Chapter XIII

Creating a Comprehensive Agent-Oriented Methodology:
Using Method Engineering and the OPEN Metamodel

Brian Henderson-Sellers
University of Technology, Sydney, Australia

Abstract

While individual agent-oriented methodologies are useful for restricted situations, a more flexible approach can be found in the use of situational method engineering. Using an underpinning metamodel, a repository of method fragments can be built up and, from this, a selected number of fragments can be abstracted to form an organization-specific or project-specific methodology. As an example, we demonstrate the use of the OPEN metamodel and repository, as extended to support agent-oriented conceptual thinking. Re-creation of existing methodologies, such as Prometheus, is demonstrated with further enhancements from other methodologies such as Tropos and Gaia.
Introduction

Individual methodologies, such as those described so far in this book, are often created with specific purposes in mind, for example, particular domains, particular segments of the lifecycle. Used within those constraints, they are ideally suited. However, users often make the assumption that a methodology is not in fact constrained but, rather, is universally applicable. This can easily lead to “methodology failure” and the total rejection of methodological thinking by software development organizations (e.g., Avison & Fitzgerald, 2003).

The search for a one-size-fits-all methodology has been long and will, ultimately, always be fruitless (e.g., Cockburn, 2000). If the creation of a single universally applicable methodology is an unattainable goal, then we must ask how we might create a methodological environment in which the various demands of different software developers might be satisfied simultaneously. It is argued here that such flexibility might be best obtained by the use of method engineering, preferably based on an underlying metamodel. Formalizing methodological concepts and constraints with a metamodel in this way allows us to identify two component parts of a methodology: (1) the process, and (2) various products created and/or used in the process (Rolland, Prakash, & Benjamen, 1999). Although all the chapters in this book deal with the broader term of “methodology” (i.e., process plus product), for our case study in this chapter, we will focus only on the “process” part of a methodology. A second assumption we make here is that an agent-oriented (AO) methodology can be formulated as a particular extension of an object-oriented (OO) methodology. In the context of method engineering and the earlier chapters of this book, we will demonstrate that this is a reasonable assumption.

In the following sections, we introduce situational method engineering as an effective approach for constructing a site-specific methodology that may be tailored or customized for individual projects in the context of OPEN, particularly as it has been partially extended to support agent-oriented software development. In the next section (CASE STUDY), we demonstrate how such methodology construction works with a simple case study and, finally, we conclude with a research and technology transfer agenda to facilitate the creation, finalization, and technology transfer of such a “standardized” approach in the context of the adoption of agent-oriented methodologies in an industrial context.
Related Content

Information Modeling and the Problem of Universals
www.igi-global.com/chapter/information-modeling-problem-universals/8606?camid=4v1a

Agency in Health Care System Modeling and Analysis
www.igi-global.com/chapter/agency-health-care-system-modeling/37010?camid=4v1a

Simulating UAV Surveillance for Analyzing Impact of Commitments in Multi-Agent Systems
www.igi-global.com/article/simulating-uav-surveillance-analyzing-impact/66066?camid=4v1a

Agent Interaction via Message-Based Belief Communication
www.igi-global.com/article/agent-interaction-via-message-based/3869?camid=4v1a