

Socially adaptive software

M. Birna van Riemsdijk

Runtime computational reasoning techniques have the potential to realize socially adaptive software so flexible that it can function as a social partner for people.

I envision a future where software systems function as partners that work with, inspire and support people in varying social situations. To this end, software systems must be capable of social adaptation, just as people adapt their behaviour when changing from one social context to another, such as travelling to another culture or getting home from work. Each social context comes with its norms, regulations and laws, which I call social requirements. When presented with a set of social requirements, people are capable of adapting their behaviour accordingly, even if that is not how they would normally behave.

As an illustration, consider intelligent electronic partners¹ for providing social support to children or the elderly by helping them get acquainted with their social and geographical environment (see also our project on socio-geographical support²). Such electronic partners should be able to support their users in setting up social activities, warning them about location-based risks and so forth. To do this effectively, the electronic partner needs to adapt to a wide variety and continuously changing set of social requirements, depending on who the user is and which social contexts he or she encounters. For example, social requirements concerning who can be contacted in which situation when help is needed or how connections should be formed with whom in setting up a social activity depend on the user, the circumstances of the situation and the other people in the relevant social contexts.

In the current state of the art in computer science and artificial intelligence, a limited form of social adaptation is achieved by fully specifying and implementing the system's behaviour in situations that were anticipated at design time.^{3,4} However, the variability and dynamics of social encounters require developing more flexible techniques if software systems are to function as social partners of people. It is not feasible to fully pre-program the required behaviour for each combination of social requirements that may be encountered, especially as it is not known at design time which combinations these will be. Consequently, the challenge I aim to address is to develop a generic computational reasoning mechanism for making software systems socially adaptive.

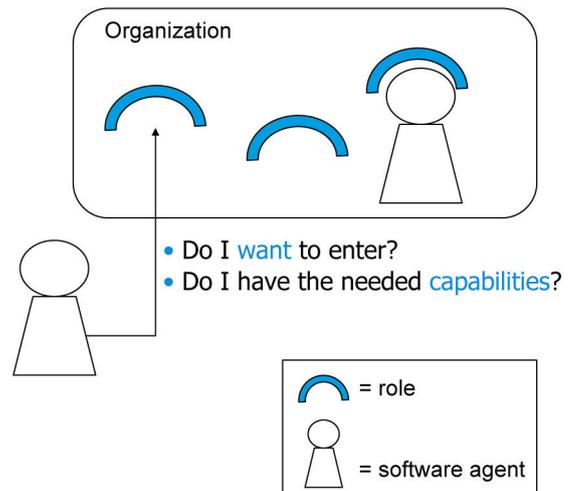


Figure 1. Role enactment. Before entering an organization, organization-aware agents should be able to decide whether they want to start playing a role in the organization and comply with the corresponding social requirements, and whether they have the necessary basic capabilities to behave as the organization requires.

This computational reasoning mechanism should augment existing software systems (which I call base programs) to synthesize socially adapted behaviour at runtime. I refer to the resulting system as a software entity. This requires a fundamental study of how to integrate at runtime the requirements from the social context, the intended behaviour of the base program and the functional capabilities of the base program. Techniques for modelling and recognizing social requirements⁵⁻⁷ and for letting a software entity decide whether it wants to adapt to them (e.g. the European Commission Framework Programme 6 project EMIL,⁸ Emergence in the Loop) have already been developed over the last decade in the area of normative and organized multi-agent systems. My goal is to investigate how a software entity can adapt its behaviour to continuously changing requirements of its social context at runtime while preserving the base program's intended behaviour. I call this intent-preserving compliance.

In previous work we studied socially adaptive software in the context of so-called organization-aware agents.⁹ These agents are able to understand and reason about the structure, work

Continued on next page

processes and norms of the organization in which they operate. The social context is thus the organization (e.g. a crisis management organization or a university), and the corresponding social requirements are the norms and regulations that agents must adhere to when playing a role in that organization.

Before entering an organization, organization-aware agents should be able to decide whether they want to start playing a role in the organization and comply with the corresponding social requirements, and whether they have the necessary basic capabilities to behave as the organization requires (see Figure 1). We have investigated reasoning about capabilities as needed in the role enactment process and have shown how this can be achieved by means of reflection.¹⁰

Once an agent has enacted a role and decided that it wants to adhere to the corresponding social requirements, it needs to adapt its behaviour accordingly. To create agents that are capable of such adaptation, it is important that it is clear what the social requirements mean, i.e. which behaviour is considered compliant. We have made this precise for the MOISE¹¹ organizational modelling language.¹²

In summary, to function as social partners of people, cooperating with and supporting us in our daily lives, software systems must be socially adaptive. We have made first steps towards this goal by investigating organization-aware agents. However, much more research is needed to realize my vision of highly flexible socially adaptive software, in particular in developing computational reasoning techniques for achieving intent-preserving compliance.

This publication was partly supported by the Dutch national programme COMMIT.

Author Information

M. Birna van Riemsdijk

Delft University of Technology
Delft, The Netherlands

M. Birna van Riemsdijk is an assistant professor. She obtained her PhD from Utrecht University (2006). She has carried out research in the design of programming languages and modelling techniques for autonomous agents and multi-agent systems, and in service-oriented systems. She is a member of the board of directors of the International Foundation of Autonomous Agents and Multi-Agent Systems and of the steering committee of the International Workshop on Engineering Multi-Agent Systems.

References

1. G. de Haan, O. A. B. Henkemans, M. Neerinx, and C. van der Mast, *SuperAssist: personal assistants for diabetes healthcare treatment at home*, **Proc. Int'l Fed. Inf. Proc. Work. Group 9.3 Home-Orient. Informat. Telemat. (IFIP WG 9.3 HOIT2005)**, pp. 261–275, 2005.
2. <http://www.commit-nl.nl/projects/wp-packages/socio-geographical-support> Delft University of Technology project on socio-geographical support. Accessed 17 January 2013.
3. D. Hirsch, J. Kramer, J. Magee, and S. Uchitel, *Modes for software architectures*, **LNCS 4344**, pp. 113–126, Springer, 2006.
4. J. Andersson, R. de Lemos, S. Malek, and D. Weyns, *Modeling dimensions of self-adaptive software systems*, **LNCS 5525**, pp. 27–47, Springer, 2009.
5. G. Boella and L. van der Torre, *Regulative and constitutive norms in normative multiagent systems*, **Proc. Knowl. Represent. Conf. (KR'04)**, pp. 255–265, 2004.
6. V. Dignum, **A Model for Organizational Interaction: Based on Agents, Founded in Logic**, PhD thesis, Utrecht University, Institute of Information and Computing Sciences, 2004.
7. B. T. R. Savarimuthu, S. Cranefield, M. Purvis, and M. Purvis, *A data mining approach to identify obligation norms in agent societies*, **Proc. 6th Int'l Conf. Agents Data Mining Interact. (ADMI'10)**, pp. 43–58, 2010.
8. <http://emil.istc.cnr.it/> European Commission Framework Programme 6 project EMIL (Emergence in the Loop). Accessed 17 January 2013.
9. M. B. van Riemsdijk, K. V. Hindriks, and C. M. Jonker, *Programming organization-aware agents: a research agenda*, **Proc. Eng. Soc. Agents' World (ESAW'09)**, pp. 98–112, 2009.
10. M. B. van Riemsdijk, V. Dignum, C. M. Jonker, and H. Aldewereld, *Programming role enactment through reflection*, **Proc. IEEE/WIC/ACM Int'l Conf. Web Intell. Intell. Agent Technol. (WI-IAT'11) 2**, pp. 133–140, 2011.
11. J. F. Hübner, J. S. Sichman, and O. Boissier, *Developing organised multiagent systems using the MOISE+ model: programming issues at the system and agent levels*, **Int'l J. Agent-Orient. Softw. Eng.** 1 (3/4), pp. 370–395, 2007.
12. M. B. van Riemsdijk, K. V. Hindriks, C. M. Jonker, and M. Sierhuis, *Formalizing organizational constraints: a semantic approach*, **Proc. Auton. Agents Multiagent Syst. (AAMAS'10)**, pp. 823–830, 2010.