

Sistemas de Software Distribuido

P. Pesado^(1,2), H. Ramón⁽¹⁾, P. Thomas⁽¹⁾, M. Boracchia⁽¹⁾, A. Pasini⁽¹⁾, M.C. De Vito⁽¹⁾,
G. Osella Massa⁽¹⁾, L. Marrero⁽¹⁾, M. B. Albanessi⁽¹⁾, L. Delia⁽¹⁾

⁽¹⁾Instituto de Investigación en Informática LIDI (III-LIDI)
Facultad de Informática – UNLP

⁽²⁾ Comisión de Investigaciones Científicas de la Provincia de Buenos Aires (CIC)

{ppesado, hramon, pthomas, marcosb, apasini, cdevito, gosella, lmarrero, balbanesi,
ldelia}@lidi.info.unlp.edu.ar

1 CONTEXTO

Esta línea de Investigación forma parte del Proyecto “Sistemas de Software Distribuidos. Aplicaciones en procesos industriales, E-government y E-learning” del Instituto de Investigación en Informática LIDI acreditado por la UNLP y de proyectos específicos apoyados por CIC, Agencia y Telefónica.

2 RESUMEN

El objetivo de este subproyecto es realizar investigación y desarrollo en temas relacionados con los aspectos de Ingeniería de Software que se orientan al desarrollo e implementación de proyectos concretos de Sistemas Distribuidos, manejo de datos físicamente distribuidos y soluciones de hardware y software para Sistemas Distribuidos de Tiempo Real.

En este contexto se trabaja en aspectos de la Ingeniería de Software de diferentes sistemas con inteligencia distribuida (computadoras, robots, teléfonos móviles).

Palabras claves: *Sistemas Distribuidos – Ingeniería de Requerimientos – Planificación – Metodologías de Desarrollo – Sistemas Web – Bases de Datos Distribuidas – Sistemas Distribuidos de Tiempo Real.- Robótica*

3 INTRODUCCION

Un sistema distribuido consiste en un conjunto de computadoras autónomas conectadas por una red y con soporte de software distribuido. Permite que las computadoras coordinen sus actividades y compartan los recursos de hardware, software y datos, de manera tal que el usuario percibe una única facilidad de cómputo integrada aunque esta pueda estar implementada por varias máquinas en distintas ubicaciones. [1]

El desarrollo de sistemas distribuidos es una necesidad a partir de la utilización de redes de computadoras y de computadores personales de alta performance.

Algunas ventajas del procesamiento distribuido son:

- Mejora de la disponibilidad: la operación es factible en una configuración reducida cuando algunos nodos están temporalmente no disponibles.
- Configuración más flexible: una aplicación puede configurarse de distintas maneras, seleccionando el número apropiado de nodos para una instancia dada.
- Control y administración más localizada: un subsistema distribuido, ejecutando en su propio nodo, puede diseñarse para ser autónomo, de modo

que puede ejecutar en relativa independencia de otros subsistemas en otros nodos.

- Expansión incremental del sistema: si existe sobrecarga, el sistema puede expandirse agregando más nodos.
- Costo reducido: con frecuencia una solución distribuida es más barata que una centralizada.
- Balance de carga: en algunas aplicaciones la carga total del sistema puede ser compartida entre varios nodos.
- Manejo eficiente de datos distribuidos físicamente.
- Mejora en el tiempo de respuesta: los usuarios locales en nodos locales pueden obtener respuestas más rápidas a sus requerimientos.

Las características de los Sistemas Distribuidos conducen a la utilidad de desarrollar prácticas de Ingeniería de Software que apunten a los distintos aspectos del desarrollo de sistemas desde la captura de requerimientos y la planificación, pasando por las metodologías de desarrollo, hasta la verificación y simulación de procesos distribuidos, incluyendo el aseguramiento de calidad.

En particular un sistema distribuido de tiempo real debe interactuar con el mundo real, en puntos físicamente distantes y no necesariamente fijos, en períodos de tiempo que vienen determinados por el contexto o las restricciones de la especificación (en muchos casos a partir de una activación asincrónica).

Algunas de las dificultades principales del desarrollo de software para sistemas distribuidos de tiempo real son [Lev90]:

- Modelizar condiciones de concurrencia y paralelismo.
- Manejar las comunicaciones inter-procesos e inter-procesadores.
- Tratamiento de señales en tiempo real.

- Tratamiento de interrupciones y mensajes asincrónicos con diferente prioridad.
- Detectar y controlar condiciones de falla, a nivel de software, de procesadores y de comunicaciones. Prever diferentes grados de recuperación del sistema.
- Asegurar la confiabilidad de los datos y analizar su migración en condiciones de funcionamiento normal o de falla.
- Organizar y despachar la atención de procesos, manejando las restricciones de tiempo especificadas.
- Testear y poner a punto un sistema físicamente distribuido.

Todas estas dificultades conducen a la utilidad de desarrollar herramientas de Ingeniería de Software orientadas a STR y SDTR, de modo de trabajar en la modelización, especificación y verificación del software considerando las restricciones temporales.

La Ingeniería de Software comprende la aplicación de principios científicos para realizar la transformación ordenada de un problema en una solución elaborada de software, y el mantenimiento subsecuente de ese software hasta el final de su vida útil [2]. La utilización de estas prácticas para resolver sistemas distribuidos y de tiempo real hacen necesaria su adaptación en función de las características de dichos sistemas.

La adopción de un enfoque ingenieril para el desarrollo de software, genera una serie de fases o estados conformando un ciclo de vida. Este ciclo de vida esta guiado por una planificación que incluye el conjunto de acciones a realizar, y los productos generados por la aplicación del plan (inclusive el mismo plan) están administrados por diferentes Metodologías de Gestión y Desarrollo [3].

En el recorrido del ciclo de vida del desarrollo del software, la fase inicial comprende a la Ingeniería de Requerimientos que permite comprender,

documentar y acordar sobre el alcance del problema, teniendo esto impacto directo sobre la Planificación y la Gestión del Proyecto de acuerdo a la Metodología de desarrollo seleccionada para el mismo [4]. Este no es el único impacto que justifica el énfasis en la Ingeniería de Requerimientos, ya que esta disciplina permite establecer claramente *que* se debe hacer posponiendo el *cómo* a etapas posteriores en el ciclo de vida, constituyendo el fundamento de la construcción de un Sistema de Software [5].

El modelo de procesos elegido para el desarrollo de software define las actividades a realizar para la generación de productos de acuerdo a los objetivos planteados [6].

Asociado con la Ingeniería de Software de Sistemas Distribuidos, está el problema de utilizar un entorno WEB para los servicios que ofrece el Sistema. La tendencia creciente al desarrollo de arquitecturas centradas en un servidor (o un conjunto de servidores distribuidos) que ofrecen una interfaz WEB a los usuarios ha generado un importante desarrollo de la Investigación en metodologías y herramientas orientadas a Sistemas WEB, así como ha obligado a establecer nuevas métricas y parámetros de aseguramiento de la Calidad para tales Sistemas. [7] [8] [9]

Por último, el modelo distribuido de datos hace posible la integración de BD heterogéneas proveyendo una independencia global del administrador de bases de datos respecto del esquema conceptual. [10]

4 LINEAS DE INVESTIGACION Y DESARROLLO

- Conceptos de procesamiento distribuido. Arquitectura, comunicaciones y software.
- Metodologías de especificación, validación y desarrollo de SSD y SSDTR.
- Metodologías ágiles de desarrollo.

- Ingeniería de Requerimientos, en particular de sistemas distribuidos.
- Planificación de tareas para desarrollo de sistemas distribuidos.
- Bases de Datos Distribuidas [11]
- Herramientas de integración y mantenimiento de proyectos distribuidos.
- Lenguajes y ambientes para procesamiento distribuido.
- Reingeniería de sistemas complejos que migran por down-sizing a esquemas cliente-servidor distribuidos.
- Sistemas de Tiempo Real. Sistemas Distribuidos de Tiempo Real.
- Tratamiento de señales en tiempo real.
- Sistemas de control de robots en tiempo real. Algoritmos colaborativos entre máquinas móviles.
- Sistemas con identificación segura en tiempo real.

5 RESULTADOS ESPERADOS/OBTENIDOS

- Avanzar en la capacitación continua de los miembros de la línea de investigación.
- Desarrollar soluciones a problemas concretos de software de sistemas distribuidos, poniendo énfasis en el desarrollo de metodologías y herramientas específicas para clases de aplicaciones. A modo de ejemplo se cita la herramienta desarrollada en el III-LIDI que permite generación de código PHP interactuando con diferentes motores de BDD [12] [13]. Actualmente se estudian herramientas que permiten tomar un modelo UML de un sistema y derivar código a partir de él o realizar una comprobación de consistencia lógica de dicho modelo.
- Investigar sobre herramientas para la planificación y el seguimiento de proyectos distribuidos. En el III-LIDI se ha utilizado una herramienta WEB para seguimiento de proyectos en forma colaborativa.
- Verificar Sistemas Distribuidos a partir del método de comprobación de

modelos. Se ha realizado una especificación del protocolo de comunicación Gnutella para redes Peer-to-Peer descentralizadas [14]. Como resultado, se han presentado dos modelos: uno completo, apropiado para la simulación del comportamiento de un nodo Gnutella, y otro acotado, que permite comprobar si determinadas propiedades expresadas en LTL (Lógica Temporal Lineal) son ciertas. En este contexto se estudió la herramienta SPIN [15], [16].

- Investigar y realizar desarrollos específicos sobre Sistemas Distribuidos de Tiempo Real, en particular Aplicaciones en control de robots. El objetivo general es el estudio orientado hacia la Ingeniería de Software de Sistemas Distribuidos, en particular, aquellos en los que los procesos se ejecutan en máquinas autónomas tales como los robots, con restricciones de tiempo real. Los temas involucrados se enfocan en particular a entornos de simulación para conjuntos de robots que trabajen cooperativamente. Actualmente se estudia la comunicación de sistemas autónomos heterogéneos vía Internet para trabajo cooperativo. El aporte principal se orienta al proyecto PAV 076 "Sistemas Inteligentes de apoyo a los Procesos Productivos" (Red formada por UNLP, UNSJ, UNS, UNSL, UTN-FRC, UNC, UCC, subsidiado por la Agencia Nacional de Promoción Científica y Tecnológica), donde interesa generar un entorno de software para el desarrollo de aplicaciones industriales multi-robot.
- Desarrollar soluciones para problemas de E-government. El III-LIDI participa en el Proyecto "Servicios de E-government con verificación de Identidad" subsidiado por CIC-Telefónica. Uno de los aspectos centrales para la calidad y valor agregado de los servicios ofrecidos desde el gobierno (para empresas o

individuos) es la verificación de la identidad. Tratándose de servicios vía internet a través de comunicaciones fijas y móviles, los componentes tecnológicos (hardware y software), manteniendo tiempos de respuestas razonables para procesos en tiempo real, resultan críticos y requieren investigación y desarrollo. Actualmente se está desarrollando un prototipo para el acceso de documentos mediante autenticación de identidad con scanners de huellas digitales.

Algunas transferencias de resultados logradas:

- **Auditoría de la Red Única Provincial de Comunicación de Datos del Gobierno de la Provincia de Buenos Aires.**

En el marco de un acuerdo entre la RedPIBA y la CICIPBA se han auditado 135 Municipios (1305 nodos) planificados, incluyendo la verificación de las instalaciones y la puesta en servicio (abarca el control de ancho de banda en tiempo real) de la red según el manual de procedimiento elaborado.

Actualmente se analiza la definición del manual de procedimiento para la mesa de ayuda que dará soporte inicialmente a 1000 escuelas conectadas a la Red Provincial.

- **Análisis, Diseño e Implementación de un Sistema de Información para la gestión estratégica de RRHH del Ministerio de Seguridad de la Provincia de Bs. As.**

Se encuentra en la etapa de mantenimiento el Sistema de Información desarrollado por el III-LIDI. Este Sistema es apto para el tratamiento de datos de todos los regímenes estatutarios sirviendo de apoyo a la gestión de unidades organizativas centralizadas y descentralizadas del Sistema de

Seguridad Pública (Policía Buenos Aires 2, Comunal, de Distrito, Siniestral, de Custodia de Objetivos Fijos y Traslado de Detenidos, de Investigaciones del Tráfico de Drogas ilícitas, Central de Atención Telefónica de Emergencias y Cuerpo de Baqueanos).

Actualmente se analiza la ampliación del sistema para la eficiente administración del personal del Ministerio con estado no policial de acuerdo a la normativa impuesta por la Ley 10430.

▪ **Software de Tiempo Real para Entrenamiento Deportivo**

El III-LIDI participa de un proyecto de desarrollo de software para entrenamiento deportivo con procesamiento en tiempo real [17] [18] [19] de señales recogidas a partir de ejercicios realizados por los jugadores.

Entre los ejercicios implementados se pueden mencionar:

Perimetral: El objetivo del ejercicio es completar un recorrido perimetral con pelota dominada en el menor tiempo posible. Sabiendo la posición de los jugadores en el recorrido y las marcas no registradas.

Tiro de Penales: El objetivo del ejercicio es asignar puntaje a tiro de penales de acuerdo a la precisión censada del impacto del mismo.

• **Ministerio de Desarrollo Humano**

EL III-LIDI participa en un proyecto de diagnóstico de la situación de los sistemas de software utilizados para la administración de los planes sociales de la Pcia. de Bs. As.

Se incluye la evaluación de los procedimientos implementados por la Dirección de Informática así como la

gestión de los planes a cargo de la Dirección de Políticas Alimentarias.

5. FORMACION DE RECURSOS HUMANOS

Los integrantes de esta línea de investigación dirigen Tesinas de Grado en la Facultad de Informática, 1 Tesis de Magister, 1 Tesis Doctoral, 1 Becario de la Agencia y Becarios III-LIDI en temas relacionados con el proyecto y participan en el dictado de asignaturas/cursos de postgrado de la Facultad de Informática de la UNLP .

Colaboran en este proyecto los pasantes alumnos Nicolás Luna y Danae López.

6. BIBLIOGRAFIA

[1] G. Coulouris. Distributed Systems – Concepts and Design. Addison-Wesley. 1994.

[2] R. Pressman. Ingeniería de Software: Un Enfoque Práctico. McGraw-Hill. 2002

[3] R. Wysocki. Effective Project Management: Traditional, Adaptive, Extreme, .Wiley .2003

[4] Loucopoulos, P; Karakosas, V.. Systems Requirements Engineering. .McGraw Hill. Book Company. 1995

[5] G. Kotonya and I. Sommerville,. Requirements Engineering: Processes and Techniques, Wiley. 1998

[6] Pleeger. Ingeniería de Software: Teoría y Práctica. Prentice-Hall. 2002

[7] Stephen Kan. .Metrics and Models in Software Quality Engineering (2nd Edition). Addison Wesley. 2003

[8] Offutt J., “Quality Attributes of Web Software Applications”. IEEE Software: Special, Issue on Software Engineering of Internet Software 19 (2):25-32, Marzo/Abril 2002.

[9] Wu, Y. y Offutt, J. “Modeling and testing web-based Applications”. <https://citeseer.ist.psu.edu/551504.html>: 1-12, Julio 2004

- [10] Silberschatz A et al: “Fundamentos de Bases de Datos”, Tercera Edicion Mc Graw Hill 1998
- [11] Ozsu M. Valduriez, P. : “Principles of Distributed Database Systems”, Segunda Edicion. Prentice Hall 1999
- [12] Mello S. J., “Executable UML”, Addison-Wesley, 2002.
- [13] S. Ceri, P.Fraternalli,A. Bongio, “Web Modeling Language (WebML): a modeling language for designing web sites”. Computer Networks, vol 33, 2000.
- [14] De Vito, M.C., Osella Massa, G.L. “Model Checking: Un modelo del protocolo Gnutella. Simulación y Verificación usando Spin”. XII Congreso Argentino de Ciencias de la Computación. CACIC 2006. Universidad Nacional de San Luis. Octubre de 2006.
- [15] Holzmann, G.J., “The Spin Model Checker: Primer and Reference Manual”. Addison-Wesley, 2003
- [16] Holzmann, G.J. “The Model Checker Spin”, IEEE Trans. on Software Engineering, Vol. 23, No. 5, pp. 279-295, 1997.
- [17] Berryman, S.J.; Sommerville, I.; Modelling real-time constraints. Software Engineering for Real Time Systems, 1991 pp 164 - 169 .
- [18] Burns & Wellings. Real-Time Systems and Programming Languages. 1996.
- [19] Goldsmith.A Practical Guide to Real-Time Systems Development. 1993