is responsible for the creation of a work order, for the rearrangement of resources (material and human) and for updating the status of resources. The state is defined for every resource presented in the ***Resource*** class. Other associations are not of general importance when scheduling.

7 Conclusions

The research discusses the problem of a general description of a scheduling domain. The ability to compare different scheduling algorithms is definitely the main advantage of the uniform representation of the scheduling problem.

The programming implementation represents an important challenge in the development of the scheduling domain description language. XML should be appropriate to describe extensible problems in general, but when converting a UML class diagram to XML, there arises the problem of generalization association. An XML document should be updated every time a new attribute or subclass is added to the model. An EAV model is used to describe an object oriented problem in the relational database. The class diagram could be implemented with an EAV model, but no constraints can be applied to the model. Since no additional software is to be used in the model description, the EAV model lacks consistency. In general Java is not widely used as a language for data storage, but it is simple to convert the class to the Java class and all associations can be implemented, including the generalization association with no major changes in the existing structure of the model.

The proposed language has been verified and validated on three different scheduling problems, the problem of production scheduling, the problem of personnel scheduling in a hospital and a problem of timetabling in the faculty. All these problems can be described with the proposed language. Different applications to the real problems of scheduling should be considered in future research.

The possibility of description using other appropriate meta-languages will be researched. Some of the advantages of the relational database and some advantages of the object oriented approach should be combined to gain the best results. In future research the description of scheduling methods will be implemented in the model.

References

Anhoj (2003). Generic Design of Web-Based Clinical Databases. *Journal of Medical Internet Research*, 5(4). 28. October 2005 available from http://www.jmir.org/2003/4/e27/.

Baggia, A. (2004). Meta-Language Framework for Personnel Scheduling. In Jašková, M. (Ur.) *ECON '04*. Ostrawa: Technical University of Ostrava.

Baggia, A. (2005). *Splošna definicija problema razporejanja virov*. Sinergy of Methodologies : Proceedings of the 24th International Conference on Organizational Science Development, Kranj: Moderna organizacija.

Baggia, A. (2006). Jezik za opis problema razporejanja virov (Resource Scheduling Problem Definition Language). Doctoral dissertation. Kranj: University of Maribor.

Brucker, P. (2001). *Scheduling Algorithms.* Berlin: Springer-Verlag.

Colindres, A. (1992). *RCSL: Resource-constrained scheduling language*. Doctoral dissertation, Huston: University of Huston.

Goatly, P. (2001). *Bolero Document Modeling Conventions*. Bolero.net The Electronic Trade Community, 24. august, 2005 available from http://xml.coverpages.org/BoleroConventionsV04.pdf

Kim, S-K. & Carrington, D. (2000). *A Formal Mapping between UML Models and Object-Z Specifications*. The 1st International Z and B Conference, York: Springer.

Kljajić, M., Breskvar, U. & Rodič, B. (2004). Computer aided scheduling with use of genetic algorithms and a visual discrete event simulation model. *WSEAS Transactions on Systems*, 2004, *3*(3):1021-1026.

Marenco, L., Tosches, N., Crasto, C., Shepherd, G., Miller, P.L. & Nadkarni, P.M. (2003). Achieving evolvable Web-database bioscience applications using the EAV/CR framework: recent advances. *Journal of the American Medical Informatics Association.* 10(5): 444-53.

Nadkarni, P. *An introduction to entity-attribute-value design for generic clinical study data management systems*. 28. October 2005, available from http://ycmi.med.yale.edu/nadkarni/Introduction%20to%20EAV%20systems.htm

OMG (2005). XML *Metadata Interchange (XMI) Specification*, 24. august, 2005, available from http://www.omg.org/docs/formal/05-05-01.pdf

Papler, A. (2001). *Interaktivno večkriterijsko razporejanje proizvodnje.* Master thesis, Kranj: University of Maribor, Faculty of Organizational Science.

Pinedo, M. (2002). *Scheduling Theory, Algorithms, and Systems*. Upper Saddle River: Prentice Hall.

Pinedo, M. (2005). *Planning and Scheduling in Manufacturing and Services*. New York: Springer Science+Business Media, Inc.

S.W.I.F.T. (2001). *SWIFT Standards XML design rules version 2.3*, Technical Specification, 24. august, 2005, available from http://xml.coverpages.org/EBTWG-SWIFTStandards-XML200110.pdf

Vanden Berghe, G. (2002). An Advanced Model and Novel Meta-heuristic Solution Methods to Personnel Scheduling *in Healthcare, Doctoral Dissertation, Gent: University of Gent.*

Wampler, B.E. (2002). *The Essence of Object-Oriented Programming with Java and UML*, Boston: Addison-Wesley.

Zentner, M.G., Elkamel, A., Penky, J.F. & Reklaitis, G.V. (1998). A language for describing process scheduling problems. *Computers and Chemical Engineering, 22(*1-2): 125-145.

Alenka Baggia is a teaching assistant at the Faculty of Organizational Sciences, University of Maribor. Her research fields are production scheduling and personnel scheduling.

Robert Leskovar is a professor at the Faculty of Organizational Sciences, University of Maribor. His research fields are multiple criteria decision making, software quality and modelling and simulation. He is a member of many international professional societies.

Miroljub Kljajić is a full professor at the Faculty of Organizational Sciences, University of Maribor. He has been the principal investigator in many national and international modelling and simulation projects. As author and co-author he has published more than 20 scientific articles recognized by the scientific community.

Implementacija modela za opis domene razporejanja

V članku je predstavljen problem splošnega opisa domene razporejanja. Ugotovljeno je bilo, da je algoritem razporejanja splošen ne glede na tip problema razporejanja. Na osnovi petih različnih domen razporejanja je bil razvit splošni model opisa. Raziskava se osredotoča na programsko aplikacijo modela razporejanja virov, ki je predstavljen kot UML diagram razreda. Za opis modela so bili preučeni različni meta jeziki. Med njimi so se XML, EAV model in objektno orientirani jeziki pokazali kot najbolj uporabni. Čeprav se Java ne uporablja pogosto kot opisni jezik, se je izkazala kot najbolj učinkovit meta jezik za opis razširljivega modela razporejanja.

Ključne besede: Razporejanje, Opis domene, Jezik za definicijo, Objektno orientirana analiza