Chapter V

Prometheus: A Practical Agent-Oriented Methodology

Lin Padgham
RMIT University, Australia

Michael Winikoff
RMIT University, Australia

Abstract

We present the Prometheus methodology for designing agents and multi-agent systems. The methodology is intended to be practical; in particular, it aims to be complete and detailed, and to be usable by industrial software developers and undergraduate students. We present the methodology using a case study, describe existing tools that support both design and implementation, and report on experiences with using Prometheus, including our experiences in teaching Prometheus to an undergraduate class over the past few years. These experiences provide evidence that Prometheus is usable by its intended target audience.
Introduction

“One of the most fundamental obstacles to the take-up of agent technology is the lack of mature software development methodologies for agent-based systems” (Luck, McBurney, & Preist, 2004, p. 224).

It is widely accepted in the agent research community that a key issue in the transition of agents from research labs to industrial practice is the need for a mature software engineering methodology for specifying and designing agent systems. In this chapter, we describe the Prometheus methodology that aims to address this need.

In developing the Prometheus methodology, there are a number of motivating considerations that have influenced (and continue to influence) the evolution of Prometheus and the choices made in its development.

- First and foremost, Prometheus is intended to be a practical methodology. As such, it needs to be both complete and detailed. Prometheus has to be sufficiently complete in that it must cover a range of activities from requirements specification through to detailed design; and it has to be sufficiently detailed in that it must provide detailed guidance on how to perform the various steps that form the process of Prometheus.
- Prometheus needs to support (but not be limited to) the design of agents that are based on goals and plans. We believe that a significant part of the benefits that can be gained from agent-oriented software engineering comes from the use of goals and plans to realise agents that are flexible and robust.
- The methodology should facilitate tool support, and tool support should be (freely) available.
- Prometheus needs to be usable by industrial software developers and undergraduate students, not researchers and post-graduate students. In particular, it is important that these groups use the methodology and that their experiences and feedback help guide the development and improvement of the methodology.

These features of Prometheus distinguish it from existing methodologies such as those described in the other chapters of this book, as well as, for example, Brazier, Dunin-Keplicz, Jennings, and Treur (1997); Burmeister (1996); Bush, Cranefield, and Purvis (2001); Collinot, Drogoul, and Benhamou (1996); Drogoul
Related Content

Hierarchical Social Network Analysis Using a Multi-Agent System: A School System Case
[www.igi-global.com/article/hierarchical-social-network-analysis-using-a-multi-agent-system/97687?camid=4v1a](www.igi-global.com/article/hierarchical-social-network-analysis-using-a-multi-agent-system/97687?camid=4v1a)

New Approach to Smooth Traffic Flow with Route Information Sharing
[www.igi-global.com/chapter/new-approach-smooth-traffic-flow/26944?camid=4v1a](www.igi-global.com/chapter/new-approach-smooth-traffic-flow/26944?camid=4v1a)

The Agent-Oriented Methodology MAS-CommonKADS
[www.igi-global.com/chapter/agent-oriented-methodology-mas-commonkads/5055?camid=4v1a](www.igi-global.com/chapter/agent-oriented-methodology-mas-commonkads/5055?camid=4v1a)

Social Simulation with Both Human Agents and Software Agents: An Investigation into the Impact of Cognitive Capacity on Their Learning Behavior
[www.igi-global.com/chapter/social-simulation-both-human-agents/46201?camid=4v1a](www.igi-global.com/chapter/social-simulation-both-human-agents/46201?camid=4v1a)