Towards A Unified Timetabling Model

Jeffrey H. Kingston

School of Information Technologies, The University of Sydney, Australia
jeff@it.usyd.edu.au
http://jeffreykingston.id.au

Abstract. Timetabling has many sub-disciplines: nurse rostering, high school timetabling, examination timetabling, and so on. Their problems are usually considered to be separate, but there might be advantages in unifying them. This paper describes some advantages of unification, analyses a few design issues, and tentatively suggests some design ideas. The work is incomplete and is offered as a stimulus to discussion, not as a formal proposal.

Keywords: Timetabling · Modelling.

1 Introduction

Timetabling has many sub-disciplines: nurse rostering, high school timetabling, examination timetabling, and so on. Their problems are usually considered to be separate, but there might be advantages in unifying them.

Timetabling problems may be modelled using *events*, which are meetings at which some *resources* (teachers, rooms, and so on) are occupied together at some *times*, and *constraints*, which are rules that limit the time and resource assignments, with penalties (hard and soft) to impose if the rules are broken. This basic common structure is what makes unification possible.

This paper describes some advantages of unification, analyses a few design issues, and tentatively suggests some design ideas. The work is incomplete and is offered as a stimulus to discussion, not as a formal proposal.

2 Why unify?

Why unify at all? Isn't it sufficient if research supervisors keep abreast of developments in the various sub-disciplines?

There is no pressing need to unify. Good progress is being made without it, and solvers for the unified problem are unlikely to perform better than solvers for specific problems. But there are other reasons for unifying, as follows.

Even if solvers do not benefit, other kinds of software might: software for evaluating solutions like the HSEval web site [6], or solve platform software like KHE [7]. These two systems currently support high school timetabling and nurse rostering, but nothing else.

Specification work done in one sub-discipline may be valuable in another. The XESTT nurse rostering model [10] was based on the XHSTT high school timetabling model [15]. It is argued below that UniTime [20], the leading university course timetabling model, would benefit from some XHSTT ideas.

Unification might also shed light on the scientific question of how related the sub-disciplines are. Student sectioning, for example, as found in universities and some high schools, seems to be different from everything else [8]. But is that really so? Such questions have little practical importance, but surely there is room in our discipline for some intellectual curiosity about them.

Perhaps the strongest reason concerns problems that fall outside, or span across, the usual categories: hospital problems other than nurse rostering, high schools whose senior students follow a university-style curriculum, and so on. These problems are widespread, but they are marginalised at present. A unified model would help to specify them and bring them into the mainstream.

Our main concern in this paper is with what might be called 'internal models': models concerned with precisely specifying instances, ready for solving. There are also 'external models', which express instances in terms that end users are familiar with. Conversion between external and internal is a significant issue that we do not address here; it is in practice aided by software, such as instance translators and interactive user interfaces.

3 Existing models

The author knows of no recent models that try to cover all or most sub-disciplines of timetabling. So this paper is inspired by analyses of several sub-disciplines and their leading models. These are the subject of this section.

Many years ago several authors designed unified timetabling models built on mathematical foundations, for example [5, 13]. This work proved to be useful only in showing what not to do. Its excessive generality was avoided in the following generation of models, many of which are discussed below. Although our proposal could be described as general because it spans multiple sub-disciplines, it is actually just as concrete as the models we'll discuss now.

High school timetabling was very fragmented for many years. Virtually no data was exchanged, at least at the international level, before the advent of XHSTT [15]. It improved the situation greatly, opening the way to the Third International Timetabling Competition [16], and also to the XHSTT-2014 data set [14], which contains real-world instances from many countries.

To this author, XHSTT's major choices still seem right ten years later. There are opportunities for increasing generality and reducing verbosity, but XHSTT's four-part structure (times, resources, events, and constraints) has worked well, and several specific ideas deserve to be passed on to new models: the resource concept (the old idea [1] that all entities that attend events, including teachers, rooms, nurses, and so on, are basically the same); the ability to define arbitrary sets of times and resources and use them in constraints; and the way that cost calculations are specified transparently and uniformly over all constraints.

Examination timetabling was the first sub-discipline to have a widely used model, the one for the Toronto instances [17, 18]. Its constraints are mainly local workload limits. It has been criticised for incompleteness (for example, it omits rooms). The more complete models that have emerged more recently have added features already found in other models, such as university course timetabling.

Nurse rostering has a solid history of modelling and data sharing. In fact, it arguably has too many models rather than too few. These models were analysed during the design of the XESTT nurse rostering format [10], which turned out to be XHSTT with some enhancements, including two new constraints. Some recently suggested nurse rostering constraints [4] are expressible in XESTT.

University course timetabling has only one widely cited model: UniTime. Its creators have investigated the requirements of many universities (at least ten [12]), and their current model seems to be close to complete. This author has used the UniTime university course timetabling model, version 2.4 [20] as his main source for university course timetabling, along with the closely related International Timetabling Competition 2019 (ITC2019) model [12].

University course timetabling has several challenging features. Its instances can be very large, with thousands of students, perhaps, each of whom needs an individual timetable that may vary from week to week. Then there is student sectioning, which can be more than assigning each student to one section of a course, since a course may have multiple configurations, each with a variety of parts (lectures, tutorials, and so on), logically related in various ways. Finally there are room capacity and walking time constraints.

Generalization would improve UniTime. Introducing the resource concept, and expressing the common parts of constraints uniformly, are two obvious steps. Arbitrary sets of times would also help: several of UniTime's 'group constraints' are identical except for the sets of times they apply to.

Sports scheduling needs one event for each fixture, holding two preassigned resources (the teams) and possibly a third one (the venue). The well-known travelling tournament problem [3] is precisely specified, but not very general. The only general model is the RobinX round-robin tournament model [19, 21]. Its creators say that it is based on analyses of many real-world tournaments.

RobinX defines 21 constraints. Some are familiar from other models. For example, there is a travel distance constraint like the university constraint on walking time. Others are more familiar than they seem. For example, RobinX schedules have a particular large-scale structure, such as *mirrored round-robin*. These are enforced by constraint `SE2`, which says that if one event is assigned a certain time, a corresponding event must be assigned a certain other time. This is similar to the nurse rostering 'complete weekends' constraint, which says that nurses who work on Saturday or Sunday must work on both days.

However, there are also constraints that do not seem to fit well with other models. Some RobinX constraints aim to produce a *fair* schedule: one which spreads defects evenly among the teams. The XHSTT approach to fairness is to specify an acceptable maximum number of defects and use a quadratic cost function to strongly penalize larger deviations. RobinX tries to minimize the maximum, over all pairs of teams, of the difference in value. And some RobinX constraints, notably `CA5` and `GA2`, seem to be far removed from other models. Altogether, then, sports scheduling will seriously test a unification project—but it is important, for that very reason.

4 Issues

This section discusses some issues raised by unification. The discussion often takes XHSTT as its starting point, but only because it is familiar, precisely specified, and already supports much of what is needed.

4.1 Times

As mentioned, the various sub-disciplines have a basic similarity that makes unification possible. But they model time differently. High school timetabling has *periods*, which are disjoint time intervals. RobinX is much the same. Nurse rostering has *shifts*, which are longer intervals that may overlap. UniTime has hierarchically defined sets of intervals, such a '9-10am Monday, Wednesday, and Friday, starting in Week 2 of semester'. But still there is a basic similarity: all models deal with intervals, or finite sets of intervals, of real time.

What seems to be needed is a finite set of *times*, each representing a finite set of intervals of real time. Importantly, each event must choose its times from this finite set: no timetabling problem known to the author allows solutions to choose arbitrary time intervals. In this way, time is discretized.

There are implementation efficiency issues in calculating overlaps between times. UniTime addresses these issues by requiring the sets of intervals that can be times to have a regular hierarchical structure (same time each week, etc.). A general model should allow a time to represent an arbitrary finite set of intervals, but in a way that optimizes calculations involving regular structures. The author has designed a time model of this kind (unpublished). In this model, the hierarchy (semesters, weeks, days) is arbitrary and is defined within the instance, and one time is very like a finite set of UniTime times.

4.2 Resources

Resources are entities that attend events: teachers, rooms, nurses, and so on. XHSTT allows any number of resource types to be defined (`Teacher`, `Room`, and so on), along with any number of resources of each type. Arbitrary sets of resources, called *resource groups*, may be defined and used in constraints.

It seems to be necessary to add resource attributes, such as room capacity and location, since some constraints in models other than high school timetabling depend on them. XHSTT does not offer resource attributes, except membership in a resource group, which is equivalent to a Boolean attribute.

Some attributes can be simulated by constraints. For example, a teacher's workload limit is an attribute of the teacher, but it is expressed in XHSTT by the upper limit of a limit workload constraint that applies to the teacher. But even if this could be made to work in all cases, it is unnatural and can become very verbose. One would not want to add an entire constraint for each pair of rooms, for example, just to hold the walking time between them.

4.3 Events

An *event* is a meeting between some resources at some times. The times and resources may be preassigned, or left to a solver to assign. We think of an event as a variable which may be assigned a set of times, but which also contains other variables called *tasks*, each of which may be assigned a set of resources.

An XHSTT task may be assigned only one resource, but problems other than high school timetabling need a more flexible design: a task that may be assigned several resources. Their number may be constrained but need not be fixed. This is the norm in nurse rostering, for the nurses assigned to a shift, and in university course timetabling, for the students assigned to a lecture.

There is a big difference between, say, a university course meeting at 9am on Mondays, Wednesdays, and Fridays, and a high school course meeting at 9am on Mondays, 11am on Wednesdays, and 3pm on Fridays. The university course meets at rigidly coupled times, so it could be represented by an event assigned a single time containing three intervals. The high school course meets at separately chosen times, so it must be represented by an event assigned three times, each containing one interval. Care is needed over resource stability: a course must be able to meet in different rooms at different times, if desired.

4.4 Constraints

Similar constraints from different sub-disciplines should be merged into a single constraint (possibly generalized) in the unified model, where possible. Several familiar constraints are easily merged, including constraints that limit events and tasks to preferred times and resources, and constraints on resources that prohibit clashes, specify unavailable times, and impose workload limits.

Here is a deeper example. Consider a constraint that limits the length of sequences of consecutive free days, as is found in nurse rostering. This is based on a sequence of sets of tasks, one set for each day, containing the tasks that a given resource is assigned on that day. Maximal sequences of consecutive empty sets must be identified and the 'length' function applied to them.

Now consider a constraint that limits the number of idle times that a resource may have on one day (times when the resource is not busy, provided it is busy earlier and later in the day). Again, this is based on a sequence of sets of tasks, one for each time of the day, and again sequences of empty sets of tasks must be identified. The function is different (the total length of empty sequences not at the ends, rather than the length of each sequence taken separately), but the two constraints have the same underlying structure.

Or consider a nurse rostering constraint requiring at least one senior nurse to be on duty at 3pm. XESTT models this by a *limit resources constraint*, which takes some tasks and a set of resources, and imposes lower and upper limits on the number of assignments to the tasks of resources from the set. In the example, the tasks are all tasks running at 3pm (possibly from several shifts), the resources are all senior nurses, and there is a lower limit of 1.

This constraint can also be used to specify that a given university student should attend one of the laboratory sessions of a given course. The tasks are the student tasks of the course's laboratory events, the resources consist of just the given student, and there are lower and upper limits, both 1.

But however cleverly we find such connections between constraints, a serious problem remains. Several sub-disciplines have constraints that cannot be fitted to a general pattern, because they use functions, such as Pythagorean distance, or the RobinX fairness function, which are essentially arbitrary. When all of them meet in a unified model, the result might be chaos.

A possible way out is to provide a single, uniform method of selecting the tasks to which a constraint applies. This would select all tasks from a given set of tasks S which are assigned resources from a given set of resources R and lie in events assigned a time from a given set of times T. Then the only arbitrary aspect would be the function to apply to the selected tasks, which could be taken from a long fixed list: the number of tasks, their total duration in minutes, the walking time between their locations, and so on. The author has designed constraints of this kind (unpublished) which can express all of the constraints of all of the models described in Section 3. Arbitrarily complex *constraint trees* are included; they are needed for student sectioning.

4.5 Reducing verbosity

It seems inevitable that a unified model will ultimately be expressed in XML. XML-based models are notorious for leading to large, verbose files.

One way to reduce verbosity is to reduce repetition. For example, XESTT allows a constraint to be defined for one day (or week etc.), and then annotated with 'and repeat this every day (or every week)'. This could be generalized to nested iterators ('for each resource r in set R, for each time t in set T, ...').

Verbosity can also be reduced in a more local way. For example, XHSTT consumes a lot of space saying whether a constraint is hard or soft, what its weight is, and so on. All this could be replaced by a single string, for example

```
cost="count:2-5|s20"
```

which says (reading the bar as 'or else'), '`count` (the number of tasks) must lie between 2 and 5, or else a soft cost equal to the distance from these limits times 20 is incurred'.

Lower and upper limits, like 2 and 5, appear in many XHSTT constraints but not all. It seems to be a good idea to give them to all constraints: it is more general and more uniform and creates no problems. It was pointed out long ago that limits on the measures of sets are basic in timetabling [11].

5 Conclusion

This paper has offered some ideas for unifying the sub-disciplines of timetabling. Although the ideas are incomplete, the author hopes that they will stimulate interest in a field which does not really exist, but perhaps should.

References

1. Appleby, J. S., Blake, D. V., Newman, E. A.: Techniques for producing school timetables on a computer and their application to other scheduling problems. The Computer Journal **3**, 237–245 (1960)
2. Ceschia, S., Nguyen T. T. D., De Causmaecker, P., Haspeslagh, S., Schaerf, A.: Second international nurse rostering competition (INRC-II), problem description and rules. oRR abs/1501.04177 (2015). http://arxiv.org/abs/1501.04177.
3. Easton, K., Nemhauser, G., and Trick, M.: The traveling tournament problem description and benchmarks. In: Walsh, T. (ed.) Principles and Practice of Constraint Programming (CP 2001), 580–584, Springer, Berlin, Heidelberg (2001)
4. Gätner, J., Bohle, P., Arlinghaus, A., Schafhauser, W., Krennwallner, T., Widl, M.: Scheduling matters: some potential requirements for future rostering competitions from a practitionerâĂŹs view. In: PATAT 2018 (Twelfth international conference on the Practice and Theory of Automated Timetabling, Vienna, August 2018)
5. Kingston, J. H.: Modeling timetabling problems with STTL, Springer Lecture Notes in Computer Science **2079** 309 (2001)
6. Kingston, J. H.: The HSEval High School Timetable Evaluator, http://jeffreykingston.id.au/cgi-bin/hseval.cgi (2010)
7. Kingston, J. H.: KHE web site, http://jeffreykingston.id.au/khe (2014)
8. Kingston, J. H.: Integrated student sectioning. In: PATAT 2014 (Tenth international conference on the Practice and Theory of Automated Timetabling, York, UK, August 2014, 489–492 (2014)
9. Kingston, J. H.: XESTT web site, http://jeffreykingston.id.au/xestt (2018)
10. Kingston, J. H.: A unified nurse rostering model based on XHSTT. In: PATAT 2018 (Twelfth international conference on the Practice and Theory of Automated Timetabling, Vienna, August 2018)
11. Kitagawa F., Ikeda, H.: An existential problem of a weight-controlled subset and its application to school timetable construction. Discrete Mathematics **72**, 195–211 (1988)
12. Müller, T., Rudová, H., and Müllerová, Z.: University course timetabling and International Timetabling Competition 2019. In: PATAT 2018 (Twelfth international conference on the Practice and Theory of Automated Timetabling, Vienna, August 2018)
13. Ozcan, E.: Towards an XML based standard for Timetabling Problems: TTML. In: Multidisciplinary Scheduling: Theory and Applications, Springer Verlag 163 (24), May 2005
14. Post, G.: XHSTT web site, http://www.utwente.nl/ctit/hstt/ (2011)
15. Ahmadi, S., Daskalaki, S., Kingston, J. H., Kyngäs, J., Nurmi, C., Post, G., Ranson, D., Ruizenaar, H.: An XML format for benchmarks in high school timetabling. Annals of Operations Research **194**, 385–397 (2012)

16. Post, G., Di Gaspero, L., Kingston, J. H., McCollum, B., Schaerf, A.: The Third International Timetabling Competition. In: PATAT 2012 (Ninth international conference on the Practice and Theory of Automated Timetabling, Son, Norway, August 2012), 479–484 (2012)
17. Qu, R., Burke, E., McCollum, B., Merlot, L., Lee, S.: A survey of search methodologies and automated system development for examination timetabling. Journal of Scheduling **12** 55–89 (2009)
18. Qu, R.: Benchmark data sets in exam timetabling, http://www.cs.nott.ac.uk/˜rxq/data.htm. (2012)
19. RobinX: An XML-driven classification for round-robin sports timetabling, http://www.sportscheduling.ugent.be/RobinX/index.php (2019). Note: the constraints are listed at http://www.sportscheduling.ugent.be/RobinX/threeField.php
20. The Comprehensive University Timetabling System, www.unitime.org, especially www.unitime.org/uct_dataformat_v24.php (2018)
21. Van Bulck, D., Goossens, D., Schönberger, J., Guajardo, M.: RobinX: an XML-driven classification for round-robin sports timetabling. In: PATAT 2018 (Twelfth international conference on the Practice and Theory of Automated Timetabling, Vienna, August 2018)