

A Summary of the Panel on Applications of ASP

The Tenth International Conference on Logic Programming and Nonmonotonic Reasoning, Potsdam, Germany

Agostino Dovier¹ and Esra Erdem²

¹ Department of Mathematics and Computer Science, University of Udine, Italy

² Faculty of Engineering and Natural Sciences, Sabancı University, Turkey

As a part of the Tenth International Conference on Logic Programming and Nonmonotonic Reasoning (LPNMR'09), we arranged a panel on the “Existing Successful Applications of Answer Set Programming (ASP)”. We invited seven researchers, all known for having applied a logic-based formalism to one or more fields:

1. Marcello Balduccini (Eastman Kodak Company, USA)
2. Martin Brain (University of Bath, UK)
3. Francesco Ricca (Univ. della Calabria, IT)
4. Martin Gebser (Univ. of Potsdam, GE)
5. Francois Gagnon (Carleton Univ., CA)
6. Alessandra Mileo (Univ. of Milano–Bicocca, IT)
7. Alexander Bockmayr (Freie Universität Berlin, GE)

Each invited panelist reported his/her experience based on the following questions:

1. How did you learn about the application?
2. Why did you decide to use ASP?
3. How does the ASP-approach compare with other approaches?
4. What are the challenges you have faced in your ASP application?
5. How and to what extent could you handle these challenges?
6. Do you consider your application successful?

where ASP was replaced by LP/CP/SAT in some cases.

Marcello Balduccini shared with us his experience on using ASP in a project about the printing workflow, at the Intelligent Systems Department of the Eastman Kodak Company. The printing workflow consists of steps that turn a manuscript specification into an actual manuscript, and involves many decisions that can be grouped in two parts: Production Planning (Which sheet size, signatures, folds? Which press/folder/cutter/binder model?) and Scheduling (When to start printing? Which press? When to start folding/cutting/binder with which folder/cutter/binder? How late will we be? Can we afford to be late?). The printing involves many devices often correlated each other, and modeling such complex devices is very hard. Furthermore, reasoning over such systems (e.g., splitting tasks over devices) requires the ability of handling numbers, and commonsense background knowledge. Due to these requirements of the problem, Marcello decided to use ASP (in particular, CR Prolog due to possible “soft constraints”) in connection with CP, for decisions about the printing devices; he implemented a prototype system (based on a CR Prolog program) which is being used by the development team. Marcello’s ASP&CP approach proved useful compared to the earlier methods that used Genetic Programming.

Martin Brain briefly reported on three applications he (with other members of the KR&R Group, led by Marina de Vos) has been working on at the University of Bath: TOAST (a tool for superoptimization), ANTON (a tool for automatic music composition), and DIORAMA (a tool that aims to create 3D maps both strategically and visually

appealing). Later he focused on the last application. Martin and his colleagues came up with the last application while playing Warzone 2100, when they got bored of playing with the same set of maps: Using ASP, they built a tool for generating new maps. Also thanks to programmer's experience, ASP seemed to be the best suited tool for some parts of the task (e.g. bi-reachability checking). "Basically ASP was there", Martin said. The final results are graphically very impressive.³ In the end, Martin has emphasized that examples people can understand (e.g. ANTON compositions that people can listen to, or new maps generated by DIORAMA that people can see graphically and play with) is a good way to communicate what we are doing. He has also pointed out the main problems he has faced while working on these applications: the need for faster solvers (especially multi-threaded clause learning), standardization, and a pre-grounding code improvement.

Francois Gagnon discovered ASP in a graduate course on knowledge representation, taught by Leo Bertossi, at Carleton University, and then he tried to solve the Operating System discovery problem with ASP. He found the problem representation very natural due to presence of disjunctions, and the description of the solutions by sets very intuitive. In this application, the ASP-based approach provided him quick prototyping. He pointed out two challenges in this application: large computation times, and lack of debugging tools.

Martin Gebser first summarized two groups of applications he (with other members of the KR&R Group, led by Torsten Schaub) has been working on at the University of Potsdam: a series of applications in System Biology (e.g., consistency checking, diagnosis, network repairs) and a multi-criteria optimization problem in Hardware Synthesis (i.e., scheduling task execution and intercommunication such that resources are not exceeded, and the response times, switching costs, communication latencies are minimal). Martin told us that people can be interested in solving their problems with our tools: "A problem may be down the hall, on your colleague's desk. To get in touch, we need catchy toy examples to show them!", Martin said. On the other hand, to understand our approaches people need to familiarize with declarative thinking, and to understand their problems we need to familiarize with the concepts related to their problems; in such cases, "a man in-the-middle is priceless". In the end, it all comes to skills and fortune that the right idea and people come together.

Francesco Ricca reported several commercial applications of DLV, such as Team Building in the Gioia-Tauro Seaport, E-Tourism: The IDUM System, Automatic Itinerary Search, e-Government, and e-Medicine. His team at University of Calabria, led by Nicola Leone, learned about these interesting applications via their collaboration with Exeura s.r.l—a company born as a spin-off of University of Calabria specialized in technology transfer (e.g., finding problems and then proposing solutions to companies). They decided to use ASP for these applications for the following reasons: natural modeling with ASP constructs, reasoning on complex knowledge, and need for fast prototyping. Francesco told us that interaction with other programming languages (e.g. data types, dates, floating-point) and debugging were challenging aspects of applying ASP to these problems; they handled these challenges by building application-oriented tools (e.g., APIs) and applying ad hoc solutions (e.g., dealing with dates through special external functions). When we asked Francesco whether he considered these applications successful, he said "Yes! They are employed in practice, and we get positive feedback from the users."

Alessandra Mileo told us about the SINDI system⁴ (developed at NOMADIS Lab, University of Milan Bicocca), where ASP is used to reason about data collected from

³ <http://www.warzone2100.org.uk/>

⁴ <http://www.nomadis.unimib.it>

sensors (in a wireless sensor networks) and support decisions for independent living. ASP is well-suited for this application not only due to its expressive and modular formalism (to represent commonsense knowledge and to integrate multiple heterogeneous sources of knowledge about health evolution) but also due to its powerful inference even in the case of incomplete information. The challenges she faced in this application are as follows: transforming the low-level data to high-level knowledge representation, encoding background knowledge which requires close interaction with geriatrics, and updating/maintaining the knowledge. Although these challenges can be handled to some extent, Alessandra pointed out the need for tools to facilitate input-output dataflow between ASP and other systems, and the need for an appropriate tool/language for the clinicians to update/maintain the knowledge.

Finally, Alexander Bockmayr shared with us his thoughts on the use of logical approaches in real world applications, in particular, molecular biology. His main message is that a logic-based representation of a system allows us to reason about its behavior (e.g., with model checking techniques), instead of simply simulating some possible evolutions based on differential equations. Of course a logical approach involves more mathematics and biologists typically do not like it, but we can do that!

As we have briefly described above, and observed at the other application-oriented events at LPNMR'09, ASP has been used to solve combinatorial search problems on the one hand, and as a prototyping language on the other hand. It has been applied to many areas as disparate as independent living, music composition, printing workflows, systems biology, hardware synthesis, games, e-tourism, e-government, genetics. Some of these applications are deployed, some even commercialized.

Let us conclude with a summary of the thoughts/remarks of the panelists:

- Applications are crucial to ASP: to find funding, and to attract the attention of others to our work.
- Applications can emerge everywhere: open your ears, and talk to others.
- We can explain the usefulness of our work: find easy-to-understand “toy examples” that can capture the attention of others.
- In addition to solving combinatorial search problems, ASP provides an excellent tool for prototyping.
- Real-world applications bring challenges with them and necessitate further improvements: debugging tools, tools for interacting with other systems, improvements of the language (e.g., with data structures) to facilitate interaction with various programming languages, faster solvers (especially with multi-threaded clause learning), a pre-grounding code improvement, etc.

In our experiences, the most challenging part of applications in a different area (e.g., biology or medicine) is communicating with the experts in that area (e.g., biologists, medical doctors). However, the applications address so important problems that we should keep trying patiently: developing a small prototype in a short amount of time can make a big difference in the long term. Also, working on challenging applications is fun!