The Scientific Contribution of Marek Sergot

Steve Barker[1], Andrew J.I. Jones[1], Antonis Kakas[2], Robert A. Kowalski[3], Alessio Lomuscio[3], Rob Miller[4], Stephen Muggleton[3], and Giovanni Sartor[5]

[1] Department of Informatics, King's College London, UK
[2] Department of Computer Science, University of Cyprus
[3] Department of Computing, Imperial College London, UK
[4] Department of Information Studies, University College London, UK
[5] CIRSFID, University of Bologna, Italy

Marek Sergot's technical contributions range over different subjects. He has developed a series of novel ideas and formal methods bridging different research domains, such as artificial intelligence, computational logic, philosophical logic, legal theory, artificial intelligence and law, multi-agent systems and bioinformatics.

By combining his background in logic and computing with his interest in the law, deontic logic, action, and related areas, and applying to all his capacity to understand the subtleties of social interactions and normative reasoning, Marek has been able to open new directions of research, and has been a reference, an inspiration, and a model for many researchers in the many fields in which he has worked.

1 Early Work in Logic Programming

Marek's early research focused on logic programming, deductive databases and legal reasoning. This led to his development of the query-the-user extension of logic programming [31], in which the user provides information during the execution of a logic program, if and when the program requires it. With query-the-user, interaction between the computer and the user is symmetric — each can ask questions and provide answers to the other.

Marek collaborated with Peter Hammond to augment Prolog with query-the-user and explanation facilities, developing the expert system shell APES [15], which was marketed by their small company, Logic Based Systems Ltd. He collaborated in many applications of APES, including the implementation of the British Nationality Act [36] and the Indian central government pension rules [39]. APES was also used to develop GLIMPSE [42], a front-end for the statistics package GLIM. To address some of the limitations of GLIMPSE, Marek and Kostas Stathis developed an alternative model of computer interaction viewed in terms of games [40]. Marek also made important contributions to the theory of logic programming in his work with Dov Gabbay on negation as inconsistency [13].

Although Marek later turned his attention to many other areas of logic and computation, he never abandoned his roots in logic programming. In recent years, he has built upon logic programming in such areas as activity recognition [5] with Alexander Artikis and argumentation [41] with Francesca Toni.

A. Artikis et al. (Eds.): Sergot Festschrift, LNAI 7360, pp. 4–11, 2012.

2 Temporal Reasoning and Action Languages

In 1986 Marek and Bob Kowalski produced a seminal contribution to artificial intelligence with their proposal of the Event Calculus for temporal reasoning [23]. This framework, naturally realizable as a logic program, provided an alternative to the Situation Calculus that arguably avoided some aspects of the frame problem, and allowed a more straightforward representation of dynamic domain features such as simultaneous and partially ordered events. Over the years the Event Calculus has become a standard formalism for reasoning about actions and change, and is often used as the foundational "database layer" for artificial intelligence applications. Its importance is illustrated by the fact that many researchers from all over the world have worked in clarifying and refining its basic concepts and continue to this date on its further development.

Marek, in his true pioneering spirit, has since been interested in both practical and theoretical frameworks for temporal reasoning. With Nihan Kesim they developed a logic programming framework for modelling temporal objects [22], motivated by problems of schema evolution and versioning of objects in deductive databases. More recently, he has looked at the larger question for action theories of what brings about actions and how this is regulated in a multi-agent system. For example, together with Rob Craven he has developed an extension of the action Language $\mathcal{C}+$ [14] called $n\mathcal{C}+$ [35,9,34] that combines action, agency and normative systems. This work shows how formalisms can be developed that are applicable in realistic multi-agent systems where the actions to be performed by the agents are governed by norms such as agent permissions, obligations and prohibitions, and other normative relations between agents.

3 Artificial Intelligence and Law

We can distinguish two main directions of Marek's initial contributions to artificial intelligence and law: on the one hand he has provided a theoretical and conceptual background for representing laws as logic programs [37], and on the other hand he has stimulated the use of logic in the development of knowledge-based systems in the legal domain. In particular, the paper entitled "The British Nationality Act as a Logic Program" [36] was hugely influential in the development of artificial intelligence and law. This paper has defined the paradigm of a declarative and isomorphic representation of legal knowledge, to be achieved by modelling legislation as an axiomatic theory (in Prolog), while delegating inference to the corresponding theorem prover. Moreover it has provided clues for future research, anticipating various attempts to provide richer logical frameworks for legal reasoning. In particular, it includes a discussion on the advantages and limitations of negation as failure, and provides pointers to nonmontonic reasoning in the law (later to be addressed in particular through defeasible argumentation), as well as developments such as the coupling of negation by failure and classical negation in logic programs. The paper also addresses the treatment of counterfactual conditionals within legal norms, an issue still to be adequately

addressed within legal logic. Marek, in collaboration with Robert Kowalski and others, has published a number of further influential contributions on logic programming and the law, where theoretical foundations were discussed (see, for instance, [24] and [7]) and various applications were presented (see, for instance, [8], [39], [11]). This research had a pervasive impact on artificial intelligence and law. It laid the foundation for the use of computational logic in the legal domain, and provided the inspiration for some successful knowledge-based systems [12].

Marek's contributions to artificial intelligence and law are not limited to the legal applications of logic programming strictly understood. He has indeed viewed the relationship between law and computing as a two way learning process: not only the application of the law can be supported by computerised tools and lawyers can use (computational) logic for analysing legal contents, but also computing can learn from the ways in which the law structures normative knowledge and governs social systems. This view is expressed in a number of technical contributions that also address fundamental aspects of the law, such as normative systems [18], normative positions involving a plurality of agents [38], contrary to duty obligations [30,29], legal-institutional powers (in particular [1] and [27]). An important domain for Marek's analysis of powers and institutions concerns the definition of open norm-governed agent systems, and the idea that is developed in [3,4,2], where the analysis of powers and institutions is often complemented with the causal logic of [14,33].

4 Deontic Logic and Norm-Governed Systems

Marek's work in deontic logic, and in the broader area of the theory of norm-governed systems, began with his collaboration with Andrew J. I. Jones in the 1990's. Their first paper took up issues regarding the potential role of deontic logic in the representation of legal knowledge [19], which in part led to their interest in the theory of normative positions, which they developed in the tradition deriving from the Kanger-Lindahl formal characterisations of the Hohfeldian rights-relations [20]. (See also [18].) Marek later generalised the Kanger-Lindahl theory, and developed methods for its automation and practical application, including implementation in the computer program Norman-G [32]. His collaboration with Jones culminated in their widely-cited paper on the formal characterisation of institutionalised power [21], which provided the first modal-logical analysis of 'counts-as' conditionals: conditionals of the form 'A counts as B in institution X'.

Another issue discussed in [19] concerned so-called 'contrary-to-duty' conditionals (CTDs): conditionals that describe those obligations that come into force when some other, more primary obligation has been violated. It has long been recognised that CTDs constitute a central challenge for Standard Deontic Logic. In joint work with Henry Prakken [30,29], Marek attempted to address the analysis of CTDs, and produced a set of benchmark examples of problematic scenarios in which CTDs play a prominent part.

The theme of norm-violation also figured prominently in research Marek carried out with Alessio Lomuscio (see, in particular, [25,26]). The focus there was

on the development of the formal machinery of deontic interpreted systems, and its application to the analysis of agents' behaviour, both when it conforms to norm, and when it fails to conform — either because of failure to do what was supposed to be done, or because something was done that is not permitted. Variations of the bit-transmission problem were used to illustrate the analyses.

In some of his more recent work, in part carried out in collaboration with Rob Craven, with Alexander Artikis, and with Jeremy Pitt, Marek has focused on formal-logical theories of action and agency, and on the development of computational frameworks for norm-governed open agent societies — see, for instance, the research reported in [35,4,34]. These are issues of central importance to current work in the field of multi-agent systems.

5 Logical Approaches to Policies and Authorization

Marek's work on logic applied to aspects of computer security reflects the wide-ranging nature of his contributions to other branches of computer science. His work on security ranges over advanced forms of novel authorisation frameworks, calculi for specifying policy administration requirements (for delegation, for example) and frameworks that go beyond "traditional" requirements and approaches (for instance, to consider empowerment in institutional contexts, and its relation to permission). Moreover, his contributions to the computer science literature (notably the Event Calculus), more generally, have often resulted in the exploitation of these ideas in specific computer security contexts.

In early work, Marek sketches out a rich access control framework that addresses traditional concerns about the effective representation of core security concepts like permissions and prohibitions but also highlights the importance of obligations in practical contexts. Later work with Jones (see, for example, [21]) was to result in a description of a rich framework of access control in which the powers that agents might exercise (often in an institutional context) were a key point of focus for security researchers: that institutional powers are distinct from the notion of permission. For example, a priest may be empowered to marry a couple but not be permitted to do so. The notion of empowerment has a number of important applications in access control. For instance, in work with Sadighi, Marek applied the concept of ability to override in the context of distributed policy administration to generate yet another access control model, the *privilege calculus*. The shortcomings of existing access control approaches (for novel forms of virtual organisational structure) are also considered by Marek and Sadighi in the context of *contractual access control* in which the notion of *entitlement* is used to refer to a strong form of permission. Some of Marek's work (for example, that on the notion of empowerment and that on the privilege calculus) is more obviously directed towards computer security than others. However, it is worth noting that his work has had wide-ranging impact beyond the scope for which the work was perhaps originally intended. This observation is supported when considering the Event Calculus in relation to research on computer security. Specifically, various temporal security systems have been described in the

literature: the work by Craven et al. [10] on obligations has been influenced by the Event Calculus and a novel form of access control model, *status-based access control* [6], has been influenced by the Event Calculus.

6 Bioinformatics

Marek has made a number of important contributions in the demonstration and application of computational techniques to biological modelling problems. In particular, in [28] it was demonstrated that the abductive logic programming provided a powerful framework for interpreting high-throughput data from biological experiments. The input data consisted of regulation patterns in microarray data, which were used to generate candidate gene interactions which explained the observations.

In later work [17] Marek showed that another logic-based artificial intelligence reasoning technique, that of argumentation, also provided a powerful tool for reasoning about alternative interpretations of biological data. In this case, argumentation was used to represent expert reasoning within the context of 3D-PSSM analysis of protein structure. Increased accuracy was demonstrated and the technique was made publicly available on a server.

Lastly, Marek contributed to the development of the SEAN system [16], which predicts single necleotide polymorphisms (SNPs) based on expressed sequence tags (ESTs). The algorithm uses SNP abundance and sequence identity to make its predictions. SEAN provides a Java viewer which supports presentation of the results.

7 Conclusions

Above we have surveyed some of Marek's most widely known contributions. They form a small fragment of his academic output. One of Marek's characteristics is his very high standards. He is known to drop ideas and draft papers that do not meet his utmost expectations. This results in a large amount of material which still remains unpublished and was not reported here. These are not half-baked ideas that have not withstood the test of rigorous exploration. Often they are essentially finished papers that perhaps require, in his view, a formal connection with other, sometimes obscure, works in the literature, or an alternative implementation to better validate experimental results. Other authors of lesser standards would have published these years ago. Somewhat ironically, some of these ideas do find their way into published literature as he often presents unpublished material and ideas are borrowed by other authors. Inevitably, they often lose much of their original appeal.

Marek remains an inspiration to many colleagues. He is known for building his own PC, compiling his Linux kernel, hacking compilers and SAT-solvers, releasing code, and at the same time focusing on formal concepts and making them relevant to today's latest developments. He is a true Computer Scientist in the fullest sense of the word.

It is the breath and depth of his thinking that we celebrate in this volume.

References

1. Allen, L.E.: Towards a normalized language to clarify the structure of legal discourse. In: Deontic Logic, Computational Linguistics and Legal Information Systems, vol. II, pp. 349–407. North Holland, Amsterdam (1982)
2. Artikis, A., Sergot, M.J.: Executable specification of open multi-agent systems. Logic Journal of the IGPL 18(1), 31–65 (2010)
3. Artikis, A., Sergot, M.J., Pitt, J.: An executable specification of a formal argumentation protocol. Artificial Intelligence 171(10-15), 776–804 (2007)
4. Artikis, A., Sergot, M.J., Pitt, J.: Specifying norm-governed computational societies. ACM Transactions on Computational Logic 10(1) (2009)
5. Artikis, A., Sergot, M.J., Paliouras, G.: A logic programming approach to activity recognition. In: Proceedings of the 2nd ACM International Workshop on Events in Multimedia, EiMM 2010, pp. 3–8. ACM, New York (2010)
6. Barker, S., Sergot, M.J., Wijesekera, D.: Status-based access control. ACM Trans. Inf. Syst. Secur. 12(1) (2008)
7. Bench-Capon, T.J.M., Sergot, M.J.: Towards a rule-based representation of open texture in law. In: Walter, C. (ed.) Computing Power and Legal Language, pp. 39–60. Greenwood/Quorum Press (1988)
8. Bench-Capon, T.J.M., Robinson, G.O., Routen, T., Sergot, M.J.: Logic programming for large scale applications in law: A formalisation of supplementary benefit legislation. In: ICAIL, pp. 190–198 (1987)
9. Craven, R., Sergot, M.J.: Agent strands in the action language $nC+$. Journal of Applied Logic 6(2), 172–191 (2008)
10. Craven, R., Lobo, J., Ma, J., Russo, A., Lupu, E.C., Bandara, A.K.: Expressive policy analysis with enhanced system dynamicity. In: ASIACCS, pp. 239–250 (2009)
11. Daskalopulu, A., Sergot, M.J.: A constraint-driven system for contract assembly. In: ICAIL, pp. 62–70 (1995)
12. Dayal, S., Johnson, P.: A web-based revolution in australian public administration. Journal of Information, Law and Technology 1 (2000)
13. Gabbay, D.M., Sergot, M.J.: Negation as inconsistency i. J. Log. Program. 3(1), 1–35 (1986)
14. Giunchiglia, E., Lee, J., Lifschitz, V., McCain, N., Turner, H.: Nonmonotonic causal theories. Artificial Intelligence 153(1-2), 49–104 (2004)
15. Hammond, P., Sergot, M.J.: A PROLOG shell for logic based expert systems. In: Proc. BCS Conference on Expert Systems 1983, pp. 95–104. British Computer Society (1983)
16. Huntley, D., Baldo, A., Johri, S., Sergot, M.J.: Sean: Snp prediction and display program utilizing est sequence clusters. Bioinformatics 22(4), 495–496 (2006)
17. Jefferys, B.R., Kelley, L.A., Sergot, M.J., Fox, J., Sternberg, M.J.E.: Capturing expert knowledge with argumentation: a case study in bioinformatics. Bioinformatics 22(8), 924–933 (2006)
18. Jones, A., Sergot, M.J.: On the characterisation of law and computer systems: the normative systems perspective. In: Deontic Logic in Computer Science: Normative System Specification, pp. 275–307. J. Wiley and Sons (1993)
19. Jones, A.J.I., Sergot, M.J.: Deontic logic in the representation of law: Towards a methodology. Artificial Intelligence and Law 1, 45–64 (1992), doi:10.1007/BF00118478

20. Jones, A.J.I., Sergot, M.J.: Formal Specification of Security Requirements using the Theory of Normative Positions. In: Deswarte, Y., Quisquater, J.-J., Eizenberg, G. (eds.) ESORICS 1992. LNCS, vol. 648, pp. 103–121. Springer, Heidelberg (1992)
21. Jones, A.J.I., Sergot, M.J.: A formal characterisation of institutionalised power. Logic Journal of the IGPL 4(3), 427–443 (1996)
22. Nihan Kesim, F., Sergot, M.J.: A logic programming framework for modeling temporal objects. IEEE Trans. Knowl. Data Eng. 8(5), 724–741 (1996)
23. Kowalski, R., Sergot, M.J.: A logic-based calculus of events. New Generation Computing 4(1), 67–96 (1986)
24. Kowalski, R., Sergot, M.J.: The use of logical models in legal problem solving. Ratio Juris 3, 201–218 (1990)
25. Lomuscio, A., Sergot, M.J.: Deontic interpreted systems. Studia Logica 75(1), 63–92 (2003)
26. Lomuscio, A., Sergot, M.J.: A formulation of violation, error recovery, and enforcement in the bit transmission problem. Journal of Applied Logic 2(1), 93–116 (2003)
27. Makinson, D.: On the formal representation of rights relations. Journal of Philosophical Logic 15, 403–425 (1986)
28. Papatheodorou, I., Kakas, A.C., Sergot, M.J.: Inference of Gene Relations from Microarray Data by Abduction. In: Baral, C., Greco, G., Leone, N., Terracina, G. (eds.) LPNMR 2005. LNCS (LNAI), vol. 3662, pp. 389–393. Springer, Heidelberg (2005)
29. Prakken, H., Sergot, M.J.: Contrary-to-duty obligations. Studia Logica 57(1), 91–115 (1996)
30. Prakken, H., Sergot, M.J.: Dyadic deontic logics and contrary-to-duty obligations. In: Defeasible Deontic Logic. Kluwer Academic Publishers (1997)
31. Sergot, M.J.: A query-the-user facility for logic programs. In: Yazdani, M. (ed.) New Horizons in Educational Computing. Ellis Horwood, Chichester (1984)
32. Sergot, M.J.: A computational theory of normative positions. ACM Transactions on Computational Logic 2(4), 522–581 (2001)
33. Sergot, M.J.: $(\mathcal{C}+)^{++}$: An action language for modelling norms and institutions. In: Technical Report 2004/8, Department of Computing, Imperial College London (2004)
34. Sergot, M.J.: Action and Agency in Norm-Governed Multi-Agent Systems. In: Artikis, A., O'Hare, G.M.P., Stathis, K., Vouros, G.A. (eds.) ESAW 2007. LNCS (LNAI), vol. 4995, pp. 1–54. Springer, Heidelberg (2008)
35. Sergot, M.J., Craven, R.: The Deontic Component of Action Language $nC+$. In: Goble, L., Meyer, J.-J.C. (eds.) DEON 2006. LNCS (LNAI), vol. 4048, pp. 222–237. Springer, Heidelberg (2006)
36. Sergot, M.J., Sadri, F., Kowalski, R., Kriwaczek, F., Hammond, P., Cory, H.: The British Nationality Act as a logic program. Communications of the ACM 29(5), 370–386 (1988)
37. Sergot, M.J.: Prospects for representing the law as logic programs. In: Clark, K.L., Tarnlund, S.Â. (eds.) Logic Programming, pp. 33–42 (1982)
38. Sergot, M.J.: Sergot. Normative positions. In: McNamara, P., Prakken, H. (eds.) Norms, Logics and Information Systems, pp. 289–308. IOS, Amsterdam (1999)

39. Sergot, M.J., Kamble, A.S., Bajaj, K.K.: Indian central civil service pension rules: A case study in logic programming applied to regulations. In: ICAIL, pp. 118–127 (1991)
40. Stathis, K., Sergot, M.J.: Games as a metaphor for interactive systems. In: Sasse, M.A., Cunningham, J., Winder, R.L. (eds.) People and Computers XI, Proceedings of HCI 1996, London, UK, pp. 19–33. Springer (1996)
41. Toni, F., Sergot, M.J.: Argumentation and Answer Set Programming. In: Balduccini, M., Son, T.C. (eds.) Logic Programming, Knowledge Representation, and Nonmonotonic Reasoning. LNCS, vol. 6565, pp. 164–180. Springer, Heidelberg (2011)
42. Wolstenholme, D.E., O'Brien, C.M.: Glimpse - a statistical adventure. In: IJCAI, pp. 596–601 (1987)