

UNIVERSIDAD NACIONAL DEL CENTRO DE LA PROVINCIA DE BUENOS AIRES
FACULTAD DE CIENCIAS EXACTAS
MAGISTER EN INGENIERÍA DE SISTEMAS

**FraMaS: Un Framework para Sistemas Multi-agente
basado en Composición**

Henri Héctor Avancini

Prof. Dra. Analía Adriana Amandi
Director

Tandil, Junio del 2000

UNIVERSIDAD NACIONAL DEL CENTRO DE LA PROVINCIA DE BUENOS AIRES
Rector: Carlos Alberto Nicolini
Vice-Rector: Eduardo Miguel
Facultad de Ciencias Exactas
Instituto de Investigación de Sistemas de Tandil (ISISTAN)
Director ISISTAN: Marcelo Campo
Junio 2000

RESUMEN

La *Ingeniería del Software basada en Agentes* ha surgido en respuesta a la demanda de los requerimientos de los Sistemas Multi-agente (SMA). Éstos sistemas son esencialmente distribuidos y están compuestos por un conjunto de *agentes de software* que interactúan a fin de satisfacer sus objetivos. Si bien los SMA son aplicados a una diversidad enorme de sistemas de computadoras (hardware o software) comparten una serie de características, entre las que se mencionan: la información es distribuida, las entidades tienen conocimiento parcial, la computación es asincrónica, no existe un sistema de control central.

Los *agentes de software* (ó *agentes inteligentes* ó simplemente *agentes*) son entidades computacionales autónomas que están dirigidas por objetivos e insertas en un entorno que pueden percibir y actuar sobre él. *Autonomía* significa que las acciones ejecutadas por el agente no son explícitamente iniciadas por el usuario u otro agente; además de que el agente persiste en el tiempo, es decir que no termina su ejecución cuando ha finalizado una tarea, sino que continúa observando su entorno, decidiendo qué acción ejecutar en el próximo instante de tiempo.

Éstas y otras características de los SMA hacen que su diseño, implementación, mantenimiento, etc. sea una tarea poco trivial. Por ello es deseable contar con herramientas que asistan en las etapas involucradas en la construcción de un SMA. Sin embargo, no existe al momento tal herramienta, más allá del gran esfuerzo en investigación dirigido en este sentido.

Una alternativa propuesta desde hace tiempo en la Ingeniería del Software consiste en reutilizar los componentes previamente construidos. Entre las técnicas de reuso empleadas en el paradigma Orientado a Objetos están la reutilización de librerías de clases, los patrones de diseño y los frameworks. Éstos últimos son el esqueleto de un conjunto de aplicaciones que pertenecen a un dominio concreto.

En particular se propone un framework para el dominio de los Sistemas Multi-agente, denominado FraMaS, que incorpora las características de los SMA. Este framework está formado por una estructura de clases e interfaces Java que modela a los SMA a través de un conjunto de *entornos multi-agente*. Cada uno de éstos entornos contiene un conjunto de agentes que interactúan autónomamente a fin de satisfacer sus objetivos usando los servicios brindados.

Cada agente diseñado usando FraMaS tiene un conjunto de *acciones básicas* que son cubiertas por la *funcionalidad avanzada*. Las acciones básicas son aquellas inherentes a la funcionalidad de agente en si y que no involucra técnicas de deliberación, negociación, etc. La idea detrás del diseño de cada agente es contar con una serie de “*decoradores*” sobre las acciones básicas que pueden ser adicionados (y eliminados) dinámicamente.

Cada uno de estos agentes está inserto en un solo entorno simultáneamente, aunque puede también mudarse de uno a otro. Los agentes perciben eventos que ocurren en otros agentes y en el entorno. La comunicación entre agentes es directa.

El framework presentado es viable para la construcción de Sistemas Multi-agente en función de las implementaciones realizadas: asistente personal, sistema de robots que interactúan a fin de satisfacer un objetivo global y la estructura de agentes para comercio electrónico.

FraMaS fue desarrollado con un enfoque *bottom-up*, es decir desde las aplicaciones al diseño. El mismo permite la *adición de nuevos componentes* a través de una serie de métodos incorporados, que simplifican la extensión de la funcionalidad del framework. Luego, otras aplicaciones construidas usando el framework pueden hacer uso de estos componentes.

Palabras claves: Sistemas Multi-agente, framework, Internet, agentes inteligentes, Ingeniería del Software Basada en Agentes.

ABSTRACT

Agent-based Software Engineering has been developed in order to satisfy the requirements of Multi-agent Systems (MAS). MAS basically are distributed and are composed of a set of *software agents* that interact in order to satisfy their objectives. Although MAS are applied to a great variety of different computer systems (hardware and/or software), they share a number of characteristics and components, i.e. the distributed information, entities with partial knowledge, no central control system.

Software agents (or *intelligent agents* or simply *agents*) are autonomous computational entities driven by objectives and inserted in an environment, which can be perceived and in which they can act. By *autonomy* we mean the set of actions performed by the agent without explicit user indication or other entity indication. Moreover, an agent is autonomous if it persists over time, i.e. it does not complete its execution when finished with an action; in fact, it continues perceiving its environment, deciding what action to carry out next, etc.

These and other characteristics of MAS make their development a non-trivial task. It is therefore desirable to have tools to assist with the different stages involved in the construction of MAS. However, despite the research efforts in that area, there are no tools at present.

The reuse of previously constructed software components is a common proposal in Software Engineering. The most common reuse techniques, in the Object Oriented paradigm, are the reuse of libraries of classes, design patterns, and frameworks. Frameworks are the skeleton of a set of applications that belong to a specific domain.

In particular, we propose a framework for the Multi-agent Systems domain called FraMaS, implemented in Java, which incorporates the characteristics of MAS systems. The FraMaS structure is made up of a set of Java classes and interfaces that model Multi-agent Systems by means of a set of multi-agent environments. Each one of these environments contains a set of agents that interact autonomously in order to satisfy their objectives through the use of the services provided by the environment.

Each agent designed under FraMaS has a set of *basic actions* covered by the *advanced functionality*. Actions that do not use any kind of deliberation techniques, learning, negotiation, etc. are basic agent actions; e.g. robot move action is a basic action, whereas the planning of the move action is an advanced functionality. The idea behind the design of each agent is to endow the agent with a set of wrappers over the basic actions that can be added (and deleted) dynamically.

Agents are inserted in only one environment in each time stamp, though they can move from one environment to another. Events that occur in the agent context (i.e. in other agents or the environment) are perceived by sensors. The communication between agents is direct.

The presented framework (FraMaS) is viable for the construction of multi-agent systems according to our experience with the design, implementation and testing of the following software systems: Meeting Scheduling Agent, Forklift agents' system, and e-commerce agent structure.

FraMaS was developed using a bottom-up approach, i.e. from applications to the design. This permits the inclusion of new components by means of a series of implemented methods that simplify the extension of the framework functionality. Therefore, other applications built using the framework can make use of these components.

Key-words: Multi-agent Systems, Framework, Internet, Intelligent Agents, Agent-based Software Engineering.