Chapter 4
Agent-Based XDSL Monitoring and Optimization

Giovanni Caire

Abstract. This chapter focuses on a large agent-based system developed and successfully deployed by Telecom Italia in the field of Fixed Network monitoring and optimization. Thanks to its natively distributed agent-based architecture, this system, called Wants-Assurance, continuously monitors about 3,000,000 of xDSL lines in real time. Wants-Assurance is developed on top of WADE (Workflows and Agents Development Environment) a domain independent software platform that allows creating distributed applications leveraging the agent paradigm in conjunction with the workflow metaphor. The chapter is organized as follows: section 1 presents WADE describing its architecture and main features. Section 2 gives an overview of the xDSL network domain highlighting the main phenomena that affect its quality. Section 3 focuses on the Wants-Assurance system, describes its internal architecture and shows how it exploits WADE features to face the challenges set by the application domain.

1 WADE

WADE (Workflows and Agents Development Environment) [1] is a domain independent software platform built on top of JADE [2], a popular open source middleware conceived to facilitate the development of distributed applications based on the agent-oriented paradigm.

As depicted in Figure 1, JADE provides a distributed runtime environment, the agent and behaviour (a task performed by an agent) abstractions, peer to peer communication between agents and basic agent lifecycle management and discovery mechanisms. WADE adds to JADE the support for the execution of tasks defined according to the workflow metaphor and a number of mechanisms and
components that help managing the complexity of the distribution both in terms of administration and fault management.

Thanks to the combination of distributed agents and workflows, WADE is particularly suited to develop applications that require a high degree of scalability and/or imply the execution of possibly long and articulated tasks.

In principle WADE supports “notepad-programming” in the sense that there is no hidden mechanism that developers can’t control. However, especially considering that one of the main advantages of the workflow approach is the possibility of representing processes in a friendly graphical form, WADE comes with a development environment called WOLF [3] that facilitates the creation of WADE-based application. WOLF is an Eclipse [4] plug-in and, as a consequence, allows WADE developers to exploit the full power of the Eclipse IDE plus additional WADE-specific features.

Fig. 1 The WADE platform

1.1 Distribution

WADE inherits from JADE a distributed runtime composed of several “Containers” that can run on different hosts and each one can contain a number of agents [5]. Agents are actually the components that make up a JADE based application as well as a WADE based application. Each type of agent corresponds to a class that inherits from the jade.core.Agent class of the JADE library and many instances of a type of agent can be active in the system.

Containers are the abstraction by means of which the agents that compose an application can be distributed across several hosts. Of course the application can be designed as if all its components (agents) were running locally. Agent distribution across containers and hosts can be defined at deployment time according to scalability requirements.