

# Constraint Programming and a Usability Quest

Laurent D. Michel

University of Connecticut  
Storrs, CT 06269-2155  
ldm@engr.uconn.edu

In 2004, Jean-Francois Puget presented [2] an analysis of the “simplicity of Use” of Constraint Programming from which he articulated a series of challenges to make Constraint Programming systems accessible and easier to use. The core of the argument was a contrast between mathematical programming and constraint programming tools. Mathematical programming adopts a *model and run* paradigm, rely on a simple vocabulary to model problems (i.e., linear constraints), support standard formats for sharing models and benefit from extensive documentation on how to model [5]. Constraint programming features a *model and search* paradigm, rich modeling languages with combinatorial objects and has a distinctive flavor of programming. While it can be construed as CP’s Achilles’ heel, it is also its most potent strength and is supported by modeling aids [3,4]. The very existence of sophisticated parameter tuning solutions for SAT solvers and Math Programming solvers to determine ideal parameters (e.g., ParamILS [1]) certainly cast a major shadow on the potency of the *model and run* mantra that is evolving into *model and search for the right parameters*.

Accessibility to CP technology is a legit concern and the appeal of turnkey solutions cannot be underestimated. CP tools are extremely pliable and uniquely adapted to classes of problems where all else fails. Retaining CP’s flexibility while delivering *model and run* solutions suitable for a large number of situations is the position adopted here. This talk explores developments and solutions to the apparent quandary. Specifically, it explores automatic search for Constraint-Based Local Search, Scheduling, and finite-domain systems, generic black-box search procedures, automatic parallelization and assisted hybridization.

## References

1. Hutter, F., Hoos, H.H., Leyton-Brown, K., Stützle, T.: Paramils: an automatic algorithm configuration framework. *J. Artif. Int. Res.* 36(1), 267–306 (2009)
2. Puget, J.-F.: Constraint Programming Next Challenge: Simplicity of Use. In: Wallace, M. (ed.) *CP 2004. LNCS*, vol. 3258, pp. 5–8. Springer, Heidelberg (2004)
3. Van Hentenryck, P.: *The OPL Optimization Programming Language*. The MIT Press, Cambridge (1999)
4. Van Hentenryck, P., Michel, L.: *Constraint-Based Local Search*. The MIT Press, Cambridge (2005)
5. Wolsey, L.A., Nemhauser, G.L.: *Integer and Combinatorial Optimization*. Wiley-Interscience (1999)