Introduction to the TPLP Special Issue on User-oriented Logic Programming and Reasoning Paradigms

STEFAN ELLMAUTHALER
Leipzig University, Leipzig, Germany
(*e-mail:* ellmauthaler@informatik.uni-leipzig.de)

CLAUDIA SCHULZ
Ubiquitous Knowledge Processing (UKP) Lab,
TU Darmstadt, Darmstadt, Germany
(*e-mail:* schulz@ukp.informatik.tu-darmstadt.de)

With the rise of machine learning, and more recently the overwhelming interest in deep learning, knowledge representation and reasoning (KRR) approaches struggle to maintain their position within the wider Artificial Intelligence (AI) community. Often considered as part of the *good old-fashioned AI* (Haugeland 1985) – like a memory of glorious old days that have come to an end – many consider KRR as no longer applicable (on its own) to the problems faced by AI today (Blackwell 2015; Garnelo *et al.* 2016). What they see are logical languages with symbols incomprehensible by most, inference mechanisms that even experts have difficulties tracing and debugging, and the incapability to process unstructured data like text.

Is KRR indeed a mere memory of good old days long gone? Not at all! It is as topical and applicable to today's problems as ever, as illustrated, for example, by the 2016 Workshop on Applications of Logic Programming[1] and the papers by Shoham (2015) and by Baral and Giacomo (2015). How then does the misconception arise?

As common with misconceptions, the main reason is likely to be a lack of understanding (Committee on Undergraduate Science Education 1997; Alkhalifa 2006). Declarative thinking and reasoning, as used in logic programming (LP) paradigms, are not commonly taught as part of computer science degrees. In addition, there is a lack of applications that aid novices to learn about KRR approaches, to understand the inference mechanisms in-depth, and to comprehend their applicability to complex real-world problems. On the other hand, even development tools aimed at KRR experts are sparse and not always user-friendly.

Consider, for example, a small logic program, representing our reasoning about the weather:

$$badWeather \; :- \; clouds.$$

$$GoodWeather \; :- \; not \; clouds.$$

[1] http://applp.cs.stonybrook.edu/.

Feeding this into the clingo answer set solver (Gebser *et al.* 2018; Gebser *et al.* 2014) returns the following error message:

> -:2:13-15: error: syntax error, unexpected :-
> *** ERROR: (clingo): parsing failed
> UNKNOWN

Even for an expert in answer set programming (ASP), this error message is rather cryptic. It appears to suggest that there is a problem with the inference symbol :−. However, the real reason for the error is the capitalization of *GoodWeather*. This indicates to clingo that *GoodWeather* is a variable rather than an atom, causing the occurrence of an unrestricted variable. Since every variable needs to have a defined domain to be instantiated in a finite setting, which is not the case for *GoodWeather*, this variable cannot be grounded. Conventions like this are formally necessary and not difficult to understand. However, the lacking user-friendliness of tools makes such conventions, and more generally the development of logic programs, seem complicated. It thus constitutes a hurdle for novices to adopt LP paradigms.

On the bright side, various directions of research have emerged that focus on making LP and reasoning paradigms more user-friendly, both for novice and expert users. In addition to tutorials and educational material (Riguzzi and Cota 2016; Vosinakis *et al.* 2016; Gebser *et al.* 2012), various integrated development environments (IDEs) have emerged (Busoniu *et al.* 2013; Febbraro *et al.* 2011), which facilitate programming with logic paradigms. Furthermore, approaches for visualizing logic programs (Kloimüllner *et al.* 2011; Febbraro *et al.* 2010) and explaining their solutions (Schulz and Toni 2016; Cabalar *et al.* 2014), as well as methods for debugging inconsistent logic programs (Oetsch *et al.* 2018; Alviano *et al.* 2015; Shchekotykhin 2015) have been developed to improve the user experience of both novices and experts. These recent efforts concerning the user-friendliness of programming with logic paradigms are further strengthened by a bi-annual "International Workshop on User-Oriented Logic Paradigms" (IULP), which started in 2015.[2]

This special issue aims to highlight the importance of LP research that keeps the user in mind and to present some of the directions such research can take. First, gaining an understanding of the solutions of logic programs, or the lack thereof, is key to solving problems with LP paradigms. This is facilitated by explanation and debugging approaches. Second, LP education is a crucial topic, since well-designed tutorials as well as teaching declarative thinking increase the understanding of LP by novices and can thus grow the user base. Third, IDEs make the development of logic programs easier, lowering the hurdle for the application of LP to solve a given problem.

Following an open call and invitations to submit extended versions of IULP workshop papers, six papers were selected for publication in this special issue. The accepted papers represent the different aforementioned topics, aiming at more user-friendly environments for applying logic paradigms to solve problems.

The special issue opens with a survey paper (Fandinno and Schulz 2018) on explanation approaches for the popular logic paradigm ASP. These can be divided into justification

---

[2] See http://iulp2015.uni-leipzig.de/ and http://iulp2017.uni-leipzig.de/.

and debugging methods. Justifications explain why an answer set is a solution to a given logic program. In contrast, debugging methods identify why a logic program is inconsistent and thus has no answer sets, that is, they identify errors in the way a problem is encoded as a logic program. Explanation approaches can be useful for both novice and expert users in understanding LP solutions.

Focusing further on novices, the following papers deal with ways of teaching logic paradigms. The second paper of the special issue (Yuen *et al.* 2018) asserts that LP is a viable paradigm to be taught to high school students as an introduction to computer science. The authors show that students engage with LP in similar ways as with other computer science paradigms, gaining similar skills: problem abstraction, representation of ideas and concepts in terms of code, and general reasoning. One hurdle for novices to logic paradigms is the installation and setup of systems to compute solutions to problems represented in such logic paradigms. The third paper (Wielemaker *et al.* 2018) tackles this issue by introducing SWISH – a web front end for Prolog. It also has a collaborative component, allowing users to work on joint programs. The authors furthermore present two ways in which SWISH has been applied to provide tutorials to logic paradigms built on top of Prolog. The fourth paper (Marcopoulos and Zhang 2018) also addressed the system setup hurdle by introducing ONLINESPARC, a web application for ASP. It allows users to save their logic programs in the application and come back to them later as well as to perform various useful ASP tasks: in addition to computing answer sets, users can query the logic program and produce visualizations of problem solutions (i.e., of answer sets). The authors also report on how they use ONLINESPARC for teaching LP to students at high school and university level.

For both novices and experts in LP, it is often cumbersome to find errors in a logic program that cause the program to be inconsistent or lead to unintended solutions. A novel debugging approach for the interactive identification and localization of bugs in ASP is introduced by the fifth paper (Dodaro *et al.* 2018). The new debugging system asks the developer a set of questions on expected answer sets, which is used to narrow down the set of rules likely to cause the error. The authors also show that the debugging is not affected by costly blowups during the grounding of logic programs, which is a novelty. On top of this, the new debugging approach along with a user-friendly graphical interface is integrated into the ASPIDE IDE for ASP.

Finally, the sixth paper (Rodosthenous and Michael 2018) presents the WEB-STAR web platform for story understanding, which is based on argumentation theory as a logic paradigm: a story along with background knowledge is represented in the logical form of argumentation and argumentation semantics are used to understand the story in more detail, that is, to draw commonsense inferences about the story. WEB-STAR also allows for integration with other systems that rely on story understanding. In addition, it has "social" features, such as a repository for public story sharing, commenting functions, and collaborative story editing. Overall, this approach can be used for collaborative development of storylines as well as educational purposes.

In conclusion, the contributions of this special issue all aim at providing a better experience for users and therefore making logic paradigms easier to use and more accessible.

References

ALKHALIFA, E. M. 2006. Effects of learner misconceptions on learning. In *Proceedings of the IADIS International Conference Cognition and Exploratory Learning in Digital Age*, International Association for Development of the Information Society (IADIS), 123–128.

ALVIANO, M., DODARO, C., LEONE, N. AND RICCA, F. 2015. Advances in WASP. In *Proceedings of the 13th International Conference on Logic Programming and Nonmonotonic Reasoning (LPNMR'15)*, Springer International Publishing, 40–54.

BARAL, C. AND GIACOMO, G. D. 2015. Knowledge representation and reasoning: What's hot. In *Proceedings of the Twenty-Ninth AAAI Conference on Artificial Intelligence*. AAAI Press, 4316–4317.

BLACKWELL, A. F. 2015. Interacting with an inferred world: The challenge of machine learning for humane computer interaction. In *Proceedings of the Fifth Decennial Aarhus Conference on Critical Alternatives (CA'15)*, 169–180.

BUSONIU, P.-A., OETSCH, J., PÜHRER, J., SKOCOVSKY, P. AND TOMPITS, H. 2013. SeaLion: An eclipse-based IDE for answer-set programming with advanced debugging support. *Theory and Practice of Logic Programming 13,* 4–5, 657–673.

CABALAR, P., FANDINNO, J. AND FINK, M. 2014. Causal graph justifications of logic programs. *Theory and Practice of Logic Programming 14,* 4–5, 603–618.

COMMITTEE ON UNDERGRADUATE SCIENCE EDUCATION 1997. *Science Teaching Reconsidered: A Handbook*. National Academy Press, Washington, DC.

DODARO, C., GASTEIGER, P., REALE, K., RICCA, F. AND SCHEKOTHIHIN, K. 2018. Debugging non-ground ASP programs: Technique and graphical tool. *Theory and Practice of Logic Programming (Special Issue on User-Oriented Logic Programming and Reasoning Paradigms)*.

FANDINNO, J. AND SCHULZ, C. 2018. Answering the "why" in answer set programming - A survey of explanation approaches. *Theory and Practice of Logic Programming (Special Issue on User-Oriented Logic Programming and Reasoning Paradigms)*.

FEBBRARO, O., REALE, K. AND RICCA, F. 2010. A visual interface for drawing ASP programs. In *Proceedings of the 25th Italian Conference on Computational Logic (CILC'10)*, CEUR-WS.org.

FEBBRARO, O., REALE, K. AND RICCA, F. 2011. ASPIDE: Integrated development environment for answer set programming. In *Proceedings of the 11th International Conference on Logic Programming and Nonmonotonic Reasoning (LPNMR'11)*, Springer-Verlag Berlin Heidelberg, 317–330.

GARNELO, M., ARULKUMARAN, K. AND SHANAHAN, M. 2016. Towards deep symbolic reinforcement learning. *CoRR abs/1609.05518*.

GEBSER, M., KAMINSKI, R., KAUFMANN, B., LÜHNE, P., OBERMEIER, P., OSTROWSKI, M., ROMERO, J., SCHAUB, T., SCHELLHORN, S. AND WANKO, P. 2018. The potsdam answer set solving collection 5.0. *KI - Künstliche Intelligenz 32,* 2, 181–182.

GEBSER, M., KAMINSKI, R., KAUFMANN, B. AND SCHAUB, T. 2012. *Answer Set Solving in Practice*. Morgan & Claypool Publishers.

GEBSER, M., KAMINSKI, R., KAUFMANN, B. AND SCHAUB, T. 2014. Clingo = ASP + control: Preliminary report. *CoRR abs/1405.3694*.

HAUGELAND, J. 1985. *Artificial Intelligence: The Very Idea*. Massachusetts Institute of Technology, Cambridge, MA, USA.

KLOIMÜLLNER, C., OETSCH, J., PÜHRER, J. AND TOMPITS, H. 2011. Kara: A system for visualising and visual editing of interpretations for answer-set programs. In *Revised Selected Papers of the 19th International Conference on Applications of Declarative Programming and Knowledge Management (INAP'11) and the 25th Workshop on Logic Programming (WLP'11)*, 325–344.

MARCOPOULOS, E. AND ZHANG, Y. 2018. onlineSPARC: A programming environment for answer set programming. *Theory and Practice of Logic Programming (Special Issue on User-Oriented Logic Programming and Reasoning Paradigms)*.

OETSCH, J., PÜHRER, J. AND TOMPITS, H. 2018. Stepwise debugging of answer-set programs. *Theory and Practice of Logic Programming 18,* 1, 30–80.

RIGUZZI, F. AND COTA, G. 2016. Probabilistic logic programming tutorial. *The Association for Logic Programming Newsletter 29,* 1, 1–1.

RODOSTHENOUS, C. T. AND MICHAEL, L. 2018. Web-star: A visual web-based IDE for a story comprehension system. *Theory and Practice of Logic Programming (Special Issue on User-Oriented Logic Programming and Reasoning Paradigms)*.

SCHULZ, C. AND TONI, F. 2016. Justifying answer sets using argumentation. *Theory and Practice of Logic Programming 16,* 01, 59–110.

SHCHEKOTYKHIN, K. M. 2015. Interactive query-based debugging of ASP programs. In *Proceedings of the 29th AAAI Conference on Artificial Intelligence (AAAI'15)*, AAAI Press, 1597–1603.

SHOHAM, Y. 2015. Why knowledge representation matters. *Communications of the ACM 59,* 1, 47–49.

VOSINAKIS, S., ANASTASSAKIS, G. AND KOUTSABASIS, P. 2016. Teaching and learning logic programming in virtual worlds using interactive microworld representations. *British Journal of Educational Technology 49,* 1, 30–44.

WIELEMAKER, J., RIGUZZI, F., KOWALSKI, B., LAGER, T., SADRI, F. AND CALEJO, M. 2018. Using swish to realise interactive web based tutorials for logic based languages. *Theory and Practice of Logic Programming (Special Issue on User-Oriented Logic Programming and Reasoning Paradigms)*.

YUEN, T. T., REYES, M. AND ZHANG, Y. 2018. Introducing computer science to high school students through logic programming. *Theory and Practice of Logic Programming (Special Issue on User-Oriented Logic Programming and Reasoning Paradigms)*.