Agent and Multi-Agent Software Engineering: Modelling, Programming, and Verification

Extended Abstract for a Course at DALT Spring School 2011

Rafael H. Bordini

Institute of Informatics
Federal University of Rio Grande do Sul
CP 15064, CEP 91501-970, Porto Alegre – RS, Brazil
R.Bordini@inf.ufrgs.br

In this extended abstract, I shall briefly describe the course I gave at the DALT International Spring School that took place at the University Residential Centre in Bertinoro, Italy, in April 2011. Before I do so, I feel compelled to say, although this was supposed to be a technical paper, that participating in that School was the most fabulous experience of my academic career. I attended many conferences and summer schools over the last decade, and in particular all the summer schools were incredible experiences, but none matched that of the DALT School in Bertinoro. The University Centre is located in an astounding medieval castle, where even Dante stayed for some time. The castle is at the top of a hill and the views from the castle where we were also accommodated are just breathtaking. The food was excellent and the people involved at all levels incredibly friendly; the organisation was impeccable. Of course just atmosphere does not make an academic event that memorable. Perhaps because this was the most specific summer school I ever attend in regards to the topics covered, which allowed the courses to go into much more depth than usual, but certainly not only because of that but also other factors such as the particular combination of people in that School, that was definitely the most technically productive school I ever attended. All the attendants cannot possibly thank enough the organisers for the amazing experience they created for us. Unfortunately, nothing is perfect. As memorable as the school was, I will never be able to forget the suffering it was to walk up those extremely steep hills either.

The course was commissioned by the School organisers with the specific title "Agent and Multi-Agent Software Engineering: Modelling, Programming, and Verification". I liked the title for various reasons. First because it makes explicit the move from single to multi agent that is an important development in agent programming languages. Second because it covers two areas that have been at the centre of my research over many years: programming and verification of multi-agent systems. Modelling is not an area I have contributed to significantly, so I used less time of the course for this part. It consisted simply of mentioning a selection of the best-known methodologies for agent-oriented software engineering as well as an overview of key concepts in modelling multi-agent systems. That

C. Sakama et al. (Eds.): DALT 2011, LNAI 7169, pp. 149–151, 2012.

was after a brief motivation to the development of *autonomous* software, showing the growing number of commercial application of autonomous software as well as pointing out that all current trends in computer science point to directions that are only possible if autonomy in software becomes commonplace. References to some of the main researchers and academic events in the area were made in the modelling as well as later in the agent programming and verification parts of the course. All the material used in the course, including the slides are available online at `http://www.inf.ufrgs.br/~bordini/DALT-SpringSchool-2011`.

The part of the course about programming referred to the variety of agent languages in the literature but focused on the AgentSpeak variant used in the *Jason* platform [5] available at `http://jason.sourceforge.net`. AgentSpeak is a BDI-based agent-oriented programming language, influenced also by logic programming. It has been much referred to in the AAMAS literature and is widely known, perhaps for being both simple and elegant, yet faithful to the BDI architecture and reactive planning systems. *Jason* is a platform that has become rather popular, with some 500 downloads a month on average. While *Jason* includes many extensions of the original AgentSpeak(L) language, and allow multiple agents and also has some support for simulating shared environments, its main strength in the language for programming autonomous (communicating) agents rather than agent organisations or agent environments. Yet both sophisticated social structures and shared environments are equally important to agent programming in complex multi-agent systems.

A platform called JaCaMo was created very recently which addresses that problem. The platform is based on *Jason* for programming agents, $\mathcal{M}$OISE [8,7] (`http://moise.sourceforge.net`) for programming organisations, and CArtAgO [10,9] (`http://cartago.sourceforge.net`) for programming environments. This course briefly presented $\mathcal{M}$OISE and CArtAgO so that JaCaMo could then be presented; JaCaMo is available at `http://jacamo.sourceforge.net`. A didactic example was presented that allowed the demonstrations of the use of first-class programming abstractions at the three levels of a multi-agent systems: social, individual, and environment; the example is available with the course material and also in the JaCaMo releases on Source Forge. For those used to traditional agent-oriented programming, it is quite revealing to see how full multi-agent oriented programming as embodied in JaCaMo can lead to much more powerful programming.

The last part of the course was about verification. As with the other parts, references to other researchers who contributed to agent verification and relevant survey papers were given and then the course centred on the work I have done with various colleagues. Our early work [3] aimed at allowing the use of model checking techniques on systems programmed in AgentSpeak particularly. Later, in a joint project with Michael Fisher and Louise Dennis [6,2], we developed an approach to model check systems programmed in various BDI-based agent languages, including systems where different agents were programmed in different agent programming languages, and focusing on the use of Java Pathfinder [12] (`http://javapathfinder.sourceforge.net/`) as the

underlying model checker. In program model checking [11], the well-known state-space explosion problem is particularly difficult, so it is even more important to use abstraction techniques. The course also mentioned briefly work we have done on property-based slicing for AgentSpeak [4]; property-based slicing is interesting because it is a *precise* form of under-approximation. It was also pointed out that model checking can be useful in practical software development even when full verification is not possible, for example model checking can be used for test case generation [1].

References

1. Artho, C., Barringer, H., Goldberg, A., Havelund, K., Khurshid, S., Lowry, M.R., Pasareanu, C.S., Rosu, G., Sen, K., Visser, W., Washington, R.: Combining test case generation and runtime verification. Theor. Comput. Sci. 336(2-3), 209–234 (2005)
2. Bordini, R.H., Dennis, L.A., Farwer, B., Fisher, M.: Automated verification of multi-agent programs. In: ASE, pp. 69–78. IEEE (2008)
3. Bordini, R.H., Fisher, M., Visser, W., Wooldridge, M.: Verifying multi-agent programs by model checking. Autonomous Agents and Multi-Agent Systems 12(2), 239–256 (2006)
4. Bordini, R.H., Fisher, M., Wooldridge, M., Visser, W.: Property-based slicing for agent verification. J. Log. Comput. 19(6), 1385–1425 (2009)
5. Bordini, R.H., Hübner, J.F., Wooldridge, M.: Programming Multi-Agent Systems in AgentSpeak Using *Jason*. Wiley Series in Agent Technology. John Wiley & Sons (2007)
6. Dennis, L., Fisher, M., Webster, M., Bordini, R.: Model checking agent programming languages. Automated Software Engineering, 1–59 (2011), `http://dx.doi.org/10.1007/s10515-011-0088-x`
7. Hübner, J.F., Boissier, O., Kitio, R., Ricci, A.: Instrumenting multi-agent organisations with organisational artifacts and agents. Autonomous Agents and Multi-Agent Systems 20, 369–400 (2010)
8. Hübner, J.F., Sichman, J.S., Boissier, O.: Developing Organised Multi-Agent Systems Using the MOISE+ Model: Programming Issues at the System and Agent Levels. Agent-Oriented Software Engineering 1(3/4), 370–395 (2007)
9. Omicini, A., Ricci, A., Viroli, M.: Artifacts in the AandA meta-model for multi-agent systems. Autonomous Agents and Multi-Agent Systems 17(3), 432–456 (2008)
10. Ricci, A., Piunti, M., Viroli, M.: Environment programming in multi-agent systems: an artifact-based perspective. Autonomous Agents and Multi-Agent Systems 23(2), 158–192 (2011)
11. Visser, W., Havelund, K., Brat, G.P., Park, S., Lerda, F.: Model checking programs. Autom. Softw. Eng. 10(2), 203–232 (2003)
12. Visser, W., Mehlitz, P.C.: Model Checking Programs with Java PathFinder. In: Godefroid, P. (ed.) SPIN 2005. LNCS, vol. 3639, pp. 27–27. Springer, Heidelberg (2005)