

---

# Tecnologías Sistema Multiagente de Simulación para la Planificación de Proyectos Software

---

Nathaly Arribasplata – Carlos Cánepa Pérez

Universidad Nacional Mayor de San Marcos  
Facultad de Ingeniería de Sistemas e Informática  
Av. Germán Amézaga s/n, Ciudad Universitaria, Lima 01, Lima, Perú

narribasplata@gmail.com, ccanepap@unmsm.edu.pe

---

## RESUMEN

El presente estudio tiene como objetivo proveer un sistema capaz de resolver de forma estratégica los problemas relacionados con la planificación de proyectos informáticos, dando mejores estimaciones que las obtenidas de los métodos actuales, basándose tanto en conocimiento como en información y en la simulación, mediante multiagentes, de las tareas y roles que participan en los proyectos de software en base al Proceso Unificado de Desarrollo de Software. Adicionalmente, se busca proveer un esquema para el desarrollo del proyecto de software, basado en las mejores prácticas de desarrollo.

**Palabras clave:** Planificación de proyectos de software, sistemas multiagente, proceso unificado de desarrollo de software, simulación de sociedades.

## ABSTRACT

The present study aims to provide a system capable of strategically resolving problems related with IT project planning, by giving better estimations than those derived from current planning methods, based on knowledge, information and simulation of the tasks and roles in software project development based on the Unified Process Software Development, using multi-agents. Additionally, it aims to provide an outline for the development of the software project, based on the best practices for IT project development.

**Keywords:** Software project planning, multi-agent systems, unified software development process, simulation societies

---

## 1. INTRODUCCIÓN

El desarrollo de proyectos de software se compone de una serie de actividades que se encuentran integradas en el sentido de que una determinada acción desarrollada en un área afectará muy posiblemente a otras áreas. Así, la gestión de proyectos trata la planificación y control de procesos, recursos, estructuras e interrelaciones integrados, además del seguimiento y control de riesgos, por lo cual es una tarea compleja. A pesar de los avances existentes en diferentes campos del desarrollo de proyectos software, el número de fracasos sigue siendo desalentador [Puche+ 2006].

De manera general, estos proyectos software están afectados por tres graves problemas: costes muy por encima de lo presupuestado, entrega fuera de plazo y falta de calidad del producto, implicando que el número de fracasos en estos proyectos no desciende suficientemente. Por tanto, parece clara la necesidad de aportaciones que permitan construir software de manera que se asegure su calidad, fiabilidad, sencillez y robustez, a un coste y plazo de entrega razonables [Puche+ 2006].

En esta propuesta se busca implementar un sistema de simulación de la implementación de proyectos de software basado en multiagentes inteligentes, tomando tanto información del proyecto así como conocimiento sobre desarrollo de proyectos de TI, con base en el Proceso Unificado de Desarrollo de Software [Jacobson+ 1999]. De este modo se trata de reemplazar las metodologías actuales de planificación de proyectos informáticos, con una herramienta de simulación basada en conocimiento, que brindará resultados en menor tiempo que el que se toma actualmente en este tipo de planificaciones, lo cual permitirá obtener las estimaciones de riesgos, costos y beneficios del proyecto de forma más rápida y precisa, además de poder hacer los ajustes necesarios a los requerimientos en caso el proyecto resulte no rentable, o en su defecto la cancelación del proyecto sin haber gastado tiempo excesivo en su planificación.

Adicionalmente permitirá que los resultados de cada simulación se almacenen de modo que sirvan de conocimiento base para futuras simulaciones de proyectos similares. En esto último, se reemplazaría progresivamente

modelos como el de COCOMO [Boehm 1981], que se basan en información de proyectos que corresponden a una realidad diferente a la de nuestro país.

El proyecto propuesto en base a multiagentes inteligentes favorece la comprensión de esta naturaleza integrada de la gestión de proyectos al describirla a través de sus procesos, estructuras e interrelaciones principales. Esto es útil especialmente debido a que, sólo quien va a desarrollar la tarea puede planificarla. Quien va a desarrollar es quien sabe los pasos que hay que dar para llegar a destino y tiene la experiencia necesaria para la estimación. El comportamiento de estos roles puede ser modelado por los agentes.

Se puede deducir que los modelos de simulación están indicados especialmente para tratar problemas situados en el nivel estratégico de los proyectos, mientras que los métodos tradicionales resultan útiles en el nivel operacional.

El sistema planteado tomará los requisitos de los clientes, y los tomará como los objetivos para la simulación. Su funcionamiento consistirá en simular todas las actividades a realizarse para satisfacer dichos requerimientos basándose también en los recursos disponibles en la organización para el proyecto y proporcionando al final de la simulación los recursos faltantes (tiempo, personal, costo, tecnologías, infraestructura, capacidades, etc.) para que el proyecto se realice exitosamente, de manera rentable, óptima y con una buena planificación de riesgos; así como la planificación óptima con los recursos actuales.

Otro punto importante que incluye esta propuesta, y que deriva de su funcionamiento principal (la simulación) es el de proveer de un esquema para el desarrollo del proyecto de software simulado. La simulación realizada para la planificación representará las actividades a realizarse posteriormente, y esta representación utilizará conocimiento de las mejores prácticas para el desarrollo de estas actividades, con lo cual no sólo se proporcionará como consecuencia un marco de trabajo óptimo, si no que será necesario utilizar dicho marco – en caso se decida seguir adelante con el proyecto – de modo que los resultados reales obtenidos durante y al final

del proyecto sean coherentes con los resultados de la simulación

El trabajo está organizado en 4 secciones. En la siguiente sección se mencionan algunos modelos y metodologías utilizadas para la planificación automatizada de proyectos de software. En la sección 3 se presenta el diseño del sistema multiagente de simulación para la planificación de proyectos de software; en donde se describen la conceptualización, y los modelos de organización, agentes, de interacción, y entorno. Las conclusiones del presente trabajo siguen en la sección 4.

## **2. ENFOQUES DE AUTOMATIZACIÓN DEL PROCESO DE PLANIFICACIÓN DE PROYECTOS SOFTWARE**

A continuación se presentan los enfoques más importantes de automatización del proceso de planificación de proyectos informáticos

### **2.1 Dinámica de Sistemas**

La metodología de dinámica de sistemas fue creada por Forrester [Forrester 1961] para el modelamiento de sistemas socio-económicos complejos basados en los principios de la teoría de control por retroalimentación. Existen varias aplicaciones de esta metodología a la gestión de proyectos de software.

La primera aplicación fue propuesta en [Abdel-Hamid+ 1982], dirigida al desarrollo de un modelo genérico de para el proceso de desarrollo de software. Trabajos similares se realizaron en el Laboratorio de propulsión de la Nasa [Tausworthe+ 1983] y luego se propuso un modelo de dinámica de sistemas para el proceso de desarrollo de software [McKenzie 1984]. En [K.Cooper+ 1993] el enfoque se dirige a la evaluación del impacto de la calidad del proceso. El trabajo de [Abdel-Hamid+ 1991] provee un estudio de la interacción humana en un proceso de software, en particular de los factores que afectan la productividad (aprendizaje, comunicación y uso del tiempo ocioso). En [Arango+ 2001] se detallan los conductores metodológicos que asociados a la gestión integral de los proyectos informáticos en un entorno exigente permiten esbozar factores críticos y signos vitales que optimizan las actividades de los mismos. Los modelos mentales mostrados en dicha investigación, y su transformación conjuntamente con los de la

empresa pueden ser utilizados en cualquier ámbito, donde se trabaje con procesos basados en las tecnologías de la información.

### **2.2 Cibernética Organizacional**

En [Puche+ 2006] se propone un enfoque basado en la Cibernética Organizacional, aplicando VSM (Modelo de los Sistemas Viables), para diseñar organizaciones que desarrollan software que sean viables, es decir, que estén dotadas de las capacidades de regulación, aprendizaje, adaptación y evolución necesarias para garantizar su "supervivencia" ante los cambios que puedan producirse en su entorno (incluso aunque éstos no hayan sido previstos al inicio de los proyectos). En el VSM se establecen las condiciones necesarias y suficientes para que un sistema sea viable, es decir, capaz de mantener una existencia independiente. Para garantizar esa "viabilidad", el VSM permite abordar la enorme complejidad (variedad) a la que se enfrentan las organizaciones. Para ello realiza un desdoblamiento de la complejidad que se acompaña con el diseño de estructuras organizativas, cada una de las cuales se ocupa de partes de la misma. Se aplica el VSM para asegurar la capacidad de existencia independiente, auto-regulación, aprendizaje y adaptación en el desarrollo de proyectos software. Entre las principales cuestiones contempladas a la hora de organizar un proyecto de este tipo con el VSM se encuentran: la calidad del software, desarrollo externo, equipos de trabajo, experiencia, formación, estimación, herramientas de desarrollo, motivación, reutilización, cambios en los requerimientos, etc.

## **3. DISEÑO DE MULTIAGENTES DE SIMULACIÓN PARA PLANIFICACIÓN DE PROYECTOS SOFTWARE**

Se eligió la tecnología de Multiagentes para la implementación del sistema de simulación de proyectos de software, como alternativa para la planificación de este tipo de proyectos.

Mediante sistemas multiagente (SMA), se puede representar agentes inteligentes que toman decisiones de forma independiente y autónoma, además se puede modelar la interacción entre ellos. Esta tecnología permite realizar simulaciones así como crear agentes que actúen de acuerdo a intereses humanos y objetivos para los cuales son diseñados.

La metodología elegida para el análisis y diseño del sistema basado en agentes incluido en este proyecto es INGENIAS [Pérez 2007]. INGENIAS concibe el SMA como la representación computacional de un conjunto de modelos. Cada uno de estos modelos muestra una visión parcial del SMA: los agentes que lo componen, las interacciones que existen entre ellos, cómo se organizan para proporcionar la funcionalidad del sistema, qué información es relevante en el dominio y cómo es el entorno en el que se ubica el sistema a desarrollar. Estos modelos o meta-modelos indican qué hace falta para describir: agentes aislados, organizaciones de agentes, el entorno, interacciones entre agentes o roles, tareas y objetivos. Se construyen mediante un lenguaje de meta-modelado, el GOPRR (Graph, Object, Property, Relationship, and Role). En la construcción de estos meta-modelos se integran resultados de investigación en forma de entidades y relaciones entre entidades. La instanciación de estos meta-modelos produce diagramas, los modelos, similares a los que se usa en UML, con la diferencia de que estos diagramas se han creado exclusivamente para definir el sistema multiagente.

A continuación se presenta en forma concisa la aplicación de INGENIAS para desarrollar un sistema de simulación del proceso de desarrollo de un proyecto informático, que represente y modele las diversas tareas realizadas en este tipo de proyectos, de forma óptima, de modo que los resultados de la simulación sirvan de guía para la gestión de recursos y riesgos de proyectos de software.

Se modelará el proceso de desarrollo desde la fase de diseño, puesto que tomará como entrada el modelo de análisis con los casos de uso bien definidos.

Las etapas de diseño, implementación, pruebas de calidad y despliegue estarán llenas de diferentes tipos de riesgos que serán analizados por el sistema (experiencia de programadores, sus conocimientos de las herramientas, conocimientos del negocio, conocimiento de los usuarios finales, recursos disponibles o necesarios para el despliegue del sistema, etc.), mediante la simulación con agentes. Además de la simulación de riesgos, se generaran los plazos, recursos necesarios, personal necesario, adecuándose a los recursos y presupuesto disponibles para el proyecto.

### 3.1 Conceptuación

El objetivo de esta fase es obtener una primera aproximación al sistema que queremos desarrollar.

#### 3.1.1 Identificación de los actores

Identificamos a los siguientes actores o roles: el actor Ingeniero de CUS, el actor Arquitecto, el actor Integrador del Sistema, el actor Ingeniero de Componentes, y el actor Ingeniero de pruebas, Ingeniero de pruebas de integración, Ingeniero de pruebas de sistema, Encargado despliegue y el actor de Entrada de modelo de análisis.

#### 3.1.2 Identificación de los casos de uso

En base al Proceso Unificado de desarrollo de software, y tomando en cuenta que la entrada del sistema son los casos de uso ya bien definidos obtenidos del modelo de análisis, se encuentran los siguientes casos de uso, para cada uno de los actores identificados.

- Actor Ingeniero de CUS: Diseño del nuevo sistema.
- Actor Arquitecto: Diseño e implementación del nuevo sistema.
- Actor Integrador del Sistema: Implementación del nuevo sistema.
- Actor Ingeniero de Componentes: diseño, implementación y pruebas al nuevo sistema.
- Actor Ingeniero de pruebas: Pruebas al nuevo sistema.
- Actor Ingeniero de pruebas de integración: Pruebas al nuevo sistema.
- Actor Ingeniero de pruebas de sistema: Pruebas al nuevo sistema.
- Actor Encargado despliegue: Despliegue del nuevo sistema.
- Actor Entrada de modelo de Análisis: Interfaz con el usuario del simulador.

### 3.2 Modelo de organización

El contexto de este problema corresponde a una realidad simulada. Dicha realidad comprende todas las actividades del proceso de desarrollo de un proyecto de software basado en el Proceso Unificado de desarrollo de software en el ámbito comprendido dentro del área de Sistemas de una organización.

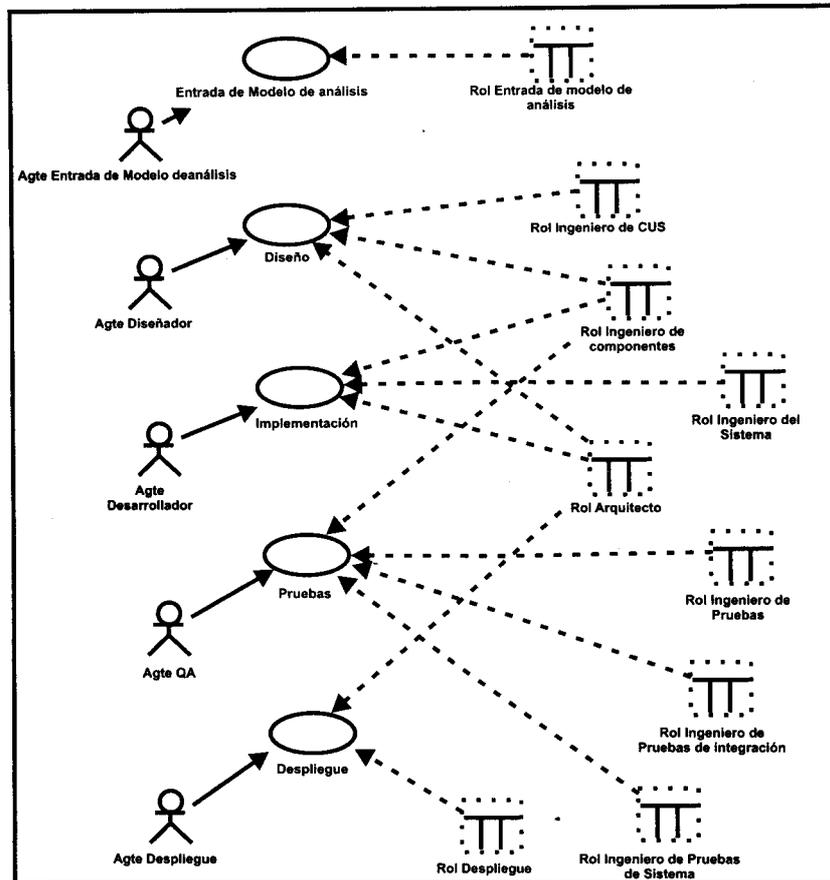


Figura 1 Modelo de casos de uso

Las funciones a simular en el sistema son las siguientes:

- **Entrada de modelo de Análisis:** Función que comunicará al usuario del sistema con el agente de diseño que iniciará la simulación, mediante el paso del modelo de análisis.
- **Diseño:** Se considerarán todas las actividades relacionadas con la disciplina de diseño de RUP.
- **Implementación:** Se tomarán en cuenta todas las actividades relacionadas con la disciplina de implementación de RUP.
- **Prueba:** Se tomará en cuenta todas las actividades relacionadas con la disciplina de análisis de RUP.
- **Despliegue:** Se modelarán todas las actividades relacionadas con la etapa de despliegue de RUP.

Se está considerando los recursos: computadoras, bases de datos, espacio de oficina, presupuesto, tiempo, personas, conocimiento, modelo de análisis.

### 3.3 Modelo de agentes

Como se ve a continuación, se han identificado 5 agentes, cada uno de los cuales tiene determinados roles, que comparte con el resto de agentes. Sin embargo, dos agentes que comparten un mismo rol, no realizan las mismas tareas asignadas a ese rol. Por ejemplo, el agente analista tiene el rol de Ingeniero de CUS, y realiza las tareas de este rol relacionadas con el modelo de análisis de casos de uso. El agente diseñador que tiene este mismo rol, realiza las tareas de diseño de casos de uso correspondientes a este rol.

La búsqueda de distribución en el problema es útil en este caso. El conocimiento de desarrollo de sistemas de software debe estar distribuido en cada uno de estos agentes. También se encuentra distribución en la información que cada uno de los agentes necesitará y generará.

- **Agente Entrada de Modelo de análisis**  
**Roles:** Entrada de Modelo de análisis.
- **Agente Diseñador**  
**Roles:** Ingeniero de CUS, arquitecto, ingeniero de componentes, encargado de despliegue.

- **Agente Desarrollador**  
**Roles:** Arquitecto, integrador de sistema, ingeniero de componentes.
- **Agente QA**  
**Roles:** Ingeniero de pruebas, ingeniero de pruebas de integración, ingeniero de pruebas de sistema, ingeniero de componentes.
- **Agente Despliegue**  
**Roles:** Encargado de despliegue.

Agente Entrada de Modelo de Análisis				
Objetivos	Planes	Conocimiento	Colaborador	Servicio
Ingreso del modelo de análisis	<ul style="list-style-type: none"> <li>• Lectura del modelo de análisis a partir del usuario del sistema</li> <li>• Utilizar estándares de definición formal de modelo de análisis para comunicarse con el usuario final</li> </ul>	<ul style="list-style-type: none"> <li>• Estándares de definición formal de modelo de análisis.</li> <li>• Entradas de recursos a modo de restricción</li> </ul>	Usuario del sistema simulador	<ul style="list-style-type: none"> <li>• Comunica modelo de análisis</li> <li>• Comunica descripción de arquitectura</li> <li>• Comunica modelo de casos de uso - análisis</li> <li>• Comunica definición clases de análisis</li> <li>• Comunica los paquetes de Análisis</li> <li>• Comunica entradas de restricción del proyecto</li> </ul>
Ingreso de la descripción de la arquitectura				
Ingreso de la realización de casos de uso – análisis				
Ingreso de la definición clases de análisis				
Ingreso de los paquetes de Análisis				
Comunicación del modelo de análisis				
Comunicación de la descripción de la arquitectura				
Comunicación de la realización de casos de uso – análisis				
Comunicación de la definición clases de análisis				
Comunicación de los paquetes de análisis				

Tabla 1. Plantilla de descripción de agentes

### 3.4 MODELO DE INTERACCIÓN

El modelo de interacción refleja el comportamiento que deben adoptar los agentes al recibir instrucciones de otro agente o de un humano. En la Figura 2 se muestra el diagrama de interacción general.

### 3.5 MODELO DE ENTORNO

Este modelo especifica a las entidades del entorno en el que se encuentra el sistema de agentes, ya sean recursos consumibles o no consumibles, aplicaciones, o agentes. Además comprende todo lo relacionado a los riesgos a los que el proyecto de software a simular se expone en cualquiera de sus etapas de desarrollo.

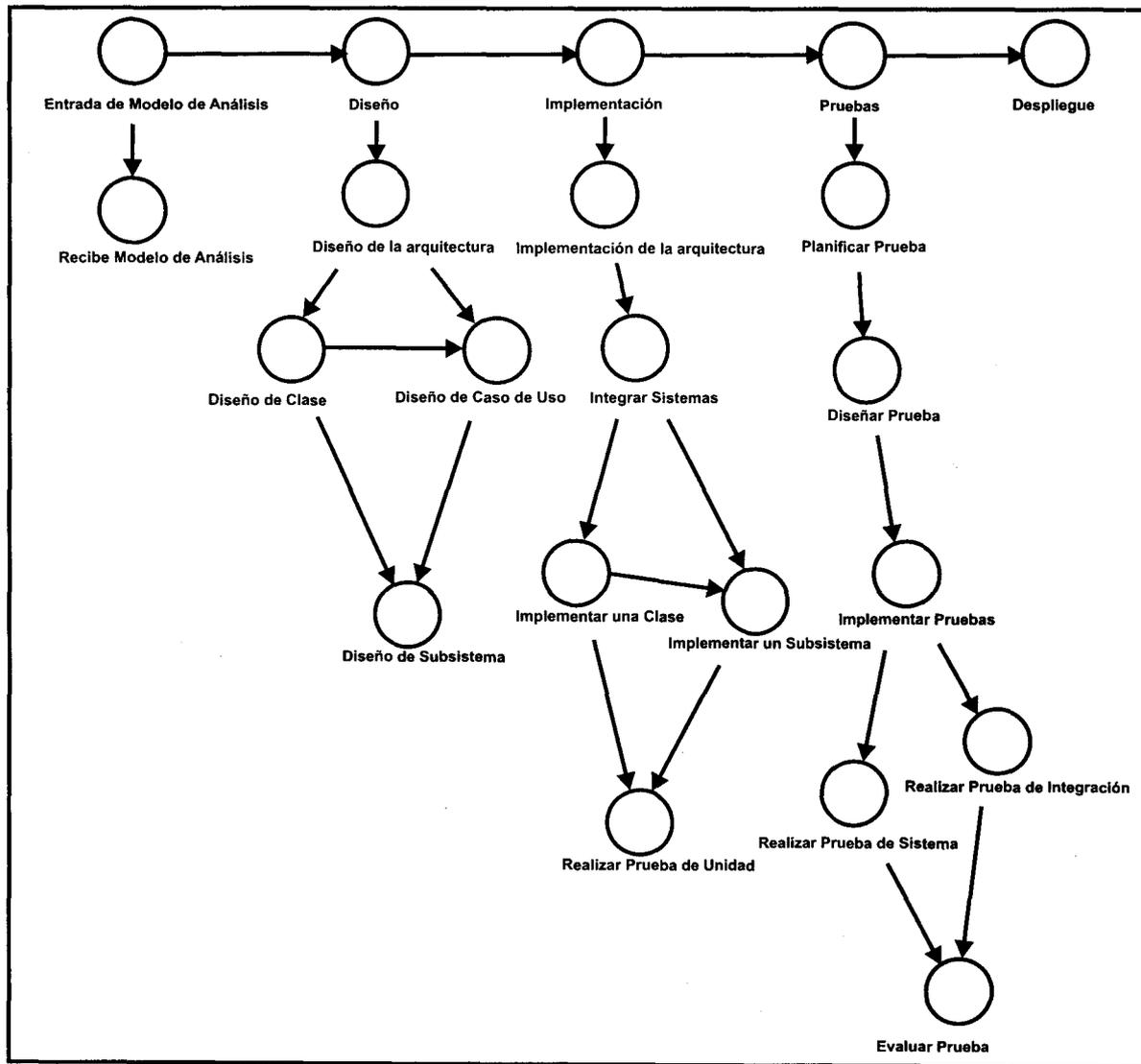


Figura 2 plantilla Modelo de Interacción

Los riesgos que son simplemente imposibles de predecir con los métodos comunes, pueden ser predichos por este sistema. Los riesgos conocidos son todos aquellos que se pueden descubrir después de una cuidadosa evaluación del plan del proyecto del entorno técnico y comercial en el que se desarrolla el proyecto y otras fuentes de información fiables (fechas de entrega poco realistas, falta de especificación de requisitos o de ámbito del software o un entorno pobre de desarrollo), los riesgos predecibles se extrapolan de la experiencia en proyectos anteriores (cambio de personal, mala

comunicación con el cliente, disminución del esfuerzo del personal a medida que atienden peticiones de mantenimiento). Pueden ocurrir pero son extremadamente difíciles de identificar por adelantado. En este sentido el sistema de simulación proporciona un gran aporte.

La identificación del riesgo es un intento sistemático para especificar las amenazas al plan del proyecto (estimaciones, planificación temporal, carga de recursos, etc.). Identificando los riesgos conocidos y predecibles, el gestor del proyecto da un paso adelante para evitarlos

cuando sea posible y controlarlos cuando sea necesario.

#### 4. CONCLUSIONES

La presente investigación propone la implementación de un sistema en base a multiagentes inteligentes para la simulación de proyectos de software. Puesto que esta tecnología favorece la comprensión de la naturaleza distribuida e integrada de la gestión de proyectos al describirla a través de sus procesos, estructuras e interrelaciones principales. Entre otras utilidades, la tecnología de multiagentes permite realizar simulaciones de sociedades y realidades complejas como la que se vive en un proceso de desarrollo de software. El comportamiento de los roles participantes en este tipo de proyectos puede ser modelado por los agentes y simulado de modo que logren un uso óptimo de recursos, para lo cual se tiene en cuenta los riesgos a los que se puede exponer el proyecto de software.

Este estudio se podría complementar con investigaciones que modelen a través de SMA el comportamiento de los usuarios. Si las especificaciones constaran del modelado de los usuarios, incluyendo las obligaciones cognitivas y las preferencias de los usuarios, entonces se podrían desarrollar las soluciones de acuerdo a sus requerimientos.

#### REFERENCIAS

- [Puche+ 2006] J. Puche, J. Pérez, P. Mayoral. Aplicación de la Cibernética Organizacional mediante VSMoD al estudio de un Proyecto Software. Actas del X Congreso de Ingeniería de Organización Valencia. España. 2006.
- [Jacobson+ 1999] I. Jacobson, G. Booch, J. Rumbaugh. El Proceso Unificado de Desarrollo de Software. Rational Software Corporation. 1999.
- [Boehm 1981] B.W. Boehm. Software Engineering Economics. Prentice-Hall, Inc. Englewood Cliffs, New Jersey. 1981
- [Forrester 1961] J. Forrester. Industrial Dynamics, Cambridge, MA: Productivity Press. 1961.
- [Abdel-Hamid+ 1982] T. Abdel-Hamid, S. Madnick. A model of software project management dynamics. Proceeding of The 6th International Computer Software and Applications Conference (COMPSAC). 1982.
- [Abdel-Hamid+ 1991] T. Abdel-Hamid and S. Madnick. Software Project Dynamics: An Integrated Approach. New Jersey, Prentice-Hall. 1991
- [Tausworthe+ 1983] R. Tausworthe, M. McKenzie, C. Lin. Structural considerations for a software life-cycle dynamic simulation model. Presented at AIAA Computers in Aerospace IV Conf. 1983.
- [McKenzie 1984] M. McKenzie. A dynamic system simulation model of the software development process. Summer Computer Simulation Conf. (SCSC). 1984
- [Cooper+ 1993] K. Cooper, T. Mullen. Swords and Plowshares: The Rework Cycles of Defence and Commercial Software Development Projects. American Programmer. 1993.
- [Arango+ 2001] M. Arango; H. Gómez, J. Montesa. La Gestión Sistemática de Proyectos Informáticos Utilizando Tecnologías de la Información. XVII Congreso Nacional de Ingeniería de Proyectos - Murcia. 2001.
- [Pérez 2007] Y. Pérez. Aplicación de Metodologías Ingenieras, Zeus, Masina al Desarrollo de Sistemas Multi-Agente, Partiendo de SMA de Subastas Para la Identificación de Mejores Practicas. Tesis de Ingeniería. Universidad de Pamplona. Colombia. 2007.