

Sistemas de Software Distribuido

**P. Pesado^(1,2), H. Ramón⁽¹⁾, P. Thomas⁽¹⁾, M. Boracchia⁽¹⁾, E. Ibañez⁽¹⁾, L. Marrero⁽¹⁾,
M. B. Albanessi⁽¹⁾, L. Delia⁽¹⁾, C. De Vito⁽¹⁾, N. Restelli⁽¹⁾, N. Luna⁽¹⁾, G. Caseres⁽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, eibanez, lmarrero, balbanesi, ldelia, cdevito, nrestelli,
nluna, gcaseres}@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:

- 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]

Toda aplicación esta dividida, al menos, en dos subsistemas, front-end y back-end. El front-end es una aplicación que interactúa en forma transaccional con los usuarios (clientes, empleados, proveedores, etc.) de una Organización y requiere del apoyo del subsistema back-end que posee características analíticas y de workflow. Un ECA-rules (Event-Condition-Action) representan la tecnología para BD activas y el método natural para soportar dicha funcionalidad. ECA rules puede ser utilizado para actividades de chequeo automático de restricciones, o mensajes. Una cuestión importante, asociada a un ECA es cómo predecir el comportamiento en tiempo de ejecución. ECA rules posee una sintaxis declarativa de alto nivel, que le permite llevar a cabo acciones automáticas en respuesta a condiciones previamente definidas. Un ECA-rules sobre un subsistema Back-End permite controlar de manera transparente para el usuario un conjunto de reglas que hacen a su negocio. Estos controles se realizan directamente en el servidor de la aplicación y, ante una acción, disparan los mensajes correspondientes sobre el usuario en su aplicación front-end. [10] [11]

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. [12].

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 utilizando frameworks propios con diferentes tecnologías.
- Ingeniería de Requerimientos, en particular de sistemas distribuidos.
- Planificación de tareas para desarrollo de sistemas distribuidos.
- Bases de Datos Distribuidas [13]
- 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.
- ECA-rules sobre un subsistema Back-End permite controlar de manera transparente para el usuario un conjunto de reglas que definen su comportamiento [14]

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.
- Framework NET4DB desarrollada en el III-LIDI que permite generación automática de código .NET, interactuando con diferentes motores de BDD [14] [15]. Actualmente se estudian herramientas que permiten tomar un modelo UML de un sistema y derivar código a partir de él o realizar una verificación de consistencia lógica de dicho modelo.
- El motor simpleECA para resolver el back-end de PHP4DB [16] y NET4DB.

6 TRANSFERENCIAS DE RESULTADOS

- **Ministerio de Desarrollo Humano de la Provincia de Buenos Aires. Sistema de Asistencia Social 2 a 5 años:** tiene como objetivo realizar una administración y seguimiento de beneficios otorgados a Instituciones para la protección de niños de 2 a 5 años. Se gestionan las liquidaciones mensuales, por cupos y/o asistencia media, a pagar con interdepósito bancario, con las posteriores rendiciones de pagos realizados. La implementación fue realizada con NET4DB.
- **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 CICPBA (inicialmente para controlar la instalación de la red de datos provincial) se plantea la necesidad de auditar en forma continua aspectos relacionados al mantenimiento de la infraestructura, controlar la sobresubscripción, incorporar nuevos nodos a la red en forma controlada, obtener información del funcionamiento y estado de la red desde el punto de vista del usuario final. Por lo cual se debe definir un proceso de características distribuidas (y el software de soporte) para soportar estas necesidades. Este proceso estará implementado en PHP4DB desarrollado en III-LIDI [16].
- **Sistema web desarrollado para la Unidad de Relaciones Internacionales Universitarias de la UNLP:** El III-LIDI ha desarrollado un sistema cuyo objetivo es mantener información relacionada a la participación de los miembros de la UNLP en actividades con el exterior, y así fomentar el conocimiento, la colaboración y la responsabilidad de informar y registrar dichas participaciones. El sistema se divide en dos subsistemas: *Gestión de convenios*; administra la información relacionada con los convenios entre la Universidad (o alguno de sus miembros) e instituciones del exterior. *Gestión de características de las facultades*; administrará información relacionada con las fortalezas y debilidades de las diferentes unidades académicas de la UNLP, además de las líneas de investigación/trabajo y la participación en diferentes redes internacionales. Actualmente el sistema se encuentra en la etapa de Testing. Este desarrollo esta basado en PHP4DB desarrollado en III-LIDI [16].
- **Sistema de Congresos Virtuales:** El III-LIDI está desarrollando un sistema para la organización y transmisión de congresos vía Internet. El mismo podrá transmitir desde una ubicación y los asistentes podrán participar desde cualquier máquina conectada a Internet. Se podrá transmitir video, audio, texto y formato tipo slide (ppt) Durante la transmisión, los asistentes podrán realizar preguntas al expositor (pregunta por demanda, hand rising) o votar por alguna pregunta formulada, como si se encontraran en una sala de presentaciones convencional. Además, el expositor podrá lanzar encuestas sencillas (de respuesta Sí o No) entre los asistentes, para saber su opinión sobre los diferentes aspectos de su exposición. La transmisión de cada congreso podrá ser grabada y almacenada en el sistema para luego ser retransmitida. Este tipo de transmisiones podrá ser programada con anticipación para que arranque en forma

automática en un determinado día y horario. Actualmente el sistema se encuentra en la etapa desarrollo

7. FORMACION DE RECURSOS HUMANOS

Los integrantes de esta línea de investigación dirigen Tesinas de Grado en la Facultad de Informática, 1 Tesis Doctoral, 1 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 .

8. 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] Automatic Control of Workflow Processes Using ECA Rules, Joonsoo Bae, Hyerim Bae, Suk-Ho Kang, Yeongho Kim, IEEE Transactions on Knowledge and Data Engineering , Volume 16 , Issue 8 (August 2004), 2004, ISSN:1041-4347
- [11] An Event-Condition-Action Language for XML, J. Bailey, A. Poulouvasilis, P. Word, WWW2002, May 7-11, 2001, Honolulu, Hawaii, USA. ACM 1-58113-449-5/02/0005.
- [12] Silberschatz A et all: “Fundamentos de Bases de Datos”, Tercera Edicion Mc Graw Hill 1998
- [12] Ozsu M. Valduriez, P. : “Principles of Distributed Database Systems”, Segunda Edicion. Prentice Hall 1999
- [13] Mello S. J., “Executable UML”, Addison-Wesley, 2002.
- [14] S. Ceri, P. Fraternali, A. Bongio, “Web Modeling Language (WebML): a modeling language for designing web sites”. Computer Networks, vol 33, 2000.
- [15] Tesis Stefan Böttger, Distributed Composite Event Monitoring. June, 2006 Ruprecht-Karls-Universität Heidelberg (Kirchhoff-Institut für Physik) – Universität Leipzig (Fakultät für Mathematik und Informatik – Institut für Informatik).
- [16] Delia, Caseres, Ramón, Thomas, Bertone; Framework para el Desarrollo Ágil de Aplicaciones Web, CACIC 2006, San Luis, 17 al 21 de octubre de 2006.