Part III: Engineering Techniques for Collective Autonomic Systems

In order to guide developers in the construction and maintenance of ensembles, the ASCENS project provides engineering techniques and tools.

The first chapter describes a cornerstone of the ASCENS approach to developing ensembles: the Ensemble Development Life Cycle (EDLC). It is comprised of two feedback loops for activities that take place during design time and during run time of an ensemble. Since many ensembles are long-running systems that are continuously improved and adapted over the course of their lifetime, a third feedback loop is established by deployment and feedback of runtime data to the design phase. These interlocking cycles serve as a framework for the whole development process. Solutions to more specific development issues arising in the individual activities of the design cycle are provided by a catalog of patterns that capture best practices for designing ensembles.

The second chapter addresses issues that arise from the interaction of many autonomous components in a system: self-organization and emergence. To build reliable ensembles these phenomena have to be engineered such that they support the goals of the system and do not lead to unintended consequences. The chapter shows how a strategy of design that follows the problem organization helps to address the issues presented by self-organization and emergence.

Chapters three and four represent two complementary methods for requirements engineering: the goal-oriented Autonomy Requirements Engineering approach that focuses mainly on high-level aspects of the system and its knowledge requirements, and the Invariant Refinement Method that relies on invariants to model both high-level system goals and low-level software obligations.

The fifth chapter describes tools that were developed as part of the ASCENS project and that support the development process, for example a compiler for the BIP language that allows the execution and verification of BIP models, the jRESP runtime environment for the SCEL language or the FACPL policy IDE and evaluation library.

M. Wirsing et al. (eds.): Collective Autonomic Systems, LNCS 8998, p. 323, 2015.