

ZEUS: A Collaborative Agents Tool-Kit

From: AAAI Technical Report WS-98-10. Compilation copyright © 1998, AAAI (www.aaai.org). All rights reserved.

Hyacinth S. Nwana, Divine T. Ndumu, Lyndon C. Lee & Jaron C. Collis

Intelligent Systems Research Unit,
MLB1, PP12, BT Laboratories,
Martlesham Heath, Suffolk,
United Kingdom, IP5 3RE
{hyacinth, ndumudt, lyndon, jaron}@info.bt.co.uk
<http://www.labs.bt.com/projects/agents/>

1. Introduction

The ZEUS tool-kit was motivated by the need to provide a *generic, customisable, and scaleable* industrial-strength collaborative agent building tool-kit. The tool-kit itself is a package of classes implemented in the Java programming language, allowing it to run on many hardware platforms.

In creating ZEUS, our design philosophy has been to delineate the *domain-level* problem-solving abilities from the *agent-level* functionality. In other words, we provide classes that implement communication, co-operation, co-ordination, task execution and monitoring, and exception handling, leaving developers to provide code to implement each agent's domain-specific problem-solving abilities.

The tool-kit facilitates the creation of agents by providing a design methodology. The first stage is for the developers to specify the attributes of individual agents through a visual editor. Once an agent is defined, another editor facilitates the definition of each task the agent is capable of performing. At this point the developer only needs to specify each task in terms of what facts are produced and consumed. Once all the agents, tasks and facts have been specified, the *Generator* tool can be invoked, this creates a Java source code *implementation* for each agent, and a *stub* for each task. The individual agent source programs can be compiled into executable agents, and so the developer's sole task is to provide an implementation for each task.

2. The Components of the ZEUS Tool-Kit

The ZEUS tool-kit is a collection of Java class libraries that can be categorised into three functional groups, as shown in Figure 1.

The *Agent Component Library* is a package of Java classes that implement the functionality of collaborative agents; i.e. these classes are the 'building blocks' of the agents created during the generation process. Included in the class library are: a set of co-ordination strategies; a number of predefined organisational relationships, such as peer and superior; and a performative-based agent communication language with a comprehensive instruction set.

Classes from the Agent Component Library also implement three standard utility agents: the Agent Name Server, the Facilitator and the Visualiser. The utility agents fulfil a support role in their society and can be used in any application without modification. The Agent Name Server provides a white pages service, matching agent names to network address just like the Domain Name Servers of the Internet. The Facilitator provides a 'yellow pages' service similar to the 'Yahoo!' index; it is used by agents looking for others who are capable of a particular task or service.

The *Visualisation Tools* are facilities of the dedicated 'Visualiser Agent' that enable users to observe their agents' behaviour. The Visualiser Agent is forwarded a copy of every message an agent sends, and these messages are then collated, interpreted and used to create an up-to-date picture of the agents' collective behaviour. This provides a solution to the inherently difficult problem of analysing and debugging a multi-agent system where the data, control and active processes are distributed.

Users interact with the Visualiser Agent through the five Visualisation Tools listed in Figure 1. Each tool depicts a different aspect of agent society; for instance, the Society Viewer shows all agents known, and the type and frequency of the messages they send; whilst the Reports Tool shows the state of agent tasks and sub-tasks. Screen-shots of these windows in use can be seen in Figure 2.

The *Agent Building Software* components implement the editors that enable users to interactively create agents by visually specifying their attributes and strategies. In order to generate the code for a specific application system, the Generator tool inherits code from the Agent Component library, and integrates the data from the various visual editors. The resulting Java program code is then compiled and executed normally. Existing systems can be linked to the agents using the Application Programmers' Interface (API) of the wrapper class that is also part of the tool-kit. Once an agent's task has been implemented, the agent can be executed and observed with the Visualisation tools provided.

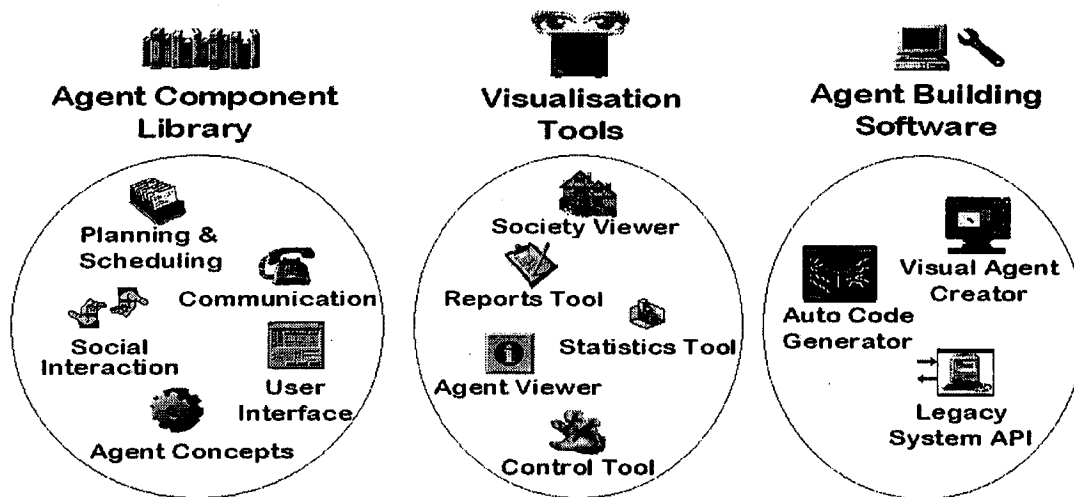


Figure 1 - The Components of the ZEUS Agent Building Tool-Kit. The Agent Component Library provides the building blocks for collaborative agents. To implement communication, a TCP/IP messaging system is provided, together with KQML and FIPA ACL communication languages. To implement social interaction we have provided a library of negotiation strategies, such as contract-net, master-slave, Dutch auction, multiple agent task priority, etc. The Agent Concepts are data structures representing key concepts like facts, attributes, tasks, constraints and acquaintances.

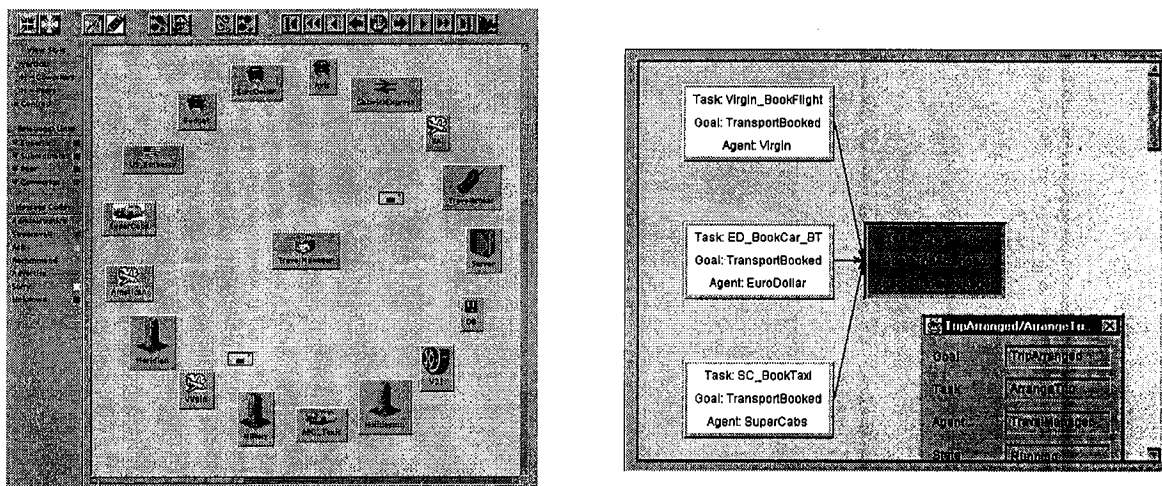


Figure 2 - The Society Viewer (*left*) and Reports Tool (*right*) windows. The Society viewer shows the agents currently active, their relationships and the messages they send; note the tool-bar at the top of the screen that can be used to save and replay agent sessions. The Reports Tool displays the progress of individual agent tasks, with the pop-up window providing extra information when necessary.

Further Reading

Nwana, H.S.; Ndumu, D.T.; Lee, L.C. 1998. ZEUS: An Advanced Tool-Kit for Engineering Distributed Multi-Agent Systems. In Proceedings of PAAM '98, p377-391, March 1998.

Nwana, H.S.; Ndumu, D.T.; Lee, L.C.; Collis J.C. 1998. ZEUS: A Tool-Kit for Engineering Distributed Agent Systems. *International Journal of Applied Artificial Intelligence*. Vol.7/8. To Appear.

These publications and others relating to our work on the ZEUS tool-kit are available online from:
<http://www.labs.bt.com/projects/agents/publish/>