Chapter 11

THE PROMETHEUS METHODOLOGY*

Michael Winikoff and Lin Padgham

Abstract In this chapter we present the Prometheus methodology for building agent-based software systems. Our goal in developing Prometheus was to have a process with associated deliverables which could be used by industry practitioners and undergraduate students without a previous background in agents. As a result, the Prometheus methodology aims to be detailed and complete, as well as being general-purpose and having tool support. Prometheus comprises three phases: system specification, architectural design, and detailed design. The Prometheus methodology has been developed over a number of years as a response to both educational and industrial needs. The methodology has been used by industrial practitioners, taught at workshops at a number of conferences, and has been taught to undergraduate and postgraduate students, as well as having been used in student projects. These experiences have been positive and we have noticed an enormous difference in the ability of our students to develop agent systems. Using Prometheus third year undergraduates are able to build reasonable agent systems in a one semester course, something that previously was challenging for graduate students.

Keywords: Agents, Software Engineering, Methodologies.

1. Introduction

Prometheus\(^1\) is a methodology for developing agent-oriented software systems. Our goal in developing Prometheus was to have a process with associated deliverables which could be used by industry practitioners and undergraduate students to develop intelligent agents systems, without a previous background.

---


\(^1\)Prometheus was the wisest Titan. His name means “forethought” and he was able to foretell the future. Prometheus is known as the protector and benefactor of man. He gave mankind a number of gifts including fire (from http://www.greekmythology.com).
in agents. To this end Prometheus aims to be *detailed* and *complete* in the sense of covering all the stages of software development as applied to agent systems.

The Prometheus methodology includes three phases:

- **The system specification phase** focuses on (i) identifying the system’s *interface*, that, since we are dealing with situated agents, consists of *percepts* (information from the environment), and *actions*; and (ii) determining the system’s goals, functionalities, and use case scenarios, along with any important shared data. The outputs from this phase are a set of functionality descriptions, percept and action descriptions, system goals, and use case scenarios.

- **The architectural design phase** uses the outputs from the previous phase to determine which agents the system will contain, how they will interact, and what significant events occur in the environment. The outputs of this phase are a system overview diagram, agent descriptions, agent interaction protocols and a list of significant events and messages between agents.

- **The detailed design phase** looks at the internals of each agent and how it will accomplish its tasks within the overall system. The outcomes of this phase are detailed diagrams showing the internal functionality of each agent and its capabilities, process diagrams that show the internal processing of the agent, as well as descriptions of data structures used by the agent, plans and subtasks and the details of plan triggers.

Figure 11.1 indicates the main design artifacts that arise from each of these phases as well as some of the intermediary items and relationships between items. The figure shows the models and dependencies, but does not show the process (although it does imply it).

The development (and revision) of the various models depicted in Figure 11.1 is intended to proceed in an iterative fashion (similar to the Rational Unified Process) where in each iteration the focus of the work gradually shifts further down towards implementation, but where it is expected that most iterations will not be exclusively concerned with a single phase and that many iterations will involve revision of previously developed models.

Figure 11.1 is divided horizontally and vertically. The three horizontal regions form the three *phases* of the methodology discussed above. The leftmost region (consisting of scenarios, interaction diagrams, interaction protocols and process diagrams) deals with descriptions of the *dynamic* behaviour of the system. The middle vertical region (data coupling, acquaintance, system overview, agent overview and capability overview) deal with *overviews* of the system while the remaining models (the right region) give detailed descriptions