# AmItalk: towards MDE/MDA Tool Support for *Ambient Systems*

**Keywords**: Adaptive Object-Models, Multi-agent Systems, MDE/MDA, Model-Transformation, Ambient Intelligence, Smalltalk, VisualWorks, Squeak, Seaside, Mobidyc, Dyctalk, SIXX, XML.
**Smalltalk platforms it runs on**: Squeak and VisualWorks
**Status**: research software, planned for distribution under open source license
**Names of developers**: Stéphane Célet, Reza Razavi, Pouryia Zarbafian
**Affiliation of developers**: University of Luxembourg, Ambient Systems Group
**URL**: http://ambiance.uni.lu
**Contact**: reza.razavi@uni.lu

## Description of the software and screenshots

### Introduction

The software presented here is being developed within a research project called *Åmbiance*[1], funded by the University of Luxembourg and launched in January 2005, in the field of *Ambient Intelligence* (AmI) as defined by ISTAG [1].

The goal of *Åmbiance* is to support the rigorous and cost-effective development and deployment of ambient systems, featuring personalized, secured and reliable adaptive services. As stated in [2], we plan to base the architecture of our ambient systems on the architectural style of Adaptive Object-Models (AOMs) [3, 4]. Specifically, we start by reusing the Dycra framework [5] and adapting it to the AmI context [6]. The software presented here is a first example in this direction.

### Context

As a first approximation of an ambient system, we consider a population of agents in individual-based simulation, as implemented by the Mobidyc project [7, http://www.avignon.inra.fr/mobidyc]. The behavior of those agents is specified by an expert in the field of ethology, using an *ad hoc* language. A salient feature of Mobidyc is that the expert can modify his specifications at run-time and thereby influence a running simulation.

We take this feature as a first example of controlled adaptivity in an ambient system. Our present aim is to push the example one step further by separating the expert from the running system and allowing her to interact through a Web interface: *AmItalk*, the piece of software that is presented here does that.

The main piece of our *AmItalk* software is a Web-based specification editor for the expert, illustrated by the screenshot of Fig. 1. This editor generates an XML file which is directly loaded in the latest version of Mobidyc. This involves modeling the language of the expert in a way that:

(1) makes it easy for the expert to write and modify agent behavior specifications
(2) ensures an adequate communication with the running Mobidyc system to bring about the changes.

The foundation for both aspects (1) and (2) is Dart [6], a formalism for describing domain specific languages which provides:

(1) a spreadsheet-like syntax, conforming with the views of B. Nardi on acceptability by nonprogrammers [8]
(2) an AOM-compliant execution framework.

[1] From Adaptive Object-Models to Meta-tool Support for Ambient Dependable Intelligence Systems.

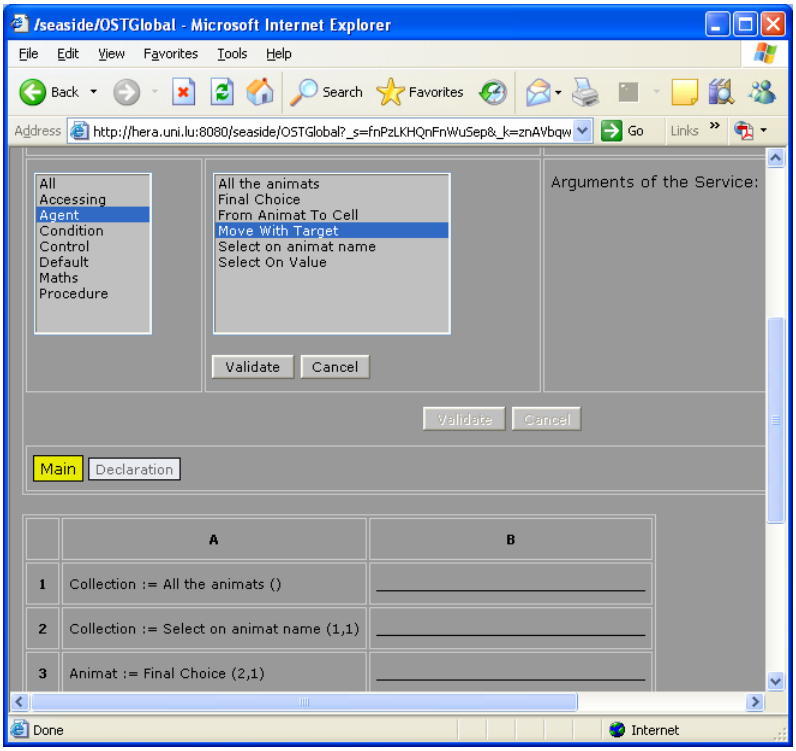


**Figure 1: screenshot of the first application of AmItalk to the case of Mobidyc**

## Architecture

The architecture of *AmItalk* comprises fours subsystems whose role, as well as relations and interactions with other subsystems is succinctly described below:

- *(Ambiance) Presentation Engine (APE)**:*** a web-enabled application which is responsible for the presentation and interaction, playing the role of the system's front-end ("View" and "Controller" in the MVC model). APE offers to experts a spreadsheet-like task-specific language for commanding the system meaningful adaptation. Web pages of this subsystem are dynamically generated at runtime in HTML format, thanks to Seaside. Some static Web pages are also integrated (e.g. online help pages). The logic for page generation is implemented in the next subsystem AME. These two subsystems are in close interaction in a client/server mode.
- *(Ambiance) Model Engine (AME)*: is responsible for representation ("Model" in the MVC model), and implements in *Squeak* and *Seaside* (http://seaside.st/) the domain-related language for experts, which is a spreadsheet-like language, conforming to Dyctalk. AME interacts with the two next subsystems. It uses the *Transformation Engine* subsystem for exporting models in XML and Smalltalk formats. It uses the *Runtime Engine* (AEE) for executing models, and then sending the results to the APE. AME uses the SIXX tool (http://www.mars.dti.ne.jp/~umejava/smalltalk/sixx/) for serializing models.
- *(Ambiance) Transformation Engine* (ATE): is responsible for model transformation (and later verification), and offers currently three functionalities: model-to-XML, model-to-Smalltalk (both implemented in Smalltalk), and model-to-Java (implemented in Java/DOM) transformations. This functionality is implemented in Smalltalk and activated by AMS. Thanks to ATE transformations, the same business logic can be run on different runtime engines and exchanged between different platforms.

- *(Ambiance) Runtime Engine* (ARE): is responsible for execution of models in an adaptive (and later, distributed, concurrent fault-tolerant) environment. We have experimented with two AREs for *Mobidyc*, one as an extension to *MadKit*, and the other one an extension to the *Mobidyc*'s runtime engine itself. We continue working on a more generic runtime engine.

**Perspectives**

*AmItalk* constitutes a first effort towards a more ambitious objective. We plan in effect, to construct a reusable metatool, called *Åmbiance metatool*, for building ambient systems with AOM & MAS architecture. Åmbiance metatool is planned to serve as a platform for developing and deploying ambient systems, through reuse and extension. Therefore, these quality attributes are subject to primary considerations. Another perspective of this work is to integrate hardware. This is in particular planned in the framework of e-Care[2] that targets a lab for *ambient healthcare systems*. Furthermore, in a later stage of the project, we plan exploring emulating human experts by monitoring autonomous adaptive agents endowed with the same expertise and learning capacities, therefore capable of delivering the same customizing instructions as domain experts.

**Acknowledgments**

This research is funded by the University of Luxembourg, Åmbiance project (FSTC R1F105K04), and supported by the Software Engineering Competence Center (Nicolas Guelfi). Stéphane Célet from *Ecole des Mines de Douai*, and Pouriya Zarbafian from *Supélec*, contributed to this work during their master internship, funded by the Åmbiance project. S. Célet contributed to the APM by defining the structure of the Web pages. He also developed the AME, as well as the model-to-XML transformer of the ATE. P. Zarbafian developed the MadKit-based ARE, as well as the XML-to-Java transformer of the ATE. R. Razavi specified the requirements, defined the architecture, implemented the model-to-Smalltalk transformer of the ATE and the Mobidyc-based ARE. He also supervised the project. Vincent Ginot and Hervé Richard offered precious assistance by providing us with Mobidyc, documentation, explanation and several examples of ecosystems. Jean-François Perrot provided valuable feedback and comments on this work.

**References**

1. IST Advisory Group: Ambient Intelligence: from vision to reality - For participation in society & business (2003)
2. Ishida, Toru; Gasser, Les; Nakashima, Hideyuki (Eds.): Massively Multi-Agent Systems I (MMAS I). LNAI Vol. 3446, ISBN: 3-540-26974-6 (2005)
3. Yoder, J.W., Johnson, R.E.: The Adaptive Object-Model Architectural Style. *3rd IEEE/IFIP Conference on Software Architecture (WICSA3)*, pp.3-27, Montréal, Canada, (2002)
4. Razavi, R., Bouraqadi, N., Yoder, J.W., Perrot, J.F., Johnson, R.: Language Support for Adaptive-Object Models using Metaclasses. *Proceedings of the ESUG Research Track*, Köthen, Germany, Sept. 2004. Also in a special issue of the Elsevier international journal Computer Languages, Systems and Structures (2005)
5. Razavi, R.: "Outils pour les Langages d'Experts --- Adaptation, Refactoring et Réflexivité". Ph.D. Thesis, LIP6-OASIS, Université Pierre et Marie Curie (Paris 6), Paris (2001)
6. Razavi, R., Perrot, J.F., Guelfi, N.: Adaptive Modeling: An Approach and a Method for Implementing Adaptive Agents. In MMAS I (2005)
7. Ginot, V., Le Page, C., Souissi, S.: A multi-agents architecture to enhance end-user individual-based modelling, Ecological Modeling 157 pp. 23-41 (2002)
8. Nardi, B.A.: *A Small Matter of Programming: Perspectives on End User Computing*, MIT Press, Cambridge (1993)

[2] In the framework of the e-Care project: Towards a Secured, Efficient Platform for the e-Commerce of Personalized Health Products, funded by the Luxembourg National Research Fund (FNR 04/01/02).