

Generative Migration of Agents

Frances Brazier Benno Overeinder Maarten van Steen
Niek Wijngaards

Dept. of Computer Science, Faculty of Sciences, Vrije Universiteit
Amsterdam; de Boelelaan 1081a, 1081 HV Amsterdam, The Netherlands
{frances,bjo,steen,niek}@cs.vu.nl

Agents, and in particular mobile agents, offer a means for application developers to build distributed applications. Given homogeneity of agent platform and code base, agent migration is possible. However, many agent platforms exist, differing substantially in the support for agents. Write once - run everywhere is not yet true for agents...

Heterogeneity of agent platforms, combined with heterogeneity in code-bases of agents, leads to an interesting question concerning agent mobility: can an agent migrate in a heterogeneous environment? The answer is relatively simple: an agent needs to be adapted to its destination agent platform and code-base, e.g. by an agent factory.

[In E. Alonso, D. Kudenko, and D. Kazakov (Eds.), *Proceedings of the AISB'02 Symposium on Adaptive Agents and Multi-Agent Systems*, 2002, pp. 116-119.]

Agent Factory

An agent factory is a facility that creates, and modifies, software agents, see [2]. It can be used to adapt agents so that they can use specific programming languages and run on different agent platforms. The design of an agent within an agent factory is based on a *blueprint*. The blueprint of an agent specifies a configuration of conceptual building blocks defining the agent's functionality and behaviour. In addition, one or more configurations of detailed building blocks are specified, defining an operationalisation of the conceptual functionality and behaviour.

A mapping is defined between building blocks at conceptual level and detailed level. A detailed description of a building block includes the operational code. For each conceptual description, a number of detailed descriptions may be devised and vice versa. These detailed descriptions may differ in the operational language, but also in, for example, the efficiency of the operational code. The conceptual descriptions may differ in the modelling paradigm, but also in, e.g., the detail in modelling an agent's functionality.

Building blocks themselves are configurable, but cannot be combined indiscriminately. The open slot concept is used to regulate the ways in which components are combined. An open slot has associated properties at both levels of abstraction that prescribe the properties of the building block to be "inserted".

Generative Migration

The principle of generative migration [1] is depicted in Fig. 1. A blueprint of an agent's functionality is transported together with information on the agent's state. At its destination, an agent factory regenerates the executable code of the agent on the basis of this blueprint. Upon activation, the agent restores its state and resumes execution.

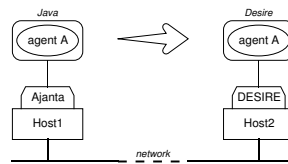


Figure 1: Example scenario of an agent migrating to a different environment.

Ideally, an agent factory is able to (re)generate an agent such that it retains all of its functionality and access to services: transparent adaption. However, this may not be possible in situations requiring generative migration. An agent needs to be aware of characteristics of its current incarnation, including limitations in functionality provided by its current incarnation and services offered by the current agent platform.

Discussion

Migration including (re)generation of agents is a more complex process, than migration without agent re-generation. Four migration scenarios are distinguished[1]: homogeneous migration, cross-platform migration, agent-regeneration migration, and heterogeneous migration.

Agent factories, and generative migration, are services available to agent platforms. Currently, they are supported by AgentScape [3], a middleware layer that supports large-scale agent systems. A prototype of generative migration is being built.

Acknowledgements

This research is supported by the NLnet Foundation, [http:// www.nlnet.nl](http://www.nlnet.nl). The authors wish to acknowledge the contributions made by Hidde Boonstra, David Mobach, Oscar Scholten and Sander van Splunter.

References

- [1] F. M. T. Brazier, B. J. Overeinder, M. van Steen, and N. J. E. Wijnngaards. Agent factory: Generative migration of mobile agents in heterogeneous environments. In *Proc. of the ACM Symposium on Applied Computing (SAC02)*, pages 101–106, 2002.
- [2] F. M. T. Brazier and N. J. E. Wijnngaards. Automated servicing of agents. *AISB journal*, 1(1):5–20, 2001. Special issue on agent technology.
- [3] N. J. E. Wijnngaards, B. J. Overeinder, M. van Steen, and F. M. T. Brazier. Supporting Internet-scale multi-agent systems. *Data and Knowledge Engineering*, 41(2-3):229–245, 2002.