A review on Multi-Agent platforms and Environmental Decision Support Systems Simulation tools

Thania Rendón-Sallard, Miquel Sànchez-Marrè

Knowledge Engineering and Machine Learning Group, Universitat Politècnica de Catalunya c/Jordi Girona 1-3, E08034, Barcelona, Spain
{trendon,miquel}@lsi.upc.edu

Abstract. This work depicts the state of the art of Multi-agent systems platforms for the development of Multi-agent systems, and reviews the Environmental Decision Support Systems Simulation tools available.

Keywords: Multi-Agent Systems platforms

1. Introduction

In order to design and develop a Multi-Agent System (MAS) to simulate different alternative scenarios for decision-making in Environmental Decision Support Systems (EDSS), we have revised the state of the art of MAS platforms and EDSS simulation tools. Particularly, we are interested in river basin systems management through a Multi-Agent system approach.

River catchments are important social, economical and environmental units. They sustain ecosystems, which are the main source of water for households, agriculture and industry. Due to population growth, industry and overexploitation, the demands made on the river basin are increasing while the capacity of the catchment to meet these demands is decreasing. Therefore the protection of all surface waters and groundwaters must be assured in their quality and quantity. The best way to fulfill these requirements is with a management system at catchment scale that integrates all the water systems involved (sewer system, Waste Water Treatment Plants and River) [2, 5].

The management of river basins involve many interactions between physical, chemical and biological processes ergo these systems become very intricate. Some of the problematic features found

in the river basin domain are intrinsic instability, uncertainty and imprecision of data or approximate knowledge and vagueness, huge quantity of data, heterogeneity and different time scales to name a few [1].

Multi-Agent Systems have the ability to cope with complex problems such those related to river basin system management. Thus, we have proposed the design of a MAS to simulate different alternative scenarios for decision-making in river basin systems management [13].

The aim of this paper is to survey the state of the art in MAS platforms and EDSS simulation tools. This review presents a variety of Multi-Agent related work such agent languages or environments for MAS development, agent platforms and tools, and a number of methodologies for the development of MAS. There also is a section dedicated to the work done in modelling and simulation of EDSS.

Our idea was to select the most suitable agent platform for our goals, and afterwards, to extend it with the new required functionalities. The most essential extension will be the ability of some kinds of agents to incorporate some knowledge and reasoning models.

1.1 Organisation of the paper.

Section 1 presented an Introduction to MAS and our "problem". In Section 2 we present the several agent platforms, languages, environments found in the literature. For each one we will give a brief description along with a comparative table that summarizes some features of the products reviewed. Section 3 introduces a list of methodologies for the MAS development, as well as the standards of communication between agents. In Section 4 the modelling and simulation tools for EDSS are presented. And finally, in Section 5 we state the conclusions.

2. Multi-Agent System platforms

Nowadays, a great number of agent platforms and languages for the MAS development can be found in the literature. This section presents a wide variety of approaches (commercial and academic) found in the agent domain. Table 1 summarizes the toolkits that will be presented here.

PLATFORM	TYPE	AVAILABILITY	SUPPLIER	WEBPAGE
3APL	Language or environment for MAS development	Academic (Free)	Universiteit Utrecht	http://www.cs.uu.nl/3apl/
ABLE	Language or environment for MAS development	Academic (Free)	IBM T. J. Watson Research Center	http://www.alphaworks.ibm.com/tech/able
ADK	Agent platform	Commercial (Licensed)	Tryllian	http://www.tryllian.org/
Agent Factory	Language or environment for MAS development	Academic (Licensed)	PRISM Lab. University College Dublin	http://www.agentfactory.com/
AgentBuilder	Agent platform	Commercial (Licensed)	AgentBuilder	http://www.agentbuilder.com/
AgentSheets	Language or environment for MAS development	Commercial (Licensed)	AgentSheets Inc.	http://www.agentsheets.com
AgentTool	MAS engineering approach	Academic (Free)	Kansas State University	http://macr.cis.ksu.edu/projects/agentTool/agentool.htm
Aglets	Agent platform	Academic (Licensed)	IBM Tokio Research Lab.	http://aglets.sourceforge.net/
A-Globe	Agent platform	Academic (Free)	Agent Technology Group (Czech)	http://agents.felk.cvut.cz/aglobe/
AMETAS	middleware platform for mobile agent-based	Academic (Licensed)	Universität Frankfurt	http://www.accsis.com/ametas/
April	Agent platform	broken link	Fujitsu Labs	http://www.nar.fujitsulabs.com/
Bee-gent	Language or environment for MAS development	Commercial (Free)	Toshiba Corporation	http://www2.toshiba.co.jp/rdc/bee-gent/index.htm
CABLE	Language or environment for MAS development	Commercial (Licensed)	Logica UK Ltd	http://public.logicacmg.com/~grace/Architecture/Cable/public/
Comet Way	JAK for automated services	Commercial (Free)	Comet Way Ltd.	http://www.comet-way.com/content.agent?page_name=Home
CORMAS	Language or environment for MAS development	Commercial (Free)	CIRAD	http://cormas.cirad.fr/indexeng.htm
Cougaar	Language or environment for MAS development	Academic (Licensed)	DARPA / BBN Technologies	http://www.cougaar.org/
Cybele	Language or	Commercial (Li-	Intelligent	http://www.cybelep

	environment for MAS develop-ment	censed)	automation, Inc	ro.com/
DECAF	Language or environment for MAS develop-ment	Academic (Free)	University of Delaware	http://www.cis.udel.edu/~decaf/
DESIRE	Agent platform	Academic (Free)	Vrije Univer-siteit Amster-dam	http://www.few.vu.nl/~wai/demas/tools2.html
DIETagents	Agent platform	Academic (Licensed)	Future Tech-nologies Group	http://diet-agents.sourceforge.net/Index.html
EXCALIBUR Agent	EXCALIBUR Agent	Architecture	AI center	http://www.ai-cen-ter.com/projects/excalibur/
FIPA-OS	Language or environment for MAS develop-ment	Commercial (Free)	Emorphia Ltd.	http://fipa-os.sourceforge.net/index.htm
Grasshopper	Language or environment for MAS develop-ment	Commercial (Li-censed)	IKV++ Technologies AG	http://www.grasshopper.de
IDOL	Language or environment for MAS develop-ment	Commercial (Li-censed)	Commercial (Licensed)	http://www.agentlink.org
IMPACT	Methodology for de-sign/developme nt	Academic (Li-censed)	University of Maryland	http://www.cs.umd.edu/projects/impact/
INDUS	Language or environment for MAS develop-ment	Academic (Free)	N/A	http://developer.au meganet-works.com/
INGENIAS	Methodology for de-sign/developme nt	Academic (Licensed)	Universidad Complutense de Madrid	http://ingenias.sour ceforge.net/
JACK	Language or environment for MAS develop-ment	Commercial (Li-censed)	Agent Oriented Software Pty. Ltd	http://www.agent-soft-ware.co.uk/shared/home/
Jackdaw	Language or environment for MAS develop-ment	Commercial (Li-censed)	Calico Jack (UK)	http://www.cali co-jack.co.uk/solutions.html
JADE	Support soft-ware	Commercial (L gen-eral public license)	TILAB	http://jade.tilab.com/
JADEX	Language or environment for MAS develop-ment	Academic (Licensed)	University of Hamburg	http://vsis-www.informatik.uni-ham-burg.de/projects/jadex/
JAM	Language or	Academic (Licensed)	University of	http://www.mar

	environment for MAS development		Michigan	cush.net/IRS/irs_downloads.html
Jason	Language or environment for MAS development	Academic (Licensed)	University of Durham	http://jason.sourceforge.net/
JATLite	Language or environment for MAS development	Academic (Licensed)	Univesity of Stanford	http://java.stanford.edu/
JATLiteBean	Support software	Academic (Free)	University of Otago	http://kmi.open.ac.uk/people/emanuela/JATLiteBean/
JESS	Language or environment for MAS development	Academic (Free)	Sandia National Laboratories	http://herzberg.ca.sandia.gov/jess/
JIAC	Framework to develop and support agent technology based on Serviceware	Commercial (Licensed)	Dai Lab. (Technische Universität Berlin)	http://www.dai-lab-bor.de/4.0.html?L=1
JINI	open architecture	Commercial (Free)	Sun Microsystems, Inc.	http://www.sun.com/software/jini/
Kaariboga	Language or environment for MAS development	Other (Free - BSD-style license)	Other	http://www.projectory.de/kaariboga/index.html
Living Markets	Language or environment for MAS development	Commercial (Licensed)	Living Systems AG	http://www.living-systems.com/pages/index.html
LOST WAX	Language or environment for MAS development	Commercial	Lost Wax	http://www.lostwax.com
MadKit	Agent platform	Academic (Licensed)	University of Montpellier	http://www.madkit.org
MAML	MAS-based modelling/simulation tool	Commercial (Free)	Agent-Lab Ltd.	http://www.maml.hu/
MASSYVE KIT	Language or environment for MAS development	Academic (Free)	Federal University of Santa Catarina	http://www.gsigma-gru-con.ufsc.br/softs.html
NARVAL	Framework to develop personal assistants	Commercial (GN general public license)	Logilab	http://www.logilab.org/
RePast	MAS-based modelling/simulation tool	Academic (Free- BSD-style license)	University of Chicago	http://repast.sourceforge.net/
RETSINA	Language or environment for	Academic (Licensed)	Carnegie Mellon University	http://www.cs.cmu.edu/~softagents/

	MAS development			
NUIN	Language or environment for MAS development	Academic (Licensed)	University of Liverpool	http://www.nuin.org/
SAGE	Agent platform	Academic (Licensed)	NUST Institute of Information Technology	http://sage.niit.edu.pk/About_SAGE.htm
SEMOA	Support software	Other (Free)	Fraunhofer Institut fuer Graphische Datenverarbeitung	http://www.semoa.org/about/news.html
SIM_AGENT	Language or environment for MAS development	Academic (Free)	University of Birbingham	http://www.cs.bham.ac.uk/~axs/cog_affect/sim_agent.html
SOAR	architecture for developing systems that exhibit intelligent behavior	Commercial	Soar Technology, Inc.	http://ai.eecs.umich.edu/soar/software.html
SPARK	Agent platform	Academic (Licensed)	Artificial Intelligence Center of SRI International	http://www.ai.sri.com/~spark/
StarLogo	Language or environment for MAS development	Academic (Free)	MIT MEDIA lab	http://education.mit.edu/starlogo/
TuCSon	Support software	Academic (Licensed)	DEIS Universita' di Bologna - Sede di Cesena	http://www.lia.deis.unibo.it/Research/TuCSoN/
Voyager	Support software	Commercial (Licensed)	Recursion Software, Inc.	http://www.objectspace.com/products/voyager/
Xraptor	Source code	Academic (Free)	Johannes Gutenberg-Universität Mainz	http://www.informatik.uni-mainz.de/~polani/XRaptor/XRaptor.html
ZEUS	Language or environment for MAS development	Commercial (Free)	BTexact Technologies	http://labs.bt.com/projects/agents/zeus/

Table 1. Multi-Agent System platforms.

3APL is a new kind of agent language, which claims to address a drawback of BDI-architectures, namely the missing support for goal deliberation. Currently there is a 3APL agent interpreter, but no multi-agent platform is yet available.

ABLE is a Java framework, component library, and productivity tool kit for building intelligent agents using machine learning and reasoning. The ABLE research project is made available by the IBM T. J. Watson Research Center.

ADK (The agent development kit) is a commercial agent platform that emphasizes the mobility and security aspects. It is used in several commercial projects especially for legacy system integration.

AGENT FACTORY has been developed as part of ongoing research at University College Dublin, that is concerned with the creation of "a cohesive framework that supports a structured approach to the development and deployment of agent oriented-applications.

AGENTBUILDER is an agent platform based on the notions of mental states, which comply to the agent language Agent-0 proposed by Shoham. The platform is KQML-compliant.

AGENTSHEETS is an authoring tool that allows non-programmers to create agents with behaviors and missions, teach agents to react to information and process it in personalized ways, and combine agents to create sophisticated interactive simulations and models. Customers use AgentSheets to create interactive games, virtual worlds, training simulations, information gathering and personalizing agents, and other interactive content.

AGENTTOOL is a Java-based graphical development environment to help users analyze, design, and implement multi-agent systems. It is designed to support the Multiagent Systems Engineering (MaSE) methodology. The system designer defines high-level system behavior graphically using the Multi-agent Systems Engineering methodology. The system design defines the types of agents in the system as well as the possible communications that may take place between agents.

AGLETS is a Java mobile agent platform and library that eases the deployment of agent based applications. An aglet is a Java agent able to autonomously and spontaneously move from one host to another.

A-GLOBE is an agent platform designed for testing experimental scenarios featuring agents' position and communication inaccessibility, but it can be also used without these extended functions. The platform provides functions for the residing agents, such as com-

munication infrastructure, store, directory services, migration function, deploy service, etc. Communication in a-globe is very fast and the platform is relatively lightweight.

AMETAS is an acronym for Asynchronous MEssage Transfer Agent System. This emphasizes the basic philosophy of this system: Agents may only communicate by exchanging asychronous messages. Method calls are not possible. We believe that this idea grants for the autonomy of agents. Therefore, agents are more than mobile programs.

APRIL AGENT PLATFORM (AAP) is a lightweight FIPA-compliant agent platform that has been written in the Agent Process Interaction Language (April) and is FIPA-compliant.

BEE-GENT is a new type of development framework in that it is a 100% pure agent system. As opposed to other systems which make only some use of agents, Bee-gent completely "Agentifies" the communication that takes place between software applications. The applications become agents, and all messages are carried by agents. Thus, Bee-gent allows developers to build flexible open distributed systems that make optimal use of existing applications.

CABLE is a key part of the GRACE system architecture. It provides support to users in the development and running of intelligent multi-agent applications. CABLE, developed by Logica for this project, is a highly productive environment for developing large and complex distributed applications for i) intelligent decision support and ii) modelling and simulation. It is particularly well suited to distributed software development projects of the type undertaken by the GRACE consortium.

COMET WAY JAK implements a simple kernel-based agent programming model and makes it easy to develop agent-based components that are modular, reusable, extendable, reliable, and intuitive. The software is free.

CORMAS is a programming environment dedicated to the creation of multi-agent systems, with a specificity in the domain of natural-resources management. It provides a framework for developing simulation models of coordination modes between individuals and groups that jointly exploit common resources.

COUGAAR follows a "Cognitive Agent Architecture" and is a DARPA-funded open-source agent platform, that offers special support for logistics problems. The platform is not FIPA-compliant.

CYBELE. In today's growing market for agent-based solutions, CybelePro provides its users with a robust high-performance infrastructure for rapid development and deployment of large-scale, high performance agent-based systems. CybelePro is the commercial release of Intelligent Automation, Inc.'s Cybele agent infrastructure that has been used extensively by the government, industry and academia for applications such as military logistics, modeling, simulation and control of air and ground transportation, communication networks and a development of open systems.

DECAF (Distributed, Environment-Centered Agent Framework) is a toolkit which allows a well-defined software engineering approach to building multi-agent systems. The toolkit provides a stable platform to design, rapidly develop, and execute intelligent agents to achieve solutions in complex software systems. DECAF provides the necessary architectural services of a large-grained intelligent agent: communication, planning, scheduling, execution monitoring, coordination, and eventually learning and self-diagnosis . This is essentially, the internal "operating system" of a software agent, to which application programmers have strictly limited access.

DESIRE (DESIGN and Specification of Interacting REasoning components) is a compositional development method for multi-agent systems based on a notion of compositional architecture and developed by Treur and collegues at the Vrije Universiteit Amsterdam. In this approach, the design of the agents is based on the following main aspects: process composition, knowledge composition, and relations between knowledge and process composition.

DIET Agents is a multi-agent platform that was developed as part of an EU project and was released as Open Source at the end of the project. The platform is developed to be lightweight, scalable, and robust and is targeted to peer-to-peer and/or adaptive, distributed applications that use bottom-up, nature-inspired techniques.

EXCALIBUR. The project's goal is to develop a generic architecture for autonomously operating agents, like computer-guided characters/mobiles/items, within a complex computer-game environment. Source code is not available yet but an early version of a subpart of the software - the DragonBreath search/optimization engine - is already available.

FIPA-OS was one of the first open-source FIPA-compliant software frameworks. It is implemented in Java and uses a simple task-based approach as internal agent structure.

GRASSHOPPER is an agent platform that is especially concerned with mobility and mobile devices. It complies both MASIF and FIPA standards.

IMPACT. The principal goal of the project is to develop both a theory as well as a software implementation that facilitates the creation, deployment, interaction, and collaborative aspects of software agents in a heterogeneous, distributed environment. IMPACT provides a set of servers (yellow pages, thesaurus, registration, type and interface) that facilitate agent interoperability in an application independent manner. It also provides an Agent Development Environment for creating, testing, and deploying agents.

INDUS is a software platform for Ubiquitous, Autonomic and Adaptive computing. The foundation of the Indus platform is a new object oriented programming language called INDUS that enables implementation of software agents and software components. The Indus platform thus enable modeling of any application as a set of concurrently executing agents that cooperatively execute tasks by coordinating with each other and composing/plugging components.

INGENIAS attempts to link analysis and design directly to implementation. INGENIAS Development Kit is a framework for analysis, design and implementation of MAS. It is based on the specification of MAS meta-models, from which tools such as the model editor and code generation are generated.

JACK is the leading edge commercial BDI-agent toolkit. It represents a legal successor of PRS and dMars, but uses an intuitive language that extends the Java programming language with certain agent specific keywords. It is the first platform with support for capabilities and generic team structures, but does not yet support the FIPA-specifications.

JACKDAW is a robust and practical multi-agent framework built to support ubiquitous computing. It is an ideal vehicle for anyone developing IT applications involving high volumes of dynamic, real-world data and requiring a high performance, scalable, flexible architecture accessible by mobile users. Calico Jack undertakes rele-

vant contract work and currently has a significant contract with Orange to develop a Jackdaw application.

JADE is a FIPA-compliant stable and efficient open-source agent platform that is developed by the TILAB. JADE is widely used as well in research as in commercial projects and has a very active user and developer community.

JADEX is a Java based framework that allows the creation of goal oriented agents and provides a framework and a set of development tools to simplify the creation and testing of agents. Jadex is build on top of JADE and therefore uses the basic JADE functionalities.

JAM is a BDI-agent platform implemented in Java and represents the successor of the UMPRS System constructed by the University of Michigan. The framework is not further developed and is not FIPA-compliant.

JASON is a fully-fledged interpreter for an extended version of AgentSpeak, a BDI agent-oriented logic programming language, and is implemented in Java. Using SACI, a multi-agent system can be distributed over a network effortlessly.

JATLITE (Java Agent Template, Lite) is a package of programs written in the Java language that allow users to quickly create new software "agents" that communicate robustly over the Internet. JATLite alone does not endow agents with specific capabilities beyond those needed for communication and interaction. In particular, JATLite does not, by itself, construct "intelligent agents".

JATLITEBEAN. A Java Bean component that encapsulates JATLite. It allows you to quickly build agents that communication over the Internet and execute in your browser. The agents speak KQML. The software is released under the GNU GPL.

JESS is a rule engine and scripting environment written entirely in Java by Ernest Friedman-Hill at Sandia (Livermore). Jess was originally inspired by the CLIPS expert system shell, but has grown into a complete, distinct Java-influenced environment.

JIAC. The goal of this project is to develop and support agent technology based on Serviceware frameworks, allowing telecommunications and telematics services to be quickly and effectively implemented and administered.

JINI. Jini network technology, which includes JavaSpaces Technology and Jini extensible remote invocation (Jini ERI), is an open architecture that enables developers to create network-centric services -- whether implemented in hardware or software -- that are highly adaptive to change. Jini technology can be used to build adaptive networks that are scalable, evolvable and flexible as typically required in dynamic computing environments.

KAARIBOGA Mobile Agents. Kaariboga is a free implementation of a framework for mobile agents. These are programs that are capable to travel from one computer to another. Kaariboga is implemented in Java. Thus it can be used on every computer supporting the Java platform. Note that kaariboga is work in progress. This could mean many bugs and many changes in the future.

LIVING MARKETS A commercially available agent server, living markets is the basis for our Adaptive Execution Software which enables companies to optimize their business processes and business relationships in real-time. The functional spectrum of Adaptive Execution Software encompasses all typical logistics and trading processes from transport management through to clearing & settlement.

LOST WAX has created one of the first systems specifically designed for the commercial enterprise. The Lost Wax Agent Framework provides a powerful and easy to use environment for the design, development and deployment of multi-agent systems. It has been designed from the start for scalability and allows the parallel execution of hundreds of agent processes distributed across multiple servers.

MADKIT is a modular and scalable multi-agent platform written in Java and built upon the AGR (Agent/Group/Role) organizational model: agents are situated in groups and play roles. MadKit allows high heterogeneity in agent architectures and communication languages, and various customizations. MadKit communication is based on a peer to peer mechanism, and allows developers to quickly develop distributed applications using multi-agent principles.

MAML (Multi-Agent Modeling Language) is an easy-to-learn language for writing models. It is only in a pre-beta phase now, but we would like to share it with possible users: evaluation of the current version can help us make progress in its development. Although the language is still in alpha stage, they do believe that, by using the

MAML-defined structure and the MAML keywords, the task of creating a simulation in Swarm becomes much easier for both programmers and non-programmers. Note that the current version of the language should be used with some programming background, however, the work to date although unfinished, has already significantly lowered the level of programming knowledge necessary.

MASSYVE KIT. The version 2.1 provides a small, easy, and interactive tool for the fast development of simple applications of multi-agent systems. It is essentially oriented towards the education and training activities, although in principle it can also be used to develop non-sophisticated multi-agent systems for various application domains.

NARVAL is a personal network assistant based on artificial intelligence and agent technologies. It executes recipes (sequences of actions) to perform tasks. It is easy to specify a new action using XML and to implement it using Python.

NUIN is a Java framework for building belief-desire-intention agents, with a particular emphasis on Semantic Web agents.

REPAST. Repast borrows many concepts from the Swarm agent-based modeling toolkit. Repast is differentiated from Swarm since Repast has multiple pure implementations in several languages and built-in adaptive features such as genetic algorithms and regression.

RETSINA. Multi-agent system infrastructure and has applied that infrastructure and its agents to many domains, including financial portfolio management; personalized web information management; book-buying auctions; logistics planning in military operations; and wireless, and mobile communications.

SAGE. The project aims to develop a distributed decentralized, fault tolerant, scalable and lightweight agent platform according to the new FIPA specifications.

SEMOA stands for "Secure Mobile Agents". It is about developing an extensible and open server for mobile agents. The server is written in Java, and agents can be written in Java as well (JDK 1.3). The focus is on all aspects of mobile agent security, including protection of mobile agents against malicious hosts.

SIM_AGENT is a free, open source, very flexible toolkit for designing agents with complex hybrid architectures (e.g. a mixture of rule-based and sub-symbolic mechanisms). The mechanisms allow control over resource allocations between agent components. There is no commitment to any particular agent architecture, but users can contribute libraries supporting particular sorts of components and architectures.

SOAR is a general cognitive architecture for developing systems that exhibit intelligent behavior. Achieving human-level reasoning and decision-making for autonomous systems requires agents that are capable of reasoning through large volumes of knowledge. A key element is the ability to resolve conflicts, solve problems, and operate in ambiguous and uncertain situations in the same way as a human expert. At Soar Technology, we develop these agents for use in training systems, exploratory experimentation, and for embedded control of unmanned and robotic systems.

SPARK (SRI Procedural Agent Realization Kit) is a new agent framework, under development at the Artificial Intelligence Center of SRI International. SPARK builds on the success of its predecessor, PRS, and shares the same Belief Desire Intention (BDI) model of rationality. SPARK has been developed to support the construction of practical agent systems, and contains sophisticated mechanisms for encoding and controlling agent behavior. At the same time, SPARK has a well-defined semantic model that is intended to support reasoning about the agents' knowledge and execution.

STARLOGO is a programmable modeling environment for exploring the workings of decentralized systems. -- systems that are organized without an organizer, coordinated without a coordinator. With StarLogo, you can model (and gain insights into) many real-life phenomena, such as bird flocks, traffic jams, ant colonies, and market economies.

SWARM is a software package for multi-agent simulation of complex systems and intended as a useful tool for researchers across disciplines. Swarm software comprises a set of code libraries which enable simulations of agent based models to be written in the Objective-C or Java computer languages. These libraries will work on a very wide range of computer platforms. The basic architecture of Swarm is the simulation of collections of concurrently interacting agents: with this architecture, we can implement a large variety of agent based models.

TUCSON, Tuple Centre Spread over the Network is a framework for MAS coordination, based on a model and a related infrastructure providing general-purpose / programmable services for supporting agent communication and coordination. The model is based on *tuple centres*, as runtime programmable abstractions whose coordinating behaviour can be dynamically specified with a logic-based language called ReSpecT.

VOYAGER A standards neutral, 100% Pure Java development platform and object request broker (ORB) for distributed computing; speeds development and improves the performance and quality of enterprise solutions. They claim to be the industry leading, distributed development platform for Java applications with mobile agent technology.

XRAPTOR. A Simulation Environment for ContinuousVirtual Multi-Agent Systems (Version 7.3.8a). XRaptor is an environment for simulation of scenarios in continuous virtual multi-agent worlds. It is written in C++ for UNIX platforms with the X Window System and Motif 1.2. XRaptor allows studying the behavior of agents in different 2- or 3-dimensional continuous worlds.

ZEUS. The ZEUS toolkit is a synthesis of established agent technologies with some novel solutions that provide an integrated environment for the rapid development of collaborative agent applications.

3. Methodologies and standards for MAS development

3.1 Methodologies

This section describes the methodologies needed for the development of some Multi-agent systems. There were two main criteria chosen for the selection of these methodologies. The first was the use of different views for the system specifications and the second the integration of engineering techniques and agent theory [12].

- VOWEL ENGINEERING [Demazeau 95] was one of the first methodologies in considering different aspects (agents, environment, interactions and organizations) in the development of MAS.

- MAS-COMMONKADS [Iglesias et al. 98] due its origins (CommonKADS [Tansley y Hayball 93]), it is a methodology oriented to the development using practical experience in expert systems.

- BDI [Kinny y Georgeff 97] has influenced the way to conceive the control of agents.

- ZEUS [Nwana et al. 99] it has its own tool that supports the methodology. It provides a platform for agents execution, agents prototypes and components.

- MaSE [DeLoach 01], it has its own tool that supports the methodology. The primary focus of MaSE is to help a designer take an initial set of requirements and analyze, design, and implement a working multi-agent system.

- GAIA [Wooldridge, Jennings y Kinny 00], studies the views definitions in a methodology and tries to integrate a software life cycle.

- MESSAGE [Caire et al. 02], defines the elements to take into account in the development of a MAS by means of the specification of meta-models. It proposes the adoption of a standard process, the RUP.

3.2 Standards

There are some standards for communication between agents including FIPA, KQML and KIF:

FIPA was originally formed as a Swiss based organization in 1996 to produce software standards specifications for heterogeneous and interacting agents and agent based systems. FIPA specifications represent a collection of standards which are intended to promote the interoperation of heterogeneous agents and the services that they can represent. In 2002, FIPA completed a process of standardising a sub-set of all its specifications.

The complete set of specifications including the ones that did not or have not yet made it to standardisation can be viewed in terms of different categories: agent communication, agent transport, agent management, abstract architecture and applications. Of these categories, agent communication is the core category at the heart of the

FIPA multi-agent system model. A repository of the FIPA specifications can be found in [6].

KQML or the Knowledge Query and Manipulation Language is a language and protocol for exchanging information and knowledge. It is part of a larger effort, the ARPA Knowledge Sharing Effort which is aimed at developing techniques and methodology for building large-scale knowledge bases which are sharable and reusable. KQLM is both a message format and a message-handling protocol to support run-time knowledge sharing among agents for co-operative problem solving. KQML can be used as a language for an application program to interact with an intelligent system or for two or more intelligent systems to share knowledge in support of cooperative problem solving.

KQML focuses on an extensible set of performatives, which defines the permissible operations that agents may attempt on each other's knowledge and goal stores. The ideas which underlie the evolving design of KQML are currently being explored through experimental prototype systems which are being used to support several testbeds in such areas as concurrent engineering, intelligent design and intelligent planning and scheduling [7].

KIF. Knowledge Interchange Format (KIF) provides a syntax for message content, which is essentially first order predicate calculus with declarative semantics. This empowers the agents to understand the meaning of expressions in the representation without the help of an interpreter for manipulating those expressions [4]. KIF is logically comprehensive, at its most general, it provides for the expression of arbitrary logical sentences. The language also provides methods for the representation of knowledge about the representation of knowledge. This allows the user to make knowledge representation decisions explicit and permits the user to introduce new knowledge representation constructs without changing the language. The language description includes both a specification for its syntax and one for its semantics [8].

4. Simulation in Environmental Decision Support Systems

This section presents a variety of work done in modelling and simulation of Environmental decision support systems (EDSS) [9].

ICMS (The Integrated Catchment Management System) is a PC-based software product, developed to facilitate the rapid development and delivery of catchment science to catchment managers. It implements a 'layered' approach to development, and delivery, through the use of four components. These are: ICMSBuilder– the central engine; Models embedded in the ICMS MDL(model libraries) – proprietary, free-ware and newly developed models; ICMS Projects– suites of linked models, and their data; ICMS Views – DLLs which provide tailored views of Projects.

Tarsier. The Tarsier modelling framework is a modular collection of Windows software that enables fast development and deployment of a wide variety of environmental computing tools. These tools include simulation models, data storage and analysis tools, and visualization systems. The system supports many structures for the organization of quantitative environmental information, including: gridded maps, networks, time series, and simple lists of geographic locations. Upon these are built analytical tools covering topics such as interpolation, statistics, sampling, and data transformation. At the top level are modules that implement a variety of simulation models, from cellular automata to stream pollutant routing models to large-scale spatial catchment hydrology models.

The Desert Model. A DSS for water quality management in river streams, developed at IIASA, Laxenburg, Austria. DESERT is a sophisticated highly integrated tool for decision support in water quality management on a river basin scale. It integrates most of the stages, which can be found in usual decision support procedure: data management, model formulation and implementation, calibration, simulation, optimization and plotting of results. DESERT has a built-in capability for evaluation of least-cost strategies through selecting of optimal upgrading alternatives of waste water treatment plants.

RIBASIM. (River Basin Simulation Model) is a generic model package for analyzing the behaviour of river basins under various hydrological conditions. RIBASIM is a modelling instrument for river basin planning and management.

Agent-based simulation

Ps-i. An environment and a simulation language for running agent-based simulations. Models are written using the standard Tcl/Tk scripting language and a graphical interface can also be used.

SeSAm (Shell for Simulated Agent Systems) provides a generic environment for modelling and experimenting with agent-based simulation. SeSAm agents consist of a body, that contains a set of state variables and a behaviour that is implemented in form of UML-like diagram. Based on an extensive number of primitive components, a user is able to design a simulation graphically without knowing the syntax of a traditional programming language. It is written in Java and it is freely downloadable.

Table 2 shows a comparison between the simulation toolkits seen above.

EDSS	Supplier	Availability	Features	Type
Icms	Australian National University	Non-Commercial Software License Agreement	Supports the rapid building and integration of catchment models	Software tool
Tarsier	Co-operative Research Centre (CRC) for Catchment Hydrology (Australia)	Non-Commercial Software License Agreement	Allows users to create, link and execute a catchment model through a graphical representation of the model	Framework
Desert Model	IIASA, Laxenburg, Austria.	Free for academic/research purposes	Highly integrated tool for decision support for water quality management on a river basin scale	Software tool
RIBASIM	Delft Hydraulics	License Software	Decision support/expert system	Software tool
Ps-i	University of Pennsylvania	Free Software	Environment for running agent-based simulations	Language
SeSam	University of Würzburg	Free Software	Generic environment for modelling and experimenting with agent-based simulation	Software tool

Table 2. Comparison of EDSS simulation toolkits.

5. Conclusions

A review of MAS platforms available and EDSS simulation tools have been presented. There are two tables describing some fea-

tures of the products revised so we believe these tables can be used as a reference tool to help the developer in the selection of a platform according to his or her needs. Particularly, we have chosen Jadex as the agent platform for designing our River Basin MAS proposal. We chose Jadex due to its capacity to implement the behaviour of intelligent agents in an easy way within a MAS (by addressing the limitations of BDI systems).

As future work, we plan to develop the extension to Jadex platform in order to support the required functionalities of the River Basin MAS. Cased Based Reasoning and Ruled Based Reasoning are the two main reasoning models to be integrated within our MAS platform. Also some learning mechanisms to integrate knowledge to the agents are foreseen. Afterwards, some preliminary testing will be done with real data coming from the real case study.

References

1. Poch M., Comas J., Rodríguez-Roda I., Sànchez-Marrè M., Cortés U., Designing and building real environmental decision support systems*. Environmental Modelling and Software,* 19 (9), 857 - 873. ISSN: 1364-8152 (2004).
2. Devesa F., De Letter P., Poch M., Rubén C., Freixó A. and Arráez J., Development of an EDSS for the Management of the Hydraulic Infrastructure to Preserve the Water Quality in the Besòs Catchment. *Procc of EULAT WORKSHOP on e-Environment, pp- 31-46* (2005).
3. Raso J., Malgrat P., Cabot, J., Integrated Modelling to improve bathing water quality at Barcelona's beaches. *Third DHI Software Conference* (June 1999).
4. Mangina E., Review of software products for Multi-Agent Systems. Applied Intelligence (UK) Ltd for AgentLink (June 2002). www.AgentLink.org
5. Rodriguez-Roda I., Sànchez-Marrè M., Comas J., Baeza J., Colprim J., Lafuente J. Cortés U. and Poch M. A hybrid supervisory system to support WWTP operation: implementation and validation. *Water Science and Technology* Vol 45 No 4-5 pp 289-297 (2002).
6. FIPA organisation : http://www.fipa.org/repository/standardspecs.php3
7. KQML specifications: http://www.cs.umbc.edu/kqml/
8. KIF specifications: http://logic.stanford.edu/kif/dpans.html
9. INGENIAS: http://grasia.fdi.ucm.es/ingenias/Spain/estado/index.php
10. Agent Platforms: http://vsis-www.informatik.uni-hamburg.de/projects/jadex/links.php
11. A collection of Modelling and Simulation Resources on the Internet: http://www.idsia.ch/~andrea/simtools.html#enviro

12. Ivanov P., Masliev I., Somlyody L. (1995) DEcision Support system for evaluating River basin sTrategies (DESERT). Proceedings of European Simulation Multiconference, Prague June 5-7, p. 517-520
13. Rendón-Sallard T. and Sànchez-Marrè. Designing a Multi-Agent system to simulate scenarios for decision-making in river basin systems. Submitted to 19th International Conference on Industrial, Engineering & Other Applications of Applied Intelligent Systems (IEA/AIE'06), Anneccy, France 2006.