Published in MISTA 2003, pp. 566-570

Towards an XML based standard for Timetabling Problems: TTML

Ender Özcan
Yeditepe Üniversity
eozcan@ics.yeditepe.edu.tr

Abstract: There is a variety of approaches developed by researchers to solve different instances of timetabling problems. During these studies different data formats are used to represent a timetabling problem instance and its solution, causing difficulties in the evaluation and comparison of approaches and sharing data. In this paper, a model for timetabling problems and a new data format for them based on XML is proposed.

Key words: timetabling, standard data format, scheduling, XML, SGML

1. INTRODUCTION

Timetabling problems are hard to solve constraint optimization problems. Since there is no common standard on specifying a timetabling problem instance and its solution proposed by a researcher, most of the results can not be compared and benchmarking becomes almost impossible.

Proposal for a common data format for timetabling is initiated by Andrew Cumming at ICPTAT'95. Studies in the area yield a language named SSTL [2, 3], that is freely available on the Internet.

Causmaecker *et. al.* [1] argues that timetabling research community can benefit from Semantic Web, introduced by Lee [5], founder of World Wide Web Consortium (W3C). One layer of this architecture requires an Extensible Mark up Language (XML) definition. XML can be used to obtain a standard data format for timetabling problems. In this paper, a preliminary study for an XML based data format, named as Timetabling Mark-up Language (TTML) is presented.

2. TTML: TIMETABLING MARK UP LANGUAGE

XML lets users to create their own set of tags, enabling them to specify the structure of their documents. Furthermore, XML can be used to define a set of grammar rules to define mark up languages. MathML, providing means to use mathematical expressions in the web, Scalable Vector Graphics, describing two-dimensional graphics in XML, are examples of standard XML based languages. MathML intends to encode both mathematical notation and mathematical meaning. All the details about XML can be found in W3C site [6]. It is vital to clearly define and represent the elements of a timetabling problem using TTML.

An XML document requires one unique root element. The root element can be selected as `time-tabling` for a timetabling problem instance (*Figure 1*). Our first aim is to enable data exchange; hence a TTML document must include input data for the problem instance. Additionally, for the research community, in order to make comparisons, test results obtained from applying an algorithm to the input data should be attached. Further attachments might be required, such as output view for the solution.

```
-<time-tabling>
   +<input-data>
   +<output>
   +<test-results>
```

```
-<input-data>
   +<variables>
   -<domain>
      +<time>…
   +<constraints>
```

```
-<constraints>
  +<no-overlap>
  +<exclude>
  +<preset>
  +<ordering>…
```

Figure 1. Main and some child elements of a TTML document, where bold elements are optional

2.1 Modelling Input Data

A timetabling problem (TTP) can be represented using (V, L, C), where V is a set of variables; $V = \{v_1, v_2, \ldots, v_i, \ldots, v_P\}$, L is a set of domains of variables; $L = \{d_1, d_2, \ldots, d_i, \ldots, d_P\}$, where d_i is the domain of the variable v_i, and $d_i \subseteq D_1 x D_2 x \ldots x D_l x \ldots x D_Q$, $1 \leq U$, and C is a set of constraints. As an example, let's consider a university course/lecture timetabling problem. V might be a set of course meetings. For simplicity, a cross product of two sets might be a domain for each variable; $D_1 x D_2$, where $D_1 = \{t_1, \ldots, t_j, \ldots, t_M\}$, representing start times (or intervals) for a course meeting and $D_2 = \{r_1, r_2, \ldots, r_k, \ldots, r_S\}$, representing the classrooms. TTP can be described as a search for finding the best *assignment* (v_i, t_j, r_k) for each variable $v_i \in V$, such that, all the constraints are satisfied in C. The assignment implies that the course meeting of v_i starts at t_j in the classroom r_k. TTML shall support description of each one of these sets as an input data.

Time in a timetable can be represented using *intervals*. As one of the sets in the domain of a variable, time set might contain discrete or continuous values. In addition, the resolution of a time interval and periodicity are relevant notions. During the declaration of time as a domain, TTML shall consider all these features.

2.1.1 Modelling Constraints

In general, six different constraint types can be identified for TTPs: *exclusions*, *presets edge constraints*, *ordering constraints*, *event-spread constraint* and *attribute constraints* (includes *capacity constraints*). The details about these constraints can be found in [4]. TTML shall support all of them. Constraints can be further classified as *hard* and *soft* for TTPs, where no violation is allowed for a hard constraint and soft constraints are the preferences that are strongly desired. TTML shall distinguish between hard and soft constraints.

In order to define constraints in a TTP appropriately, an additional set is proposed, that is *H(S)*, representing a set of *classifications* defined on *S*, where *S* is a set. A classification is a set of some subsets of *S*, representing a logical grouping in the context of timetabling. For example, one can group the course meetings with respect to the lecturers. As a result, the lecturer classification contains the course meetings of each lecturer. Another classification is possible by grouping the course meetings with respect to the students. It is much more proper to define a single constraint for lecturers or students, as an example, declaring that the course meetings of each lecturer or each student in a classification should not overlap, as compared to defining multiple constraints, where each constraint declares a pair of course meetings that should not overlap. TTML shall support declaration of classifications, hence both of these constraint declaration styles.

Set definitions in timetabling should be extended to allow announcement of an ordered set of *attributes* for each member of a discrete set. Attributed sets allow declaration of more complex constraints. For example, an attribute for a course meeting might be the total number of students taking the course, and an attribute for a classroom might be its capacity, a possible constraint would be the total number of students taking a course should not exceed the capacity of the classroom. Constraints can be defined using direct values of attributes or using a mathematical formula combining them. TTML shall support attributed sets and describing constraints on them.

MathML supports many notions regarding to the theory of sets for content markup, such as, defining sets, lists and functions that can be applied on them to produce sets. TTML can benefit from MathML by either imitating it, or directly embedding it into itself. In either way, users will be able to represent complex constraint functions using TTML.

2.1.2 Modelling Output and Test Results

Optional `output` element determines the visualization of the results with respect to classifications. This element is for future use in web services. By default, all the assignments for each variable should be generated by a TTML processor. Yet, a client might desire a different view of the output, e.g., weekly schedules of all lecturers or students. TTML elements for test results are for researchers to enable benchmarking. It will contain an indicator showing the type of algorithm is used, evaluation function and the best result obtained. Test results should contain multiple runs. Representing an evaluation function based on penalties is easier, since there are not so many different such functions. MathML will be also helpful in the design of the evaluation function element, allowing user-defined complex functions.

3. CONCLUSIONS

TTML can model real-world timetabling problems, providing benefits for developers, clients (end users) and researchers. Data sharing will be easy and fast. Any TTML document, produced by any person, can be processed by an expert timetabling application accepting TTML input, and generate a solution. Researchers can develop their own application to perform their own experiments on a given data, subject to given constraints, then attach their results for comparison and publish their data as a TTML document. TTML provides all the advantages and strengths of XML, allowing web based application development. The requirements for a standard data format in timetabling can be summarized as universality (assuming a closed world), completeness and convertibility. The latter requirement is, totally, satisfied by TTML, just by being an XML standard. TTML requires more work to satisfy universality and completeness. Model explained above is a strong candidate to reach this goal with extra power of MathML.

4. REFERENCES

1. P. D. Causmaecker, P. Demeester, Y. Lu and G. Vanden, *Using Web Standards for Timetabling*, PATAT'02, pp.238-258, 2002.
2. E. K. Burke, P. A. Pepper and J. H. Kingston, *A Standard Data Format for Timetabling Instances*, Springer Lecture Notes in Computer Science, 1408:213-222 (1997).
3. J.H. Kingston, *Modeling timetabling problems with STTL*, Springer Lecture Notes in Computer Science, 2079:309, 2001.
4. H.L. Fang, *Genetic Algorithms in Timetabling and Scheduling*, PhD thesis, 1994.
5. T.B.-Lee, J. Hendler and O. Lassila, *The Semantic Web*, Scientific American, May 2001.
6. World Wide Web Consortium web site, http://www.w3c.org, 2002.