CLP(FD)-based Information Systems in Space Management Optimization

Irene Rodrigues*, **Daniel Diaz**†, **Salvador Abreu***

* University of Évora and LISP – Portugal
{ipr,spa}@di.uevora.pt

† University Paris-1 and CRI – France
Daniel.Diaz@univ-paris1.fr

Abstract

Declarative techniques such as Constraint Logic Programming (CLP) can be very effective in modeling and assisting management decisions. We present a CLP-based local search method and a tool for managing university classrooms which extends the previous design to deal with spatial resource optimization issues, along two dimensions: classroom use and occupancy rates, while still meeting student and lecturer preferences. Our method was experimentally found to reach a 30% improvement in space utilization, while preserving the quality of the timetable, both for students and lecturers.

1. Introduction

Declarative methods and tools have long been a hallmark of Artificial Intelligence-enabled applications. One is *Constraint Logic Programming* (CLP) (Colmerauer, 1990; Jaffar and Lassez, 1987), which extends Logic Programming by allowing variables to take values over a specified domain, while ensuring that relations amongst them hold true – these relations are called *constraints*. For some classes of problems, CLP provides very efficient solvers while retaining the ability to formulate the problem declaratively, relying on high-level concepts which may be very close to the application domain. Constraints may be used to formulate Constraint Satisfaction Problems (CSPs) as well as Constraint Optimization Problems (COPs).

Space management in higher education institutions (HEIs) is a recognized research subject, with published work by several authors (Planning, 2006; Beyrouthy et al., 2010; Abdullah et al., 2012). One of the topics associated with space management in HEIs is the preparation of class schedules, which has also been the focus of research and applied work (Rudová et al., 2011).

The preparation of class schedules is a process with a very significant impact on the teaching activities of universities, since it entails managing physical resources which are not amenable to change at will, such as the capacity of a classroom. Moreover, this process may be overconstrained by contradictory requirements, for instance: from the student's point of view one wants to concentrate classes in the same part of the day while for the lecturer one wants to minimise the number of days, which could imply having classes both in the morning and in the afternoon, thereby defeating the first constraint.

We claim that CLP is useful in designing and implementing components of an information system. To illustrate the point, we describe an actual application – the development and deployment of a timetabling system for Higher Education establishments, which incorporates several forms of resource optimization.

2. Class Scheduling in HEIs

Class scheduling is at the heart of the activity of Higher Education institutions: it is a mission-critical task, with a clear impact on the physical resource usage but also on the percieved performance of the organisation. In the specific case of the University of Évora, the process of preparing the schedules is done by means of an in-house application that satisfies a set of underlying general requirements governing timetable preparation, which include:

- Classes within a week are organized in blocks of 90 or 120 minutes each,
- Classes are concentrated in a particular part of the day, either the morning or the afternoon, to ensure the existence of free time for students to study on their own,
- Classes occur in the 08:00-20:00 hour range, with the exception of the 13:00-14:00 period which is reserved for lunch,
- The week goes from Monday to Friday,
- From the standpoint of the lecturers, classes tend to concentrate in at most two distinct weekdays.

In our previous work, we extended the design methodology to include requirements specified as *constaints*, which led us to develop an application (Rodrigues et al., 2013) that has proven able to meet the general objectives outlined above.

However, subsequent analysis of schedules generated by the application has revealed that, from the point of view of space management, we tend to get a relatively low level of frequency of use and rate of occupancy of classrooms.

3. Persistent Contextual Constraint Logic Programming

The design and implementation of this web-based information system component relies on the ISCO (Abreu, 2001) language and tools, which takes CLP as provided by GNU Prolog (Diaz et al., 2012) and extends it with

Object-Oriented mechanisms: the Contextual Logic Programming language CxLP. Furthermore, the Prolog basis is augmented with persistence via the ISCO language, which transparently relies on external data providers such as relational databases, to store and retrieve structured data.

One important aspect of CLP is its hybrid search procedure: on one hand it behaves like Prolog, with non-deterministic variable bindings and chronological backtracking to enumerate all the possible values. On the other, we have constraints which bind variables in a way which proactively narrows the admissible set of values for variables, just because they occur in constraints with other variables. CLP thus provides a mix of *a-posteriori* search-space exploration with *a-priori* search-space pruning (via constraint propagation.)

For this application we added the built-in predicate `call_with_timeout/2`, which runs a goal until success or a given time has elapsed. This was essential to carry out best-effort optimisation tasks in bounded time.

4. The Timetable Manager

The timetable manager deals with all aspects of specific timetables: the generation, editing, display and other bookkeeping tasks. The timetable manager has the option to manually edit partial timetables and to automatically generate partial or complete timetables. The automatic timetable generation can be done incrementally by collecting a set of class (course) variables, as required for a given degree curriculum.

The system enforces the timetable hard constraints both for the automatic generation and for manual editing. In the latter case, it makes no sense to take into account the soft constraints, however it is possible to subsequently evaluate the solution after it has been persistently stored.

The timetable may be incrementally built, resulting in a partial schedule that is represented in a way which is always guaranteed to be coherent.

5. Optimization

The application is able to optimize the university timetable, according to a utility function that enables us to evaluate and compare timetable instances. The optimization can be done gradually since the current best solution is always kept in persistent storage, which allows for an incremental usage pattern.

We use Hill Climbing for the optimization algoritm. It starts with the current timetable, a solution. A set of classes is randomly choosen and their values are recalculated in order to obtain a new candidate solution. Whenever an improvement is obtained, the new timetable is is updated in persistent storage. The Constraint Optimization Procedure COP uses the FD constraints provided by GNU Prolog and ISCO to interface the persistent storage, supplied by a PostgreSQL object-relational database. Notice that COP operates by selecting the best configuration which may be obtained *within a given time limit*.

6. Closing Considerations

In this article we have taken a real-life case study: the timetabling problem for a University, which we had already shown to benefit from using constraints in the design and implementation process of the information system. We pushed the envelope by following the same design philosophy when dealing with the more complex issue of optimization for physical space management.

In the process, we mixed the power and expressiveness of the constraint-based local search optimizer for space management, with the flexibility of an incremental user-controlled application interface. It turns out that this mix of reassuring conservativeness and resource-efficient eager optimization strikes an effective balance, which is appreciated by the end-users. Moreover, the end results translate to a significant savings in physical resource usage, with the ensuing economic and organisational benefits.

The timetabling application is presently managing over 60 curricula, totaling over 2500 courses taught by about 500 faculty members in more than 200 classrooms distributed over 10 different buildings. The information system and constraint solver run very well on a low-power virtual machine, meaning that response time is barely perceptible to solve a single timetabling problem.

7. References

Abdullah, Shahabudin, Hishamuddin Mohd Ali, Ibrahim Sipan, Mariah Awang, Mohd Shahril Abdul Rahman, Suleiman Aliyu Shika, and Jibril D. Jibril, 2012. Classroom management: Measuring space usage. *Procedia - Social and Behavioral Sciences*, 65(0):931 – 936. International Congress on Interdisciplinary Business and Social Sciences 2012 (ICIBSoS 2012).

Abreu, Salvador, 2001. Isco: A practical language for logic-based construction of heterogeneous information systems. In *INAP*.

Beyrouthy, Camille, Edmund K. Burke, Barry McCollum, Paul McMullan, and Andrew J. Parkes, 2010. University space planning and space-type profiles. *J. Scheduling*, 13(4):363–374.

Colmerauer, Alain, 1990. An introduction to prolog III. *Commun. ACM*, 33(7):69–90.

Diaz, Daniel, Salvador Abreu, and Philippe Codognet, 2012. On the implementation of GNU Prolog. *Theory and Practice of Logic Programming*, 12(1-2):253–282.

Jaffar, Joxan and Jean-Louis Lassez, 1987. Constraint logic programming. In *Conference Record of the Fourteenth Annual ACM Symposium on Principles of Programming Languages, Munich, Germany, January 21-23, 1987*. ACM Press.

Planning, Kilner, 2006. Space utilisation: practice, performance and guidelines. Technical Report 38, Space Management Group, http://www.smg.ac.uk/documents/utilisation.pdf.

Rodrigues, Irene Pimenta, Nuno Matos, Salvador Abreu, Rébecca Deneckère, and Daniel Diaz, 2013. Towards constraint-informed information systems. In Roel Wieringa, Selmin Nurcan, Colette Rolland, and Jean-Louis Cavarero (eds.), *RCIS*. IEEE.

Rudová, Hana, Tomás Müller, and Keith S. Murray, 2011. Complex university course timetabling. *J. Scheduling*, 14(2):187–207.