

# 3APL Platform

E.C. ten Hoeve, M. Dastani, F. Dignum, J.-J. Meyer

Institute of Information and Computing Sciences

Utrecht University

## Abstract

The 3APL Platform is the first distributed multiagent hosting environment for cognitive agents. We demonstrate a Java implementation of the 3APL Platform developed at the Institute of Information and Computing Sciences at Utrecht University. More information on this project can be found at <http://www.cs.uu.nl/3apl/thesis/tenhoeve/tenhoeve.html>. The demonstration gives an overview of the functionalities of the platform. In particular, we will show how 3APL multiagent projects consisting of several interacting cognitive agents can be implemented and executed on this platform. Various aspects of the 3APL Platform such as its distribution across different machines, agent communication, agent management system, and graphical user interface will be demonstrated.

## 1. 3APL Platform

In recent years agent platforms, such as JADE [4], JACK [5], and ZEUS [6], have been created to develop and execute multiagent systems. Although these platforms provide communication facilities as well as specific tools and components such as agent management systems and directory facilitators to register agents and their services, they do not support direct development and execution of *cognitive* agents. The 3APL platform is the first platform that supports easy and direct implementation and execution of cognitive agents. The platform can be distributed across different machines connected in a network. On this platform, cognitive agents can be implemented directly in terms of mental attitudes such as beliefs, basic actions, goals, and reasoning rules [1]. The specifications of FIPA on agent platforms [2] were followed as a guideline. For example, the 3APL platform has an Agent Management System (AMS), which maintains a list of hosted agents on the platform. The AMS handles the lifecycle, i.e. when agents are created their identities are added to the list of available agents; agent identities remain on this list until the agents are removed from the platform.

An important feature of the 3APL platform is that it provides a message transport layer that supports agent communication when the platform is distributed across different machines that are connected in a network. Moreover, the 3APL platform is FIPA compliant in the sense that agents running on this platform can in principle communicate with agents that run on a different FIPA compliant platforms such as JADE. The 3APL platform has a graphical user interface that allows the observation of agent configurations and their changes at run-time. It also allows observing agent interactions and the messages they pass to each other. The GUI supports also easy addition and deletion of agents at run-time by providing template agents and a syntax-highlighting editor. The list of available agents, maintained by the AMS, is presented as a list of trees. Agent identities and their current status are roots of the trees, while their capabilities are presented as children of the trees. Optional tools for debugging purposes are provided by the GUI.

The platform is implemented in Java. An agent on the platform is a java class with private attributes that represent its beliefs, goals, basic actions and reasoning rules. This class has at least two methods, one for configuration of the 3APL object and one for running the object. The first method consists of parsing high-level specifications of agent beliefs, goals, actions, and reasoning rules, which are provided by the agent programmer, and representing them as private attributes. The second method is the implementation of the agent interpreter that determines the execution of the agent. The inter-platform communication was implemented with the use of the Java Agent Services API [3]. The Java Agent Services project is an initiative to define an industry standard for the development of network agent and service architectures.

## References

- [1] <http://www.cs.uu.nl/3apl>
- [2] <http://www.fipa.org>
- [3] <http://www.java-agent.org>
- [4] <http://sharon.cselt.it/projects/jade>
- [5] <http://www.agent-software.com.au>
- [6] <http://more.btexact.com/projects/agents/zeus/index.htm>