

# CARTAGO and Friends

Open Societies of Heterogeneous Cognitive and Reactive Agents Playing Altogether with  
Artifacts in Shared Workspaces

Alessandro Ricci<sup>1</sup> and Michele Piunti<sup>1,2</sup>

<sup>1</sup> Alma Mater Studiorum, Università di Bologna, DEIS, Cesena, Italy.

<sup>2</sup> Institute of Cognitive Sciences and Technologies, ISTC-CNR, Rome, Italy.  
{a.ricci,michele.piunti}@unibo.it

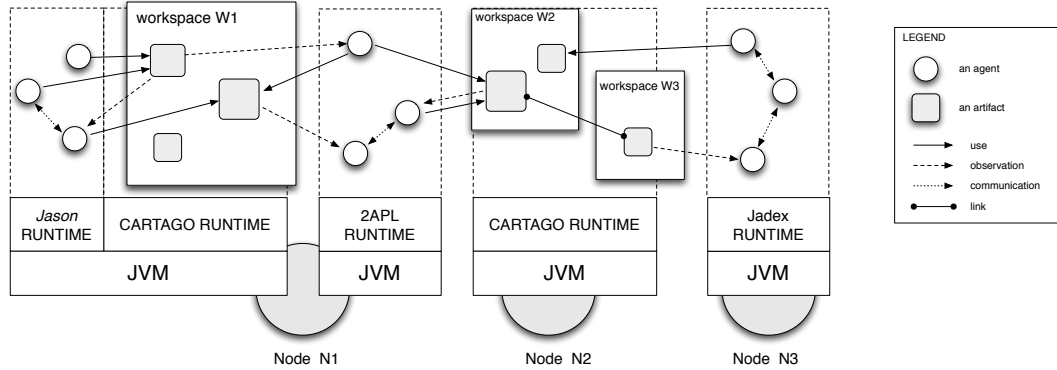
## 1 Introduction

The “Agents and Artifacts” (A&A) [4] and CARTAGO infrastructure [7] are becoming increasingly popular as respectively a conceptual & programming model and related infrastructure for developing artifact-based environments. Here we show the use of CARTAGO integrated with existing agent technologies to develop and execute open, distributed, and heterogeneous multi-agent systems composed by heterogeneous agents—both cognitive and reactive—developed with different agent platforms working together in distributed workspaces held by CARTAGO infrastructure. The platform can be exploited then as an enabling general-purpose technology for experimenting open and heterogeneous MAS integrating different kinds of cooperation strategies among agents working together, based both on direct communication and on environment-mediated interaction (through artifacts, in this case). Some basic case studies are provided with the technology, in order to demonstrate the benefits of the approach in the development of complex multi-agent systems.

## 2 An Integrating Platform for Heterogenous MAS

The technology enables the development and execution of heterogeneous and open MAS where heterogeneous agents belonging to different agent platforms, both cognitive—namely *Jason* [1], *2APL* [2] and *Jadex* [5]—and non cognitive—namely *JADE* [3] and *simpA* [8]—work together in the same CARTAGO workspaces. Fig. 1 shows an abstract example of the kind of multi-agent systems that are possible by exploiting the technology (a description can be found in the caption of the figure).

The integration technology is fully open-source and developed in Java; it is composed by a set of libraries developed for each supported agent platform. A set of new actions extends the basic repertoire and perception capabilities according to the model described in [6]. CARTAGO actions have similar signatures and the same semantics across the integrated agent platforms, and basically make it possible for an agent to: dynamically join and leave workspaces; use an artifact by acting on its usage interface and perceive observable events generated by artifacts; observe an artifact, without using it; creating (and removing) artifacts, specifying the name of the template defining their structure and behavior (programmed by MAS developers); discovering artifacts available in a workspace; composing artifacts together by means of their link interface;



**Fig. 1.** In the picture there are three **CARTAGO** workspaces (called W1, W2, W3) spread among three network nodes (N1, N2, N3). W2 and W3 reside on the same **CARTAGO** node (and so the same Java Virtual Machine, since **CARTAGO** is Java based). W1 artifacts are shared and used by some *Jason* agents running on the same virtual machine and remotely by some *2APL* agents, running on a different virtual machine on a different nodes. The same *2APL* agents are working also with an artifact of W2 workspace, which is linked to an artifact belonging to W3. By means of W2 and W3 artifacts *2APL* agents interact with some *Jadex* agents, residing on a different node and virtual machine.

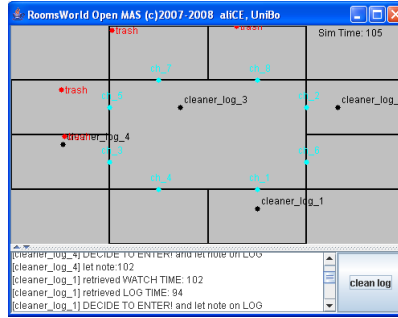
reading an artifact manual, to be aware of the artifact's functionality and operating instructions.

Each library provided with a specific agent platform functions as simple *bridge* for interacting with **CARTAGO** workspaces, which can be either local to the agent runtime or residing in a remote node. The former case refers to the possibility to install and launch **CARTAGO** runtime on the same Java Virtual Machine hosting the agent platform runtime: this is possible since all the agent platforms considered here are based on the Java platform. The integration technology as well as first case studies of MAS based on it are available on **CARTAGO** web space<sup>3</sup>.

### 3 Features of the Integrated Technology

With respect to individual agent platforms, the integration enables the exploitation of a uniform, well-defined, high-level general purpose model for designing and developing agent (work) environments, with strong conceptual foundation based on theories such as Activity Theory and Distributed Cognition. This is meant to replace the weak support typically provided by current agent platforms for modeling the environment and enabling agent-environment interaction, typically provided in terms of low-level mechanisms such as method execution on Java objects. Besides, the technology enables concretely the integration of articulated strategies in solving complex problems in multi-agent systems, in particular coordination/cooperation problems, integrating agent communication capabilities with the cognitive use of suitably designed artifacts, for instance to automate coordination processes, to manage shared knowledge and rule over emergent dynamics.

<sup>3</sup> <http://apice.unibo.it/xwiki/bin/view/CARTAGO/>



**Fig. 2.** The RoomsWorld scenario. Agents holding to heterogeneous platforms are engaged in the same environment, with the goal to clean trash items spread over rooms. Agents rely on additional artifact-based facilities, such as logs and watches

Even more, the distribution model of CARTAGO, based on workspaces possibly spread among different network nodes, and the capability of joining multiple possibly remote workspaces enabled by the integration technology, make it possible in the overall to implement distributed multi-agent systems working in distributed environments, avoiding centralisations.

From the security point of view, in CARTAGO a role-based access control approach (RBAC) is adopted for managing organization and security aspects inside a workspace: thanks to the integrated technology, this model is automatically inherited for free by the specific agent platforms, without the need to change anything in their model/technology. In particular, the approach makes it possible to define at a workspace level a dynamic set of roles with related policies in accessing/using artifacts, policies that can be defined and changed either by human administrators or by agents exploiting suitable artifacts (see CARTAGO documentation for details).

## 4 Towards the Engineering of Truly Open Heterogeneous Agent Societies

Built with CARTAGO technology, the RoomsWorld scenario captures pivotal features of artifact based system built with CARTAGO. In particular, it realises an open system where agents holding to heterogeneous platforms have the possibility to join, test and interact with artifact based workspaces. As in Fig. 2, RoomsWorld environment is composed by a number of virtual rooms separated by walls and doors. Once a room is entered, agents have the goal to find and clean trash items which may appear across rooms with arbitrary frequencies. The main problem for a team of agents engaged in cleaning activities is to coordinate their actions to globally maintain the lower amount of trashes across rooms and thus to avoid wasting resources (i.e., cleaning the same room at the same time).

At a design level artifacts in RoomsWorld cover a twofold role. From the one side, they are assumed to embed the environment and its dynamics, i.e., holding physical consistence of environmental entities and enabling agents in their pragmatic and epistemic activities. On the other side, particular kinds of artifacts are conceived to promote alternative strategies in coordinating agents activities: in this view, artifacts operations

and observable properties are assumed to promote uncoupled and mediated interaction between agents having heterogeneous knowledge models. This is the case, for instance, of *watch* artifacts, supplying agents with timing, *logs* and *checklists*, supplying agents with reports and activity lists.

In the overall, the integrated technology enables the development of MAS exhibiting several degrees of *openness*: in terms of heterogeneous agents that can dynamically join and leave workspaces, in terms of the configuration of workspaces where the set of artifacts can change dynamically depending on the working activities of agents, and in terms of the roles and organisational rules constraining agent actions. A further level of *interoperability* is introduced then besides the common one based on agent communication language such as FIPA-ACL and common ACL ontologies, a new interoperability level based on cooperation based in open agent societies based on sharing, understanding and co-using the same type of artifacts, i.e. resources and tools.

## References

1. R. Bordini, J. Hübner, and M. Wooldridge. *Programming Multi-Agent Systems in AgentSpeak Using Jason*. John Wiley & Sons, Ltd, 2007.
2. M. Dastani, D. Hobo, and J.-J. Meyer. Practical extensions in agent programming languages. In *In Proc. of 6th Int. Joint Conf. on Autonomous Agents and Multiagent Systems (AAMAS'07)*. ACM Press, 2007.
3. G. R. Fabio Bellifemine, Agostino Poggi. JADE - a FIPA-compliant agent framework. In *Proceedings of PAAM'99*, pages pp.97–108, London, Apr. 1999.
4. A. Omicini, A. Ricci, and M. Viroli. Artifacts in the A&A meta-model for multi-agent systems. *Autonomous Agents and Multi-Agent Systems*, 17 (3), To appear. Draft at: <http://projects.deis-ce.unibo.it/public/alice/download/tech-rep/JAAMAS08-draft.pdf>.
5. A. Pokahr, L. Braubach, and W. Lamersdorf. Jadex: A BDI Reasoning Engine. In R. Bordini, M. Dastani, J. Dix, and A. Seghrouchni, editors, *Multi-Agent Programming*. Kluwer, 2005.
6. A. Ricci, M. Pianti, L. D. Acay, R. Bordini, J. Hubner, and M. Dastani. Integrating Artifact-Based Environments with Heterogeneous Agent-Programming Platforms. In *In Proc. of 7th Int. Joint Conf. on Autonomous Agents and Multiagent Systems (AAMAS'08)*, 2008.
7. A. Ricci, M. Viroli, and A. Omicini. CArtAgO: A framework for prototyping artifact-based environments in MAS. In D. Weyns, H. V. D. Parunak, and F. Michel, editors, *Environments for MultiAgent Systems III*, volume 4389 of *LNAI*, pages 67–86. Springer, Feb. 2007.
8. A. Ricci, M. V. Viroli, and G. Piancastelli. simpA: A simple agent-oriented java extension for developing concurrent applications. In *Workshop on Languages, Methodologies and Development Tools for MAS (LADS'007)*, 2007.