# SEAGENT MAS Platform Development Environment (Demo Paper)

Oğuz Dikenelli
Ege University
Computer Engineering Department
Bornova, Izmir. 35100 TURKEY
oguz.dikenelli@ege.edu.tr

## Categories and Subject Descriptors

I.2.11 [**Distributed Artificial Intelligence**]: Multi-agent systems

## General Terms

Design

## Keywords

Test-driven MAS development, MAS development platform.

## 1. INTRODUCTION

SEAGENT is a goal-oriented MAS development platform developed by Ege University MAS research group. SEAGENT project was officially started in 2002. Its architecture evolved gradually till 2006 and then became stable and actively used within our research group.

SEAGENT [1] has two distinguished characteristics, which differentiate it from the former development frameworks. The first one is its interactive development environment. This environment support test-driven style plan development [1]. Plans and test plans can both be written using the HTN Editor and build-in unit testing tool verifies written plans using the test plans. The development environment also supports verification of plan cooperation, which simplifies the development of cooperating agents.

Second distinguished characteristic of SEAGENT is Semantic Web support. This support has two dimensions. Firstly, SEAGENT manages all of its internal knowledge using OWL ontologies such as goal ontology, plan ontology etc. As a second dimension, it provides a planer level support for the usage of semantic web services [2]. OWL-S ontologies of external services are stored inside the agent and planner of SEAGENT can directly execute OWL-S description when agent intends to use an external semantic service.

[1] For further information *http://seagent.ege.edu.tr*

**Cite as**: SEAGENT MAS Platform Development Environment (Demo Paper), Oğuz Dikenelli, *Proc. of 7th Int. Conf. on Autonomous Agents and Multiagent Systems (AAMAS 2008)*, Padgham, Parkes, Müller and Parsons (eds.), May, 12-16., 2008, Estoril, Portugal, pp. 1671-1672.



## 2. SEAGENT Multi-Agent Framework

The SEAGENT MAS framework consists of two main environments: The Development Environment that provides visual developing tools (Eclipse plug-ins) to SEAGENT developers, and Runtime Environment, which contains infrastructure modules. These environments are shown in Figure 1. In the following section the SEAGENT framework architecture is explained in detail.

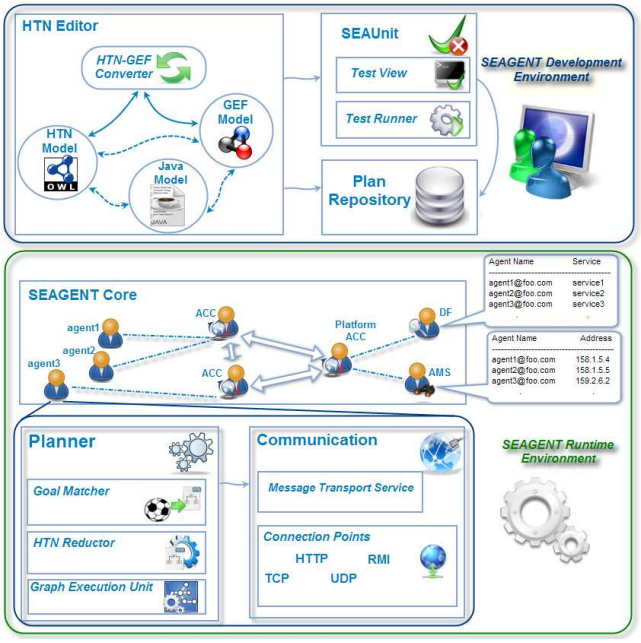


**Figure 1. SEAGENT Development & Runtime Environments**

### 2.1 SEAGENT Development Environment

The development environment (SDE) is designed to assist multi-agent system developers. For this purpose, it provides visual tools that are implemented as Eclipse plug-ins as shown in Figure 2. The SDE consists of three developing tools: HTN Editor, SEAUnit, and Plan Repository.

HTN Editor, labeled with number 1 in Figure 2, visualizes the development process of agent planning. In the SEAGENT framework, agent plans are developed using HTN (Hierarchical Task Network) planning approach. So the HTN Editor provides HTN structures like tasks and parameter links to the developer. It uses GEF (Graphical Editing Framework) to model the HTN structure in the editor view. Agent plan that is designed by

developer is expressed by a GEF model. Atomic tasks in this model can be generated as JAVA classes by code generation facility and whole HTN model can be converted to HTN ontology that contains references to generated JAVA classes. The editor can also execute the designed plan.

SEAUnit, labeled with 2 in Figure 2, is a unit testing tool that is used to develop unit tests of the HTN plans that are developed on the HTN Editor. SEAUnit verifies HTN test plans via the HTN Editor Plan execution facility. It consists of two basic components: Test Runner and Test View. Test Runner executes test cases and notifies test results to the visual part of the testing tool using related events. Test view visualizes the test results of the HTN test plans similar to JUnit.

Plan Repository, labeled with 3 in Figure 2, is a tool which ensures reusing of HTN models that are implemented previously. It presents an Eclipse view and it's integrated with the HTN Editor.

## 2.2 SEAGENT Runtime Environment

The SEAGENT Runtime Environment (SRE) comprises the infrastructure elements of the SEAGENT framework. SRE can be examined from two different viewpoints: multi-agent and agent view points.

From agent viewpoint, SRE provides the internal structure of an agent. It includes HTN Planner and Communication modules as agent's internal component. From multi-agent viewpoint, it specifies the main entities of a multi-agent platform like AMS, DF, and ACC.

Planner is responsible for executing agent plans that are represented by HTN ontology and that implement an agent goal. The Planner consists of three important sub-modules to fulfill agent goals: Goal Matcher, Reductor, and Graph Execution Unit. Goal Matcher is used to find an HTN plan for a given agent goal.

After goal matching is completed, the Planner uses the HTN Reductor Unit to parse and reduce HTN ontology. The HTN Reductor converts the HTN ontology to an executable graph. The produced graph is used as an input by the Graph Execution Unit. During graph execution, it also provides execution monitoring.

Agents achieve their communication requirements using communicator module. This module maintains physical communication over several protocols like TCP, UDP, RMI, and HTTP.

SEAGENT Core implements the main structures of a multi-agent system compatible with FIPA Abstract Specification.

## 3. ACKNOWLEDGMENTS

Since 2002, former and current members of SEAGENT research group have been contributed the development of SEAGENT namely; Rıza Cenk Erdur, İbrahim Çakırlar, Övünç Çetin, Erdem Eser Ekinci, Alpaslan Burak Eliaçık, Özgür Gümüş, Önder Gürcan, Geylani Kardaş, Hüseyin Kır, İnanç Seylan, Ali Murat Tiryaki.

This work is supported by The Scientific and Technological Research Council of Turkey (TUBITAK) Electric, Electronic and Informatics Research Group (EEEAG) under grant 106E008.

## 4. REFERENCES

[1] Tiryaki A.M, Öztuna S., Dikenelli O., and Erdur R.C. Sunit: A unit testing framework for test driven development of multi-agent systems. In AOSE, volume 4405 of LNCS pages 156-173, 2006

[2] E. E. Ekinci, A. M. Tiryaki, O. Gurcan, and O. Dikenelli. A planner infrastructure for semantic web enabled agents. In On the Move to Meaningful Internet Systems 2007: OTM 2007 Workshops, Lecture Notes in Computer Science, pages 95-104, 2007

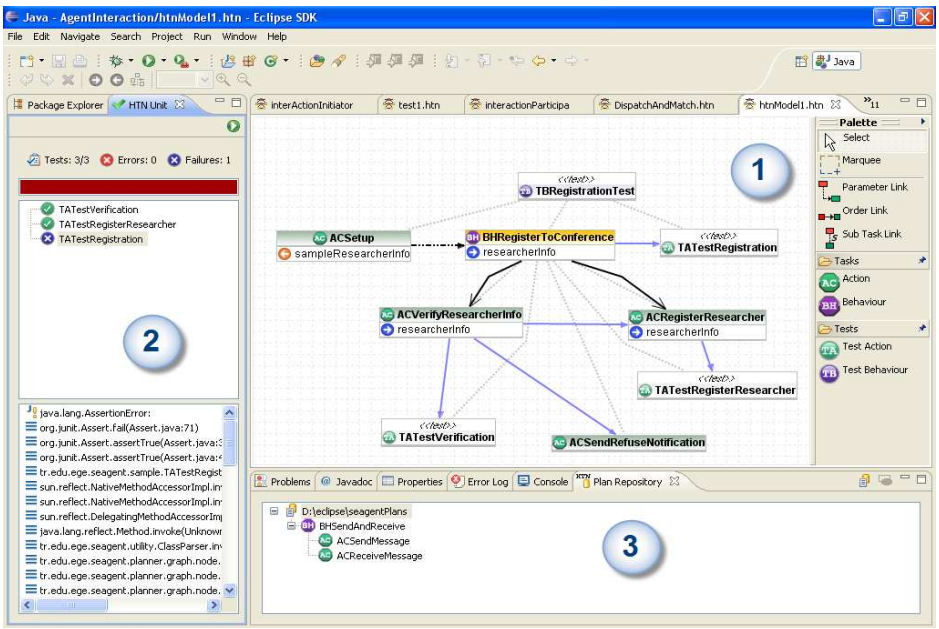


**Figure 2. SEAGENT Development Environment**