Declarative Modelling and Efficient Optimization using Constraint Technology

Ulrich Geske[1], Armin Wolf[2]

[1]University of Potsdam, [2]Fraunhofer FIRST, Berlin
Ulrich.Geske@uni-potsdam.de, Armin.Wolf@first.fraunhofer.de

Classic methods for planning, simulation and optimization are mainly designed for conflict resolution. Conflict resolution means that a solution or partial solution is generated and subsequently tested to determine whether the conditions are met (generate-and-test procedure). This track of the Informatik 2008 workshop program describes the use of the Constraint Programming (CP) technology to deal with complex problems. The main advantage of the CP paradigm is the strategy to avoid conflicts a-priori. The use of this conflict-avoiding strategy is advantageous, when strongly branched decision trees arise. The main advantage of constraint-based methods is on the one hand that any decision, e.g. an admissible value is assigned to a variable, leads directly to a reduction of the search space and on the other hand that inadmissible values for variables are immediately recognized and rejected. This is achieved unlike in the classic programming, by placing equality or inequality conditions early in the program (and its execution), even if they are not yet testable because the included variables have not been assigned (see also Figure 1).

Former research and industrial projects have shown that constraint-based methods are suitable for complex problems achieving results highly efficiently and with good optimality properties. In addition, these methods can be easily combined with interactivity (for online access to the knowledge of a domain expert) and rule-based processing (e.g. for integration of background knowledge). The theoretical possibility of exponential runtimes for large problems still remains, but the size of problems being dealt with can be often extended considerably by using CP because any additional constraint is immediately used to prune the domains of the constrained variables. Therefore any change of the variables domain – which is yet another constraint – is propagated until a fix-point is reached. During search this will prune the search space as early as possible such that dead ends are detected early. So, if an inconsistency is detected during search, backtracking will be performed to make other decisions, i.e. to add other constraints (e.g. assign other values to variables). This processing of constraints appears costly, but in fact, it improves the overall efficiency compared with classical methods especially enabling the solution of NP-complete problems. Thus Constraint Programming paradigm essentially consists of two interleaved parts: pruning of infeasible values and search for a solution While pruning leads by definition to a reduction of the complete search space, exhaustive search will find any existing solution. Practical applications have shown that problem-specific tailored search procedure is important to find a solution efficiently.

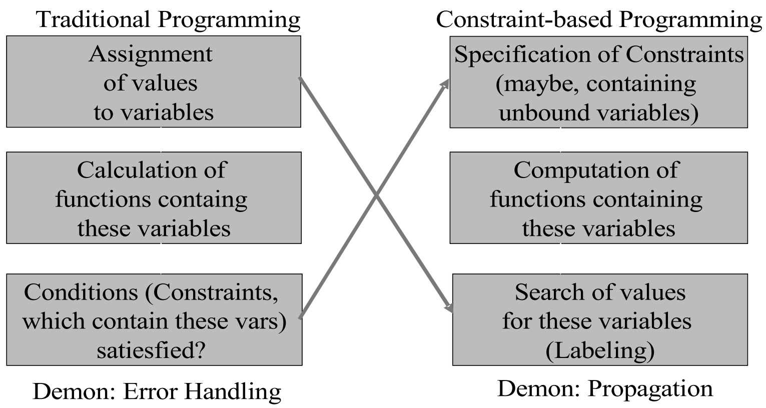

Fig. 1: Comparison of paradigms: classical and constraint-based programming (Sequence of the specification and processing of instructions)

The papers accepted for the workshop start with two contributions to the search problem while the other papers deal with special aspects of application problems.

Henry Müller proposes in "Search Design Patterns" a platform independent, design method for „search modules". Such modules realize complete search algorithms. They are flexible, reusable and easy to understand.

The paper on "Impact-Based Search in Constraint-based Scheduling" by Armin Wolf describes an efficient search strategy applied to job-shop scheduling problems. Search based on impacts applies a general purpose search strategy recently adopted from Linear Integer Programming.

The paper on "Representation and Processing of Preferential Rules" by Ulrich Geske, Hans-Joachim Goltz, and Armin Wolf presents easy-to-use techniques for applying preferential rules in case of non-availability of pre-defined procedures which deal with "soft constraints" in a constraint solver.

The paper on "BestBasket, an Application for Efficient Indirect e-Procurement in Small and Mid-size Businesses" by Georg Ringwelski deals with the problem of finding optimal suppliers in B2B marketing. The BestBasket system is used by 1100 customers each day and 35% of them accept the optimized supplier-porfolio computed by the system.

Another application problem, the dynamic programming construction of bundles within the framework of the Winner Determination Problem in Combinatorial Auctions, is dealt with in „(Co)^{4}At — *Co*mputational *Co*mplexity in *Co*nstraint-based *Co*mbinatorial *A*uc*t*ions" by Michael Thomas Egner and Walter Hower. The exponential search space is the challenge and appropriate techniques are discussed.