History, Evolution and Trends in Mathematical Modeling Tools

Susanne Heipcke[1]

FICO Xpress Optimization, FICO House, Starley Way, Birmingham B37 7GN, U.K.
susanneheipcke@fico.com

Mots-clés : *mathematical modeling software, optimization, project deployment.*

1 Introduction

This talk reviews the development of modeling languages and tools for the formulation and solving of mathematical optimization problems since the 1980s when many of today's commercial tools were first published. With the advent of more powerful processors and highspeed network connectivity the general usage patterns of OR tools have evolved over time and user expectations on modeling software include an increasingly large range of functionality.

2 Modeling languages : Historical notes

"Traditional" algebraic modeling languages developed in the 1970s or 80s like GAMS [3], AMPL [4] or Xpress Mosel's precursor mp-model [5] were designed as special-purpose, declarative languages that allowed users to state mathematical optimization problems in a close to natural way with facilities for importing data in some pre-defined format(s), but without any direct connection to solvers : the modeling tools exported the problem matrix to a file ; the optimization solver, a standalone program, read the file, solved the problem and created a solution file ; the modeling tool could then be used to generate a report from this output. This design was to a large extent conditioned by the impossibility to fit both, model formulation and the problem representation for the solver, into the memory of typical commercially available hardware.

By the early 90s hardware capacities had improved considerably, making it possible to embed optimization into other programs : optimization solvers were made accessible in the form of libraries (XOSL, Cplex), later completed with model building libaries (such as LP-Toolkit, Ilog Concert, or Xpress BCL). However, the separation into a "modeling language" and algorithmic operations written with a "scripting [1] language" is still present for OPL [6] with OPL-script in the mid-1990s. Other new modeling systems such as AIMMS [7] and MPL [8] introduced graphical interfaces for working with models.

Many approaches to solve classical Operations Research problems developed since the 90s (for example branch-and-cut algorithms) require an interaction between the modeling and solving phases. The programming language based model builders provide such a close interaction between modeling and solution algorithms, but many practitioners prefer to state their model using an algebraic modeling language. Xpress Mosel, first published in 2001, has therefore introduced the concept of a completely *integrated modeling and solving* language. It also addressed other typical expectations of modeling software users :

— representation of optimization problems in close to natural / algebraic form
— efficient handling of sparse data / conditions in constraint definition
— easy access to external data sources (spreadsheets, databases)
— support for different solver types including Constraint Programming, an emerging technology in the 90s

1. Generally interpreted, line-oriented languages with restricted programming functionality

— possibilities for deployment into existing company systems (embedding via programming language interfaces, in-memory data exchange)
— graphical development environment

3 Current trends

In the 2000s-2010s most modeling software tools have been updated to provide possibilities for interleaving modeling and solving tasks, either by combining their own scripting facilities with their modeling language, or by allowing their users to invoke other languages for programming tasks from within the model source code.

Beyond the integration of modeling and solving, in recent years new user requirements can be added to the previous list :
— support for parallel and distributed computing, including on the modeling level
— collaborative development and deployment as multi-user web apps
— interfaces for combination with other tools and use of pre-existing specialized code implemented in other languages

Many of these features relate to programming tasks : software tools for modeling are being enhanced with new features and deployment capabilities [9]. At the same time one can observe a growing interest in (programming) languages such as Python that also provide modeling capabilities, but are predominantly used by other scientific communities.

Instead of being run as standalone programs, optimization models are more and more frequently deployed as distributed, multi-user solutions within company networks or in cloud-based environments. This trend is driving enhancements to modeling tools with respect to data handling (new types of data sources, capabilities for handling very large scale instances), support of concurrent, distributed, or remote computing, the increased importance of visualization and reporting, and possibilities of user interaction including during the solving process. Taking this one step further, user expectations currently tend towards integrated applications and platforms that facilitate the unsiloed use of various technologies such as a combination of mathematical optimization with analytic tools (*e.g.* Machine Learning algorithms).

Références

[1] Kallrath, J. (ed.) : Modeling Languages in Mathematical Optimization. Kluwer Academic Publishers, Norwell (2004)

[2] Colombani, Y., Daniel, B., Heipcke, S. : Mosel : a Modular Environment for Modeling and Solving Problems. In : [1], pp. 211–238

[3] Bussieck, M. R., Meeraus, A. : General algebraic modeling system (GAMS). In : [1], pp. 137–157

[4] Fourer, R., Gay, D., Kernighan, B.W. : AMPL : A Modeling Language for Mathematical Programming. The Scientific Press, San Francisco (1993)

[5] Ashford, R.W., Daniel, R.C. : LP-MODEL : XPRESS-LP's Model Builder. IMA Journal of Mathematics in Management **1**, 163–176 (1987)

[6] Van Hentenryck, P. : The OPL Optimization Programming Language. MIT Press, Cambridge (1998)

[7] Bisshop, J., Roelofs, M. : The modeling language AIMMS. In : [1], pp. 71–104

[8] Kristjansson, B., Lee, D. : The MPL modeling system. In : [1], pp. 239–265.

[9] Heipcke, S. : Xpress-Mosel : Multi-Solver, Multi-Problem, Multi-Model, Multi-Node Modeling and Problem Solving. In : Kallrath, J. (ed.), Algebraic Modeling Systems : Modeling and Solving Real World Optimization Problems, pp. 81–114. Springer, Heidelberg (2012)