ACP EC Election: Election Statement H. Simonis

Constraint Programming (CP) is facing challenging times; our association has to react if we want to maintain an important role in the decision support software domain. Key problem areas are:

- The main competing technologies (MIP, SAT) have made great progress in problem solving, if not in the problem modelling part. CP solving has made many improvements as well, but they are not as visible and not as widespread. Other techniques like rule-based reasoning seem to be much more attractive to users.
- The CP tool area is far too fragmented. There is little commonality in the tools, and each solver has some unique features, which can't be used in other systems. This means that for solving a wide range of problems, you have to learn and master a large number of incompatible tool sets.
- Industrial use of CP is all but invisible. Current industrial users rarely participate in ACP sponsored events, and attracting new industrial users is quite difficult. We also lack convincing case studies to bring new people into our area.

At the same time, we have made progress in a number of important directions:

- High quality, open sourced tools make CP much more accessible than before. Licence costs no longer are prohibitive. For academic users, choices are improved further by free academic access to some of the commercial CP tools.
- Interesting new domains have been opened up with new constraint techniques, be it testing and verification of software or increased use of continuous domains for different problems.
- A number of systems now allow access to multiple technologies (Local search, MIP, SAT, finite domains) within the same modelling framework. More advanced systems allow to build hybrid solutions with reduced implementation effort.

I think that in order to thrive as a community, we must be able to answer a few key questions:

- We have to show that/how CP solves important problems, in order to attract new researchers, new funding and new users. As an example, it is alarming that CP does not seem to have any input in the FP7 framework program.
- We should be able to define for which problems CP is best. Of course, "best" can mean many different things: it provides the best solutions, or it is the most cost effective solution to develop, or the easiest to adapt to changes in the environment.
- What is our relation to the SAT, Integer Programming and Local Search fields?
- We should have a clearer understanding what "state of the art" means for CP, and how we can train people in its use, and which capabilities in a tool are really important.

As steps on this path I would consider the following activities:

- The training program for PhD students both in the ACP Summer Schools as well as the doctoral program at the CP conferences have been a big success. It there a way to doing something similar for attracting and training industrial users?
- One way of bringing the tool market together is by standards activities, defining a common base-line that CP solvers should support. The existing standardisation effort is struggling, can we improve the process and get more involvement from the community?

I'm already working (in a small way) on different of these topics, often together with other practitioners:

- I have developed an ELearning course on constraints (http://4c.ucc.ie/~hsimonis/ELearning/index.htm) to explain some core features of constraint programming. This material is freely available under a CC BY-SA-NC license.
- We have developed a generic interface for finite domain solvers, CP-Viz, available as open-source from http://sourceforge.net/projects/cpviz/ for multiple constraint systems.
- I'm trying to collect examples of industrial applications for CP in my application blog (http://hsimonis.wordpress.com).