Österreichisches Forschungsinstitut für / Austrian Research Institute for / Artificial Intelligence

TR–2011–04

Christian Eis, Marcin Skowron, Brigitte Krenn

Virtual Agent Modeling in the RASCALLI Platform

● Freyung 6/6 ● A-1010 Vienna ● Austria ●

● Phone: +43-1-5336112 ●

● mailto:sec@ofai.at ●

● http://www.ofai.at/ ●

Österreichisches Forschungsinstitut für / Austrian Research Institute for / Artificial Intelligence

TR–2011–04

Christian Eis, Marcin Skowron, Brigitte Krenn

Virtual Agent Modeling in the RASCALLI Platform

The Austrian Research Institute for Artificial Intelligence is supported by the Austrian Federal Ministry for Science and Research and the Austrian Federal Ministry for Transport, Innovation and Technology.

Citation: Eis C., Skowron M., Krenn B.: Virtual Agent Modeling in the RASCALLI Platform. Technical Report, Österreichisches Forschungsinstitut für Artificial Intelligence, Wien, TR-2011-04, 2011

Virtual Agent Modeling in the RASCALLI Platform

Christian Schollum
Research Studios Austria
Vienna, Austria
christian.schollum@researchstudio.at

Marcin Skowron and Brigitte Krenn
Austrian Research Institute for Artificial Intelligence
Vienna, Austria
{marcin.skowron,brigitte.krenn}@ofai.at

Abstract — The RASCALLI platform is both a runtime and a development environment for virtual systems augmented with cognition. It provides a framework for the implementation and execution of modular software agents. Due to the underlying software architecture and the modularity of the agents, it allows the parallel execution and evaluation of multiple agents. These agents might be all of the same kind or of vastly different kinds or they might differ only in specific (cognitive) aspects, so that the performance of these aspects can be effectively compared and evaluated.

Keywords: *Cognitive Agents, Agent Modeling and Evaluation*

I. Introduction

This paper gives an overview of the architecture and functionality of the RASCALLI platform, developed as part of the RASCALLI project[1]. In this project, the platform is used as the underlying software environment for the development and execution of so called RASCALLI (Responsive Artificial Situated Cognitive Agents that Live and Learn on the Internet). It provides the facilities for an user-agent and agent-agent communication and serves as a testbed for the evaluation of various incarnations of the agents that use different sets of action-perception tools, action selection mechanisms and knowledge resources[2] [3]. The platform supports a modular development style, where agents are assembled from small re-usable building blocks. The agents of different kinds and configurations can run simultaneously within a single platform environment. This enables the evaluation and comparison of different agents as well as the evaluation of whole agent communities.

II. Project Objectives

The project RASCALLI aims at the development of virtual agents that perform tasks related to accessing and processing information from the Internet, domain specific databases and knowledge repositories. RASCALLI agents, further referred to as Rascalli, represent a growing class of cooperative agents that do not have a physical presence, but nevertheless are equipped with major ingredients of cognition including situated correlates of physical embodiment to become adaptive, cooperative and self improving in a virtual environment, given certain tasks. The project objectives cover the following topics: development of a computational framework for realization of cognitive agents providing intelligent assistance capabilities, cognitive architecture and modeling, perception and action, reasoning, learning, communication, agent-to-agent and agent-to-user interfaces. The Rascalli agents answer the users' questions, learn the users' preferences and interests, and use this knowledge to present the users with new and possibly interesting information. The agents exist in an environment consisting of external knowledge sources on the Internet, such as search engines and RSS feeds, and music domain-specific knowledge bases. They communicate with their users as well as with other Rascalli. Rascalli agents are developed in a modular fashion, which allows individual agents to be built from different sets of components. This allows e.g. a creation of agents which are "experts" in different knowledge domains. During their lifetime, each agent adapts to its user's interests and thus further specializes in a certain sub-domain and evolves in accordance to the user's preferences. This results in a community of specialized agents, which can communicate with each other to provide a requested information to their users. The modular approach allows evaluation and comparison of different sets of components, including different cognitive aspects (e.g. different learning strategies) by executing multiple Rascalli in the same environment and evaluating their performance (e.g. by means of user tests).

III. Related Work

Research in the software platform for cognitive systems has been conducted in multiple directions, including multi-agent platforms (e.g. JADE[3] and other FIPA[4] compliant platforms), as well as platforms and development methodologies for cognitive systems, such as BOD [5] [1] and AKIRA[6] [2]. However, none of the systems or methodologies meets all of the requirements of the RASCALLI project.

IV. Rascalli Platform Architecture

A. Runtime Features

Multi-Agent - the platform supports the concurrent execution of multiple agents, including the agents of the same kind,

[1]EC Cognitive Systems Project IST-027596-2004 RASCALLI.

[2]The set of the developed action-perception and communication tools as well as the RASCALLI platform will be released under an open source license.

[3]Java Agent Development Framework (http://jade.tilab.com/)

[4]Foundation of Intelligent Physical Agents (http://www.fipa.org/)

[5]Behavior Oriented Design (http://www.cs.bath.ac.uk/ai/AmonI-sw.html)

[6]https://sourceforge.net/projects/a-k-i-r-a/

as well as the agent of different kinds, ranging from very similar to vastly different.

Multi-User - each agent has a single user (but one user may have multiple agents).

B. Development Features

Component-Based Architecture - all the parts that are required to set up a specific agent are developed in a component-based fashion so that individual Rascalli can be assembled from these components in a Lego-like manner.

Multi-Version - in order to support the concurrent development of different kinds of agents based on shared components, the platform supports the execution of multiple versions of the same components at the same time.

System Integration - external (and possibly legacy) components need to be integrated only once and made available to all agents running in the platform in an easy-to-use manner.

Distributed Development - multiple developers use a single platform instance concurrently for developing different agents, with minimal interference between developers.

The RASCALLI platform is implemented as a three-layered architecture:

C. Infrastructure Layer

The infrastructure layer contains basic tools and components used in the RASCALLI project. Specifically, these are Java, Maven[7] and OSGi. In addition, this layer contains custom-made development and administration tools for the RASCALLI platform, such as user interfaces for agent configuration and deployment tools.

D. Framework Layer

The framework layer comprises general platform services and utilities employed by the Rascalli, including communication (user to agent, agent to agent), event handling, RDF handling, technology integration (Perl, web services, etc.), and various other platform services.

E. Agent Layer

The agent layer is the application layer of the platform and contains the components of the actual agents. This layer consists of multiple sub-layers, which are presented in Fig. 1.

Agent Architecture Layer - an agent architecture is a blueprint defining the architectural core of the particular type of Rascalli. More precisely, it sets the roles of agent components and provides means for defining and assembling a specific agent. The architecture can also contain implementations of common components shared by all agent definitions.

Agent Component Layer - contains implementations of the roles defined on the Agent Architecture Layer.

Agent Definition Layer - an agent definition is an assembly of specific components of the Agent Components Layer of a specific agent architecture. Different agent definitions for the same agent architecture might contain different components for certain roles.

[7]http://maven.apache.org/

Agent Instance Layer - contains the individual Rascalli instances. Each Rascallo is an instantiation of a specific agent definition.

V. Agent Evaluation in the Rascalli Platform

One of the goals of RASCALLI is the investigatation how the use of cognitive aspects in an agent can improve the user expercience. Due to its modular approach and the support for multiple agent architectures and agent definitions, the RASCALLI platform is well suited as a testbed for the evaluation and comparison of such agents. Since the different agents can exist at the same time in the same environment and thus be subject to the same external influences, they can be realiably compared and evaluated.

Within the scope of the RASCALLI project, the two agent definitions (Simple and DUAL Music Companion) are compared in a user study to evaluate the benefit of using a cognitive architecture. While there are currently no automated performance measures built into the platform, its modular nature allows for the easy integration of such measures with existing and future agent architectures.

VI. Conclusions

The RASCALLI platform meets a unique set of requirements, that is not targeted by any of the investigated platforms or methodologies. In particular, it supports the development and execution of multiple agents of different kinds. Furthermore, it supports the evaluation and comparison of such different agents within a single environment.

References

[1] J.J. Bryson (2001), *Intelligence by Design: Principles of Modularity and Coordination for Engineering Complex Adaptive Agents*. PhD thesis, MIT, Department of EECS, Cambridge, MA, June 2001, AI Technical Report 2001-003.

[2] G. Pezzulo and G. Calvi (2007), *Designing Modular Architectures in the Framework AKIRA*. Multiagent and Grid Systems, 3, 65–86.

[3] M. Skowron, J. Irran and B. Krenn (2008), *Computational Framework for and the Realization of Cognitive Agents Providing Intelligent Assistance Capabilities*. Proceedings of the Cognitive Robotics Workshop at the 18th European Conference on Artificial Intelligence (to appear).