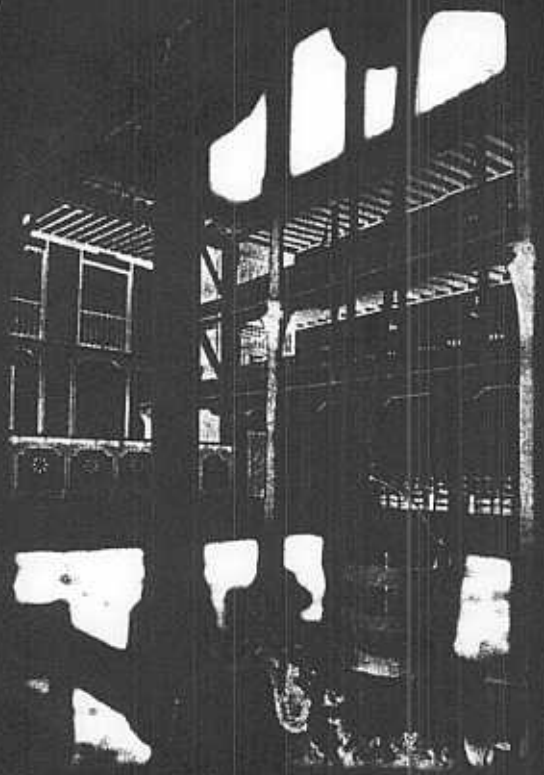


VI jornadas

Ingeniería del Software y Bases de Datos

Almagro (Ciudad Real)
21 - 23 de Noviembre de 2001

Grupo Alarcos
Departamento de Informática
Escuela Superior de Informática

Editores:
Óscar Díaz
Arantza Illarramendi
Mario Piattini

JISBD' 2001

Jornadas de Ingeniería del Software y Bases de Datos

21 y 23 de Noviembre de 2001
Almagro (Ciudad Real)

Grupo Alarcos

Departamento de Informática

Escuela Superior de Informática

Editores

Oscar Díaz
Arantza Illaramendi
Mario Piattini

Editores

Oscar Díaz
Arantza Illaramendi
Mario Piattini

Organizadas por:

Grupo Alarcos

Departamento de Informática

Escuela Superior de Informática

Primera edición, 2001
I.S.B.N.: 84-699-6275-2
Depósito Legal: CR-506/2001
Imprime: GRAFICAS GARRIDO, S.L.

José Francisco Alc
Idoia Alarcón
João Araújo
Manuel Barrena
Pere Botella
Hilario Canos
Matilde Celma
João A. Correia L
Dolors Costal
Carmen Costilla
Juan Manuel Cue
Pablo de la Fuente
Carlos Delgado K
Xavier Franch
Juan Garbajosa
Jesus García Moli
Alfredo Goñi
Juan Hernández
Arturo Jaime
Natalia Juristo
Esperanza Marco
José Manuel Mar
Eduardo Mena
Roberto Moriyón
Juan José Moren
José Parets
Oscar Pastor
Ernesto Pimentel
António Rito Silv
Nieves R. Brisab
Félix Saltor
José Samos
Ernest Teniente
Ambrosio Toval
Jose María Troya
Antonio Vallecil
Juan Carlos Yeln

Prólogo del Comité de Programa

Este volumen recoge los trabajos presentados en las VI Jornadas de Ingeniería de Software y Bases de Datos celebrados en Almagro los días 21, 22 y 23 de noviembre de 2001. Estas edición ha buscado cohesionar más si cabe, a las comunidades de Bases de Datos e Ingeniería de Software, estableciendo un único comité de programa, y esforzándose por buscar talleres y conferenciantes de interés común. Todo ello con el objetivo de fomentar la creciente necesidad de equipos multi-disciplinares que puedan afrontar los cada vez más exigentes desafíos de la sociedad de la información.

Para participar en las sesiones técnicas se recibieron un número record de trabajos: 89. Este número avala, por un lado, la creciente madurez del área, y por otro, el prestigio de las Jornadas, fruto del buen hacer de las ediciones anteriores. Finalmente, se aceptaron 38 trabajos.

A continuación facilitamos algunos datos sobre la evaluación, ya que velar por la calidad del proceso de revisión es nuestra labor principal como presidentes del comité de programa. La práctica totalidad de los trabajos fueron revisados por tres miembros del comité de programa. El evaluador tenía que definirse como "experto", "conocedor" o "poco familiarizado" con la temática del trabajo, siendo cuatro los posibles resultados: "rechazar", "posiblemente rechazar", "posiblemente aceptar" y "aceptar". El 82% de los trabajos contó con un evaluador "experto". De los trabajos aceptados, ninguno de sus revisores "expertos" decidió rechazarlo, el 37% contó con un "experto" que concluyó con "aceptar", y en el 87% de los casos NO hubo disparidad de opiniones, es decir los tres revisores concluyeron con un "aceptar" o un "posiblemente aceptar". De los trabajos rechazados, el 58% tuvo un evaluador "experto" que concluyó con un "rechazar".

Pero además de las sesiones técnicas, esta edición ha apostado por los talleres y tutoriales. Los talleres promovidos por los comités respectivos, favorecen un intercambio amplio y ágil, tanto de ideas ya asentadas, como de barruntos o intuiciones que en un entorno más formal difícilmente saldrían a la luz. Con esta finalidad, se organizaron seis talleres, a saber: Impacto de la tecnología XML en las bases de datos, Ingeniería del software basada en componentes distribuidos, Almacenes de datos y tecnología OLAP, Ingeniería del Software orientada a la Web, Desarrollo de software preciso y Evolución del software.

Contamos también con una mesa redonda y tres conferenciantes invitados. La primera trato el tema de *"La investigación en Ingeniería del Software"* y fue coordinada el Dr. Pere Botella. Respecto a los conferenciantes, se contó con la presencia de F. Arbab que habló sobre Panta Rei, con P. Fankhaus que trató sobre XQuery y una tercera conferencia que, en el momento de escribir este prólogo aún quedaba por determinar.

Ya para finalizar con el programa, esta edición ha introducido dos novedades: la sesión dedicada a trabajos ya publicados, y los tutoriales. La primera nace con el deseo de presentar en las Jornadas aquellos trabajos nacionales que, habiendo sido ya publicados en revistas de prestigio, puedan servir de referencia a otros grupos por su

Índice

SEGESOFT: Entorno de Entrenamiento para la Gestión de Proyectos Software

J. C. Riquelme[1], I. Ramos[1], J. Aguilar-Ruiz[1], F. Ferrer[1], M. Toro[1], J.J. Dolado[2], A. Ruiz de Infante[2], J. Tuya[3], P. Fernández[3], M.A. Prieto[3], M. Ruiz-Carreira[4], D. Rodriguez-García[5], M. Satpathy[5], R. Harrison[5], R. Matilla[6], M.A. Álvarez[6]

[1]Universidad de Sevilla
riquelme@lsi.us.es
[2]Universidad del País Vasco
dolado@si.ehu.es
[3]Universidad de Oviedo
tuya@lsi.uniovi.es
[4]Universidad de Cádiz,
[5]University of Reading,
[6]THALES Information system

Abstract. La gestión de proyectos se puede considerar todavía como un arte en el cual el uso de la información cuantitativa tiende a fomentar un enfoque más riguroso de la gestión. En este trabajo presentamos la estructura y los elementos principales de un entorno de entrenamiento para directores de proyectos. El objetivo del sistema es proporcionar una estructura uniforme para que se puedan incorporar nuevas técnicas en la estructura de forma gradual. El sistema reúne y almacena los datos del proyecto real y del simulado e implementa diferentes técnicas basadas en aprendizaje automático, modelado dinámico, monitorización de proyectos, etc. El propósito básico de este trabajo es el de presentar un entorno que facilite la toma de decisiones integrando diferentes técnicas y líneas de investigación.

Palabras claves: gestión de proyectos software, simulación, minería de datos

Un Entorno de Entrenamiento

La gestión de proyectos es una de las actividades de la ingeniería del software que aún necesita de unas bases técnicas sólidas. Cada paso que se da en las áreas de estimación, seguimiento, interpretación de los datos, etc., acerca un proyecto al objetivo de calidad y cumplimiento de tiempo y presupuesto estimado. Los sistemas de entrenamiento que presentan diferentes escenarios a los directores de proyectos permiten superar la ausencia de datos relacionados con la gestión de proyectos.

En los actuales entornos de gestión, el director tiene que tomar decisiones en función de una descripción aproximada del entorno del proyecto. El hecho de que la mayoría de la información de los proyectos finalizados sea desconocida o contenga un elevado grado de incertidumbre hace difícil averiguar los parámetros del proyecto. Es

JISBD 2001
I.S.B.N.: 84-699-6275-2

más, los datos disponibles provienen de múltiples fuentes y la integración de todos esos datos no es una tarea sencilla de realizar con las herramientas software actuales.

Con estas ideas en mente, hemos desarrollado los principales módulos del sistema SEGESOFT para poder apoyar el entrenamiento de la gestión del software. El sistema implementa diferentes técnicas para manejar diversos usos de los datos disponibles en la herramienta de gestión de proyectos.

2 Integración de Técnicas para la Gestión del Software

Nuestra herramienta integra diversos módulos que corresponden a diferentes técnicas utilizadas en la gestión de proyectos. En las siguientes subsecciones presentamos brevemente algunas de las técnicas implementadas y la estructura general del sistema.

2.1 La Simulación y los Modelos Dinámicos

Los modelos dinámicos y la aparición de entornos de simulación potentes y amigables (Stella, Vensim, iThink, Powersim, etc.) aplicados a la gestión de proyectos software al comienzo de los años 90 permitieron la creación de herramientas (llamadas normalmente simuladores de proyectos), que posibilitan la simulación del comportamiento de dichos proyectos. Con estas herramientas de simulación, los responsables de proyectos software pueden "experimentar" sin ningún tipo de coste el efecto que tiene sobre el proyecto la aplicación o no de diferentes políticas de gestión [1][2].

Un modelo dinámico para proyectos software permite conocer la evolución del proyecto. Pero es importante resaltar que dicha evolución y, por tanto, la consecución de los objetivos del proyecto va a depender de: a) las estimaciones iniciales de los recursos necesarios, b) las políticas de gestión que se apliquen, c) las características del proyecto y d) las características de la empresa de desarrollo.

Un modelo dinámico para proyectos software incorpora una serie de parámetros que permiten definir los aspectos anteriores, es decir, estimaciones iniciales, políticas de gestión y características del proyecto y de la organización. Un simulador de proyectos software permite realizar diferentes análisis del proyecto dependiendo del estado en el que se encuentra:

1. Un análisis a priori del proyecto antes de comenzar su ejecución.
2. Una monitorización del proyecto durante la ejecución del mismo para ir adaptando las estimaciones iniciales a la evolución del proyecto.
3. Un análisis post-mortem del proyecto una vez finalizado para conocer cómo podrían haberse mejorado los resultados del mismo.

En definitiva, un simulador de proyectos software permite responder a las siguientes cuestiones: "¿qué ocurrirá si..?" antes de comenzar, "¿qué está ocurriendo..?" durante la ejecución y "¿qué habría ocurrido si..?" una vez que el proyecto ha finalizado.

2.2 Descubriendo Conocimiento o Aprendizaje Automático

Las técnicas y herramientas computacionales diseñadas para sustentar la extracción de conocimiento útil a partir de bases de datos se conocen tradicionalmente como aprendizaje automático (machine learning en inglés). Recientemente, términos como minería de datos (data mining) o descubrimiento de conocimiento en bases de datos (knowledge discovery in databases o KDD) se utilizan frecuentemente en vez de aprendizaje automático. En general, las técnicas así denominadas tratan de extraer automáticamente una información útil para la toma de decisiones o la exploración del origen de los datos [3].

Un proceso estándar de aprendizaje automático está compuesto por varios pasos tales como preparación de los datos, selección, limpieza, minería propiamente dicha e interpretación de los resultados. Por tanto, la minería de datos se puede considerar como un paso dentro de un proceso más global; este paso consistiría en aplicar algoritmos específicos para extraer patrones a partir de los datos.

2.3 Otras Técnicas

La medición es esencial para gestionar, evaluar, predecir y mejorar la calidad de los procesos y productos software. Las actuales herramientas de gestión de proyectos trabajan con tiempo y recursos pero no consideran los aspectos de calidad. La herramienta SEGESOFT proporciona facilidades para el control de calidad.

Para controlar la calidad de los productos y procesos es necesario definir modelos de calidad para ambos. Además es necesario definir diferentes modelos para los diversos componentes generados durante el ciclo de vida del software. Algunos de los modelos siguientes son necesarios para analizar el estado del proyecto.

Entre los modelos disponibles para la evaluación de los procesos, los más influyentes son el *Capability Maturity Model* [4] (CMM), ISO/IEC 12207, ISO 9000, el modelo BOOTSTRAP y el ISO 15504 [5] (SPICE). Los modelos para evaluar la calidad de los productos varían desde los modelos jerárquicos fijos como Boehm, FCM (Factor Criteria Metric) de McCall e ISO 9126 (actualmente bajo revisión) a enfoques más flexibles como QMS (Quality Management Subsystem) de Kitchenham. Todos estos modelos ayudan a clarificar qué aspectos de calidad son considerados y porqué.

Aunque los modelos de proceso ayudan a incrementar la calidad de los productos que producen, la relación entre modelos de procesos y calidad de productos no está clara en los modelos mencionados anteriormente y solamente la metodologia PROFES [6] aborda parcialmente el problema. Para aliviar este problema, Satpathy et al. [7] definen un modelo genérico como una plantilla que puede ser instanciada para ser el modelo de calidad de cualquier proceso individual. En el entorno SEGESOFT se introduce un módulo que permite definir modelos de proceso y modelos de calidad. Posteriormente se pueden evaluar los procesos de acuerdo a las mediciones y datos obtenidos y al modelo de calidad especificado

3 Estructura del Sistema

3.1 Arquitectura General

La arquitectura de la versión actual del sistema está formada por un conjunto de componentes desarrollados sobre una base de datos para el almacenamiento de diferentes tipos de métricas. La importancia de disponer de un módulo que permita registrar todo tipo de métricas ha sido reseñada en los trabajos de Paul et al. [8], donde se propone una estructura general para una base de datos de proyectos con una estructura relacional de la información, aunque no se presenta ninguna herramienta que materialice dichas ideas.

En este artículo se intenta dar un paso más allá integrando sobre una misma base de datos de métricas diferentes herramientas analíticas y proporcionando una estructura abierta que permita incorporar futuras necesidades. Desde el punto de vista de la implementación se ha tratado de reutilizar en lo posible software desarrollado o en desarrollo, licenciado como "freeware" en numerosas ocasiones. Se ha utilizado Java 2 SDK (versión 1.2 ó 1.3) para el desarrollo.

La estructura general del sistema se representa en la Fig. 1 donde se incluyen los principales módulos del sistema, los cuales son suficientemente coherentes e independientes entre sí. En dicha figura las flechas representan una relación de uso de la información. A continuación se describen brevemente la función de cada módulo:

- El módulo de *Guiones de Entrenamiento* organiza las lecciones, datos y guías para el uso de la herramienta.
- El módulo de *Medición* permite recoger la información relativa a las métricas de los proyectos.
- El módulo de *Estimación* incluirá algunos métodos típicos de estimación de proyectos.
- El módulo de *Seguimiento* está específicamente diseñado para monitorizar la evolución del proyecto. Podría considerarse como parte del módulo de medición, pero se diferencia por su interés en resaltar información relativa al estado de un proyecto en curso durante y al final del mismo.
- El módulo de *Simulación* permite al usuario modelar la organización utilizando un enfoque de modelos dinámicos para obtener datos simulados.
- El módulo de *Aprendizaje Automático* implementa algunos algoritmos utilizados para resumir grandes volúmenes de datos.
- La *Base de Datos de Métricas* que es el corazón del sistema donde se almacena toda la información.
- El módulo de *Visualización*, está siendo desarrollado con el objetivo de permitir la presentación de información de interés para el gestor en forma de gráficos y diagramas
- El módulo de *Evaluación de Calidad* implementa criterios para la evaluación continua de la calidad de un proyecto.

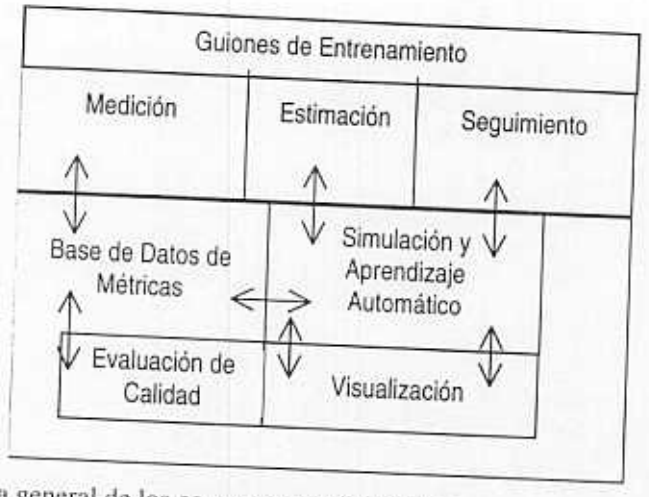


Fig. 1. Estructura general de los componentes del sistema

3.2 Base de Datos

El modelo de la base de datos permite almacenar diferentes proyectos ejecutados por una organización. La figura 2 muestra de forma esquemática e informal la estructura de la información. La raíz presenta la organización en su globalidad, que se estructura por una parte en el conjunto de proyectos (diferenciando entre activos y pasados) y en la línea base de proyectos (que guarda las mediciones globales para proyectos finalizados) desglosadas por tipos de proyectos. Cada uno de los proyectos tendrá diferentes versiones correspondientes cada una a las diferentes planificaciones que se han realizado a lo largo del tiempo.

Cada una de las versiones del proyecto continúa representándose de forma jerárquica. Por ejemplo, en el caso de un proyecto activo, este se desglosa en diferentes versiones para reflejar las replanificaciones que se realizan durante el transcurso de la ejecución del proyecto. Dependiendo de éste se encuentran las estructuras jerárquicas básicas siguientes:

- RBS (Resource Breakdown Structure): Recursos disponibles para el proyecto.
- WBS (Work Breakdown Structure): Tareas del proyecto (desglosadas de forma jerárquica, a partir de las cuales se construye el diagrama Gantt de planificación).
- PBS (Product Breakdown Structure): Productos generados a lo largo del proyecto.
- PRBS (Process Breakdown Structure): Procesos del proyecto (tipos de actividades del proyecto).
- MBS (Metrics Breakdown Structure): Información de métricas de diferentes tipos.
- SBS (Simulation Breakdown Structure): Parámetros y estructuras utilizadas para la simulación del comportamiento del proyecto.
- MLBS (Machine Learning Breakdown Structure): Parámetros de los algoritmos de aprendizaje automático utilizados.

Estas estructuras básicas jerárquicas se van desglosando en otros nodos que representan mayor nivel de detalle. Existe otra estructura que relaciona nodos correspondientes a la jerarquía. Como ejemplo, en la figura anterior se pueden ver las

predecesoras de una tarea, los recursos planificados en una tarea, o el progreso y esfuerzo realizado en la ejecución de una tarea.

La información de los proyectos que se muestra en la figura 2 se ha transformado posteriormente en un diagrama de clases especificándolo como una estructura entidad-relación.

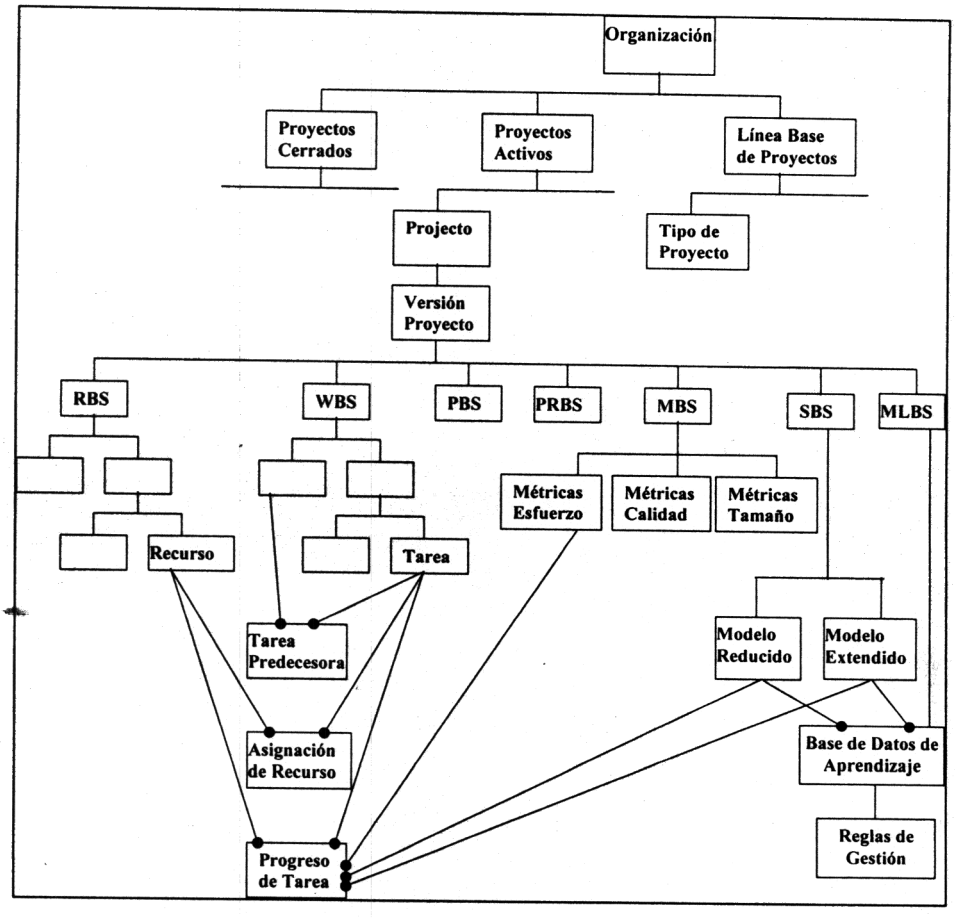


Fig. 2. Esquema de la Estructura de la Base de Datos

3.3 Interfaz de Usuario

En relación con la interfaz de usuario, se pueden apuntar una serie de líneas generales que homogeneizan el uso y navegación del usuario a través de las estructuras anteriores. En la figura 3 se representa el aspecto de una pantalla.

En la parte izquierda se dispone de un navegador que permite localizar rápidamente cada uno de los nodos de las estructuras anteriores y las opciones básicas de crear, borrar, mover y renombrar los diferentes elementos. En el panel de información se representará la información de cada objeto. En general, los tipos de información a mostrar son los siguientes:

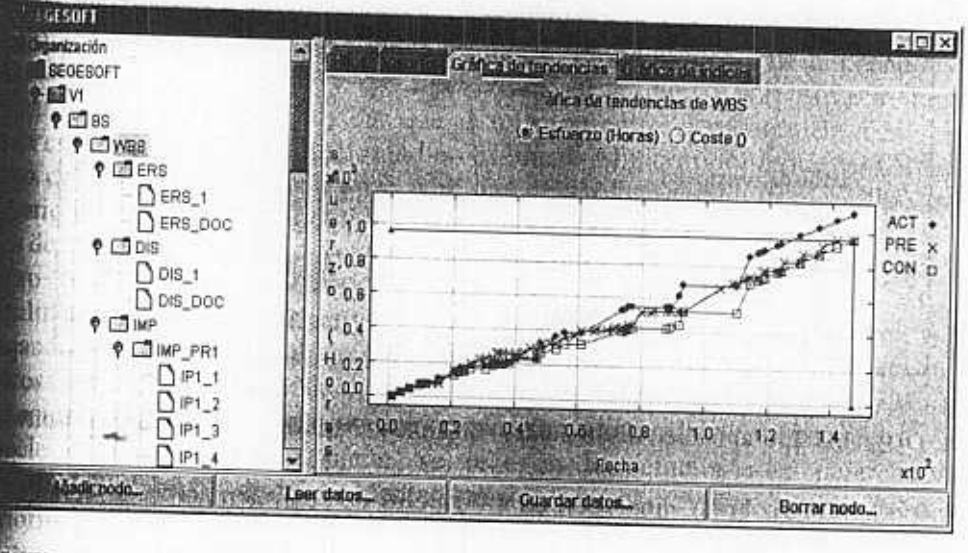
- Datos correspondientes al objeto (sus atributos).

ciones (una o más pestañas para permitir modificar los datos de las relaciones te objeto).

uctura en forma de matriz (correspondientes al objeto y sus descendientes) de a que se puedan realizar actualizaciones y visualización rápida. Por ejemplo, na WBS se mostraría una matriz todos los descendientes en filas y en mnas los atributos más relevantes.

uctura en forma de gráfico (correspondiente al objeto y sus descendientes). Por plo, mostrar gráficamente el PERT, GANTT o WBS.

os datos en forma gráfica (por ejemplo, curvas de progreso).



3. Ejemplo del interfaz gráfico

odos los objetos (navegador y cada uno de los paneles) están diseñados de forma se puedan reutilizar en diferentes contextos. Por ejemplo, en la figura 3 se ha ccionado un proyecto (SEGESOFT) al cual se accede en su versión primera (V1) estructura de partición del trabajo (WBS) de la cual dependen las tareas. A la cha se han seleccionado las pestañas correspondientes a las curvas de valor mulado que muestran la evolución temporal de esfuerzo o coste (actual o real, upuestado y conseguido). Si se seleccionase una tarea (por ejemplo el diseño: S) se mostrarían los mismos datos relacionados solamente con dicha tarea a lquier nivel de detalle.

4 Modelado y Simulación

objetivo de este módulo es el de recoger el funcionamiento y los principales mponentes de entornos de simulación tan conocidos como Vensim, Stella, owersim, etc...

te módulo implementa un modelo dinámico simplicado que se puede utilizar para mular el comportamiento temporal de un proyecto. En la figura 4 se indica el iagrama Causal de un modelo dinámico simplificado [9]. En este diagrama se

recogen las relaciones de influencia de las variables más importantes y los bucles de realimentación básicos que permiten modelar la evolución temporal de un proyecto software.

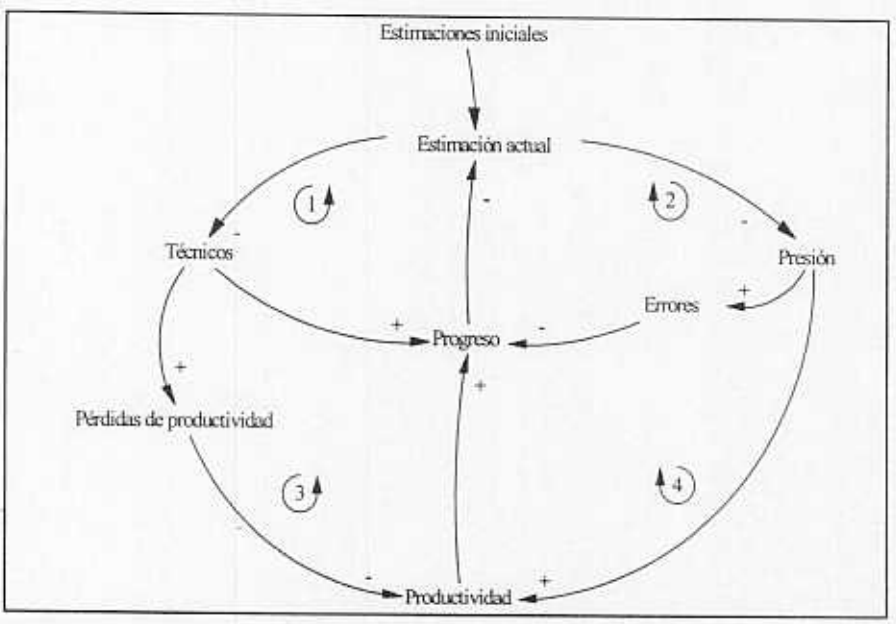


Fig. 4. Diagrama Causal Básico de un modelo dinámico simplificado para proyectos software.

Como podemos observar, aparecen cuatro bucles de realimentación. A continuación se describen brevemente cada uno de ellos:

Bucle 1: Las estimaciones iniciales de coste y de tiempo se obtienen, fundamentalmente, a partir del tamaño estimado del producto a desarrollar. A partir de éstas se procede a contratar los técnicos necesarios. Conforme el proyecto avanza se obtiene información sobre el estado del progreso del mismo. La comparación de los datos sobre la situación actual del proyecto con los datos estimados inicialmente puede, a menudo, sugerir una modificación en las estimaciones de tiempo y/o de coste del proyecto. La modificación de estas estimaciones puede venir acompañada de la necesidad de incorporar más técnicos al proyecto, especialmente si se detecta que éste se está retrasando. En definitiva, este bucle de realimentación regula el nivel de técnicos asignados al proyecto, en función del progreso estimado en el mismo en un momento dado.

Bucle 2: En este bucle se ilustran los efectos que producen la presión y los errores cometidos por los técnicos sobre el progreso del proyecto. La presión se obtiene al comparar los datos planificados con los estimados, en un momento dado. Los datos estimados dependerán de cuál es nuestra percepción real sobre el progreso del proyecto. Si los datos estimados superan a los planificados, estamos en una situación de presión positiva, es decir, nos queda para acabar el proyecto más tiempo y/o esfuerzo del que se había planificado. En general, una presión positiva se suele contrarrestar mediante la incorporación de más técnicos y/o mediante la realización de trabajo extraordinario. El trabajo extraordinario sostenido durante mucho tiempo favorece el factor de agotamiento entre los técnicos, con lo que la probabilidad de

neración de errores es mayor que en una situación de trabajo normal. El aumento de os errores incide directamente sobre el progreso del proyecto ya que cuanto mayor ea el número de errores que se cometen, mayor será el esfuerzo que habrá que dicar a la detección y corrección de los mismos, lo que produce una desviación de erzo desde las actividades de producción hacia las actividades de corrección. ando se detecta esta disminución del progreso se hace necesario un nuevo ajuste de estimaciones (coste y/o plazo) que tratarán de suavizar la presión cerrándose de a forma el bucle.

Bucle 3: En él se recoge el efecto que tiene la incorporación de técnicos nuevos bre la productividad del equipo. Cuando se decide incorporar a nuevos técnicos se ben tener en cuenta los efectos negativos que esta decisión puede tener sobre riables fundamentales del proyecto. Es indudable considerar que un incremento del mero de técnicos en el proyecto va a traducirse en un aumento de la productividad edia del equipo de trabajo y que, por tanto, el progreso del proyecto también mentará. Pero no se debe olvidar que la relación entre el número de técnicos que abaja en un equipo y la productividad no es lineal. Al incorporase nuevos técnicos al quipo de trabajo se producen unas pérdidas de productividad en el equipo que es cesario contemplar. Estas pérdidas se producen, por un lado, por el aumento de las eas de comunicación (oral y/o escrita) que se establecen en el equipo al aumentar el mero de técnicos que lo forman. Y por otro lado, el personal nuevo no tiene, rmalmente, la misma productividad que los técnicos que ya llevan algún tiempo abajando en el proyecto y, en general, cometen un mayor número de errores que los cnicos expertos. Por otro lado, los técnicos nuevos requieren de una etapa de decuación y/o formación para conocer a fondo la peculiaridades del proyecto.

Bucle 4: En este cuarto bucle, se recogen los efectos que tiene sobre la oductividad el hecho de que aparezca presión en un proyecto. Como se ha visto nteriormente, la presión es un elemento que tiende a romper el equilibrio del oceso. Cuando se detecta que el proyecto va retrasado, el equipo de trabajo tiende a crementar su productividad bien por eliminación de los tiempos muertos o por la alización de trabajo extraordinario, con lo que se produce un incremento en la oductividad que puede rebajar dicha presión. Sin embargo, ésta es una circunstancia e no puede mantenerse durante mucho tiempo, ya que aumenta el agotamiento de s técnicos. Por el contrario, si la presión es negativa los técnicos se comportan, en neral, de manera contraria: se produce una relajación, por lo que la productividad isminuye tendiendo a consumir la sobreestimación de recursos detectada. Por este otivo, si el valor de la presión negativa es alto se tiende a revisar las estimaciones on el fin de adaptarlas al valor estimado.

Los parámetros que modelan el sistema pueden se pueden ver en la figura 5: los arámetros relacionados con el proyecto (tamaño, número inicial de técnicos, etc.), os parámetros relacionados con la organización de desarrollo (retrasos medios en la ontratación y despido de los técnicos, etc.) y los parámetros de control de la imulación (duración e incremento temporal que se utilizará en la simulación del odelo dinámico).

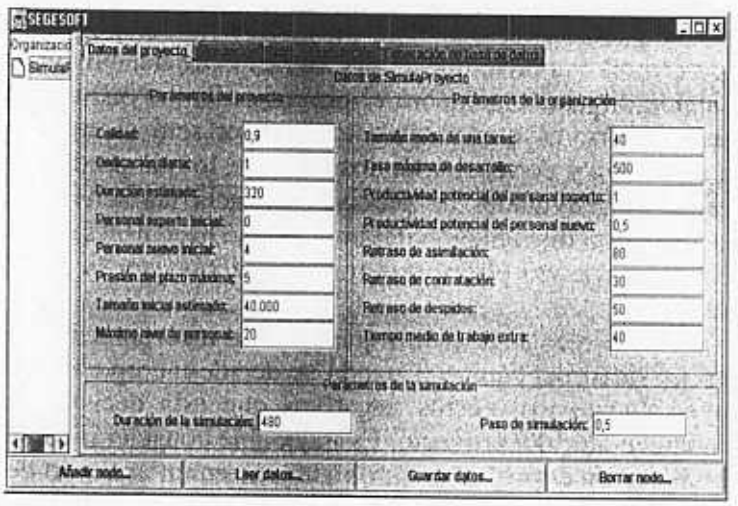


Fig. 5. Parámetros del proyecto

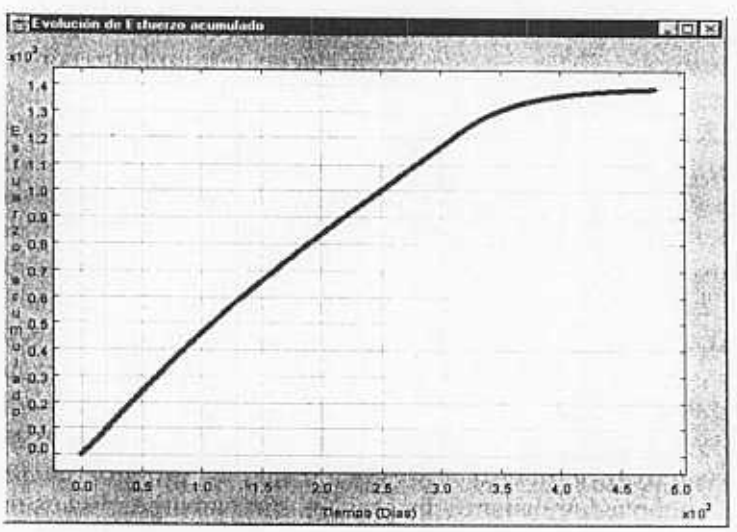


Fig. 6. Evolución del esfuerzo acumulado

Una vez introducidos los valores de estos parámetros se puede obtener la representación gráfica de la variable que interese analizar. En la figura 6 se muestra la evolución del esfuerzo acumulado del proyecto que se ha simulado.

Este módulo, mediante el modelo dinámico simplificado, permite simular el comportamiento del proyecto tantas veces como se desee y almacenar en una base de datos los valores obtenidos. Esta base de datos se utilizará posteriormente en otro módulo para aplicar las técnicas de aprendizaje automático. En cada simulación se obtiene un registro de la base de datos en el que se guardan los valores de los parámetros, obtenidos aleatoriamente dentro de un intervalo de valores posibles, y los valores finales de las variables del modelo [10] Para ello, previamente se debe indicar los parámetros que se deseen analizar y el intervalo de valores donde pueden moverse cada uno de ellos.

icación de la Minería de Datos

un proceso de minería de datos esperamos poder proporcionar una ción útil a partir del volumen de los datos generados mediante la simulación proyectos [11]. Existen algoritmos de aprendizaje para proporcionar reglas de tales como: C4.5 [12] (árboles de decisión), COGITO [13] (listas de n), a priori [14] (reglas de asociación), etc. Sin embargo, estas aproximaciones conjuntos de reglas para cada etiqueta discreta presente en la base de datos, bargo, las bases de datos generadas mediante el simulador de proyectos tienen quetas continuas; esfuerzo y tiempo de entrega. Esto hace imposible la ón inmediata de los algoritmos anteriores.

ello, el algoritmo debe empezar por calcular los valores que podríamos nar "buenos" para ambas variables siguiendo el criterio del director del to. La pantalla de entrada de datos para el módulo de generación de reglas de n, en un proyecto hipotético, la podemos ver en la figura 7. En ella se puede ar cómo al modificar los intervalos de valores de estas variables (EFFORT y ERYTIME, esfuerzo gastado y tiempo de entrega del proyecto ivamente), se está definiendo que se entiende por proyectos "buenos". Por , originalmente el valor de la variable EFFORT resultado de las simulaciones entra entre 120 y 5279 técnicos-días, y el de la variable DELIVERYTIME 9 y 273 días. Sin embargo, sólo consideraremos proyectos válidos (41 según observar en la figura 7 abajo a la derecha) aquéllos cuyos valores simulados [120, 273] y [149, 151] respectivamente.

ste proceso convertimos la Bases de Datos (BD) resultado de la simulación ecto en una BD etiquetada, de forma que cada registro está formado por los de los parámetros del modelo y una etiqueta de "bueno" o "no bueno", en de que los valores de las variables esfuerzo y tiempo se encuentren en los os señalados en la interfaz. Una vez realizado este preprocesado, diseñamos o algoritmo de aprendizaje debido a que los algoritmos anteriormente ados obtienen reglas de decisión para cada etiqueta, mientras que nosotros amos interesados en encontrar reglas para proyectos "buenos". Es decir, ente los algoritmos de aprendizaje supervisado buscan reglas para todas las existentes en la BD, sin embargo en el caso que nos ocupa, el gestor ente sólo estará interesado en reglas para la etiqueta "bueno":

vez que las etiquetas son generadas, se proporciona al algoritmo cuántas seamos generar y la tasa de error permitida. Por último, un valor debe indicar de pruebas a efectuar: dado que el proceso de búsqueda es probabilístico indica la cobertura de la búsqueda realizada, es decir, el número de puntos ados en el proceso de generación inicial de las reglas.

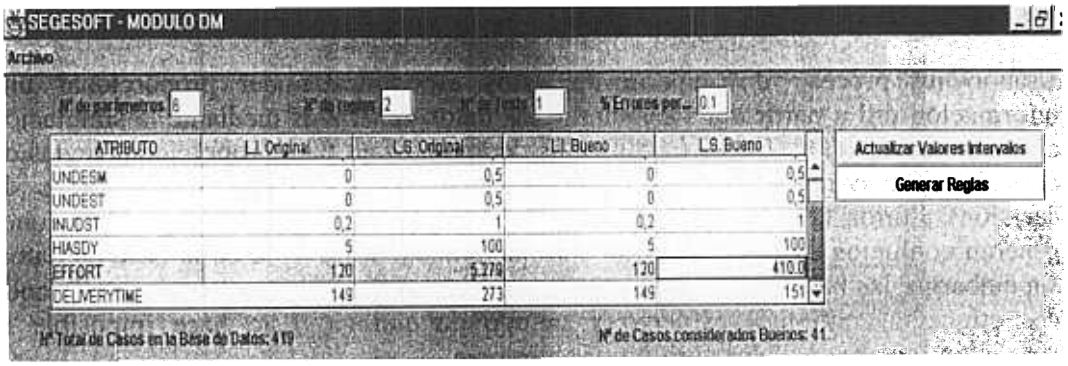


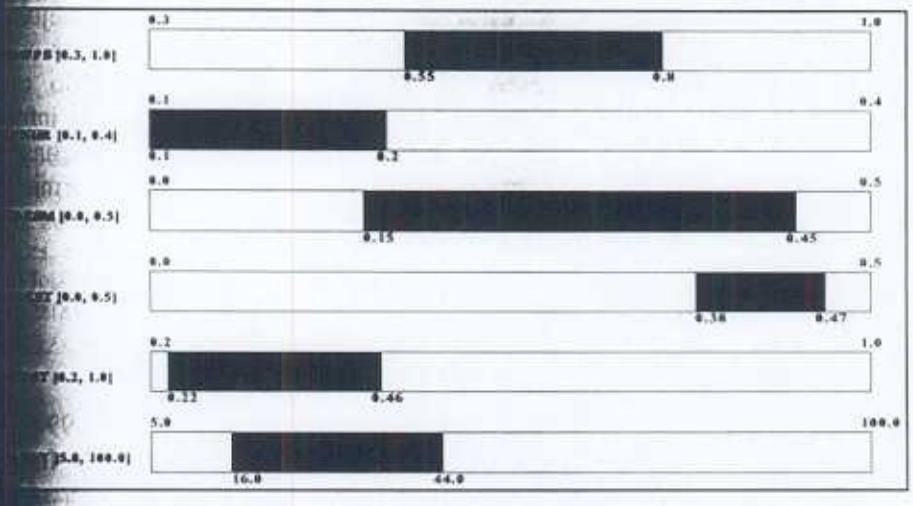
Fig. 7. Interface del módulo de data mining

A continuación, describimos brevemente el algoritmo que se implementó en este módulo (ver figura 8). Las reglas de decisión son generadas para informar si la ejecución del proyecto se realiza conforme a las estimaciones iniciales y deben proporcionar un conjunto de decisiones a tomar para evitar la violación de éstas. El algoritmo sigue una estrategia de escalada monótona con multiarranque.

```
Para cada regla inicial r
 Seleccionar aleatoriamente un ejemplo e de BD
 Para cada parámetro p de BD
    Ordenar BD por p crecientemente
    Buscar los ejemplos más cercanos a e (menor y mayor)
    con diferente clase
    Establecer ambos valores como extremos del intervalo
    de ese parámetro
 Para cada regla r que pueda cubrir más ejemplos de BD
    Elegir aleatoriamente un parámetro p
    Expandir el parámetro p por un extremo (izquierda o
    derecha)
```

Fig. 8. Pseudocódigo

Un ejemplo de salida gráfica que se obtiene en este proceso de aprendizaje se muestra en la figura 9. Dado que estamos interesados en conseguir "buenos" proyectos, los intervalos resaltados en negro indican los posibles valores que deben tomar los parámetros del proyecto para finalizarlo en las condiciones iniciales estimadas al principio.



sentación gráfica de la primera regla

icado de la figura 9 es el siguiente: si el parámetro ADMPPS (dedicación personal) está en el 55% y el 80% y TRPNHR (dedicación media de los expertos a formación) está entre el 10% y el 20% y UNDESM nación inicial del esfuerzo estimado) está entre 15% y 45% y UNDEST nación inicial del número de tareas estimadas) está en [38%, 47%] y (infraestimación inicial del número de técnicos estimados) está en [22%, HIASDY (retraso medio en la contratación y adecuación de los técnicos está entre [16, 44] días entonces el proyecto en cuestión evolucionaría según terio definido como "bueno" por parte del responsable.

demás las reglas pueden filtrarse considerando sólo los atributos cuyos intervalos recogen los valores de la estimación inicial para la simulación. Es decir, si el valor icial del parámetro INUDST se encuentra en el rango [22%, 46%] entonces la ondición sobre este parámetro puede desaparecer de la regla anterior, implificándose su lectura.

4 Conclusiones y Desarrollos Adicionales

Un uso inteligente de los datos del proyecto puede tener considerables efectos en los resultados del proyecto. Esto a su vez depende de la habilidad y experiencia del director del proyecto para abordar múltiples fuentes de información. El entrenamiento en la gestión de proyectos software es un requisito para avanzar en la profesión de ingeniero del software. Este entrenamiento se puede conseguir mediante el uso de entornos educativos que incorporen métodos de extracción y análisis de información.

Hemos presentado en este trabajo la primera versión de un entorno que ha sido desarrollado con el propósito de integrar diferentes aspectos de la gestión de proyectos. Uno de los principales beneficios de nuestro enfoque es que se construye con la intención de explorar el uso de diferentes tipos de técnicas desarrolladas formalmente para otras disciplinas de la ingeniería.

Los posteriores desarrollos y esfuerzos se dirigirán hacia la mejora de la visualización de toda clase de datos, uso de la información de múltiples orígenes o fuentes, la introducción de nuevos métodos, etc. Hemos detectado como un prerrequisito básico la falta de una metodología para interpretar toda la información que el director de proyecto reúne. Nuestra herramienta, además de conseguir un incremento en la información presente en las herramientas anteriores, permite evaluar acciones correctivas en forma de reglas de gestión para alcanzar los objetivos deseados.

Agradecimientos

El proyecto SEGESOFT está subvencionado por CICYT: TIC 99-0351. La participación de SYSECA Cantábrico (ahora denominada THALES Information System) ha sido de gran ayuda para nuestro trabajo.

5 Referencias

1. Mandl-Striegnitz P., et al.: Simulating Software Projects - An Approach for Teaching project Management. En Proc. of INSPIRE'98, (1998) 87-98
2. Rodrigues, A.G., Williams, T.M.: System dynamics in software project management: towards the development of a formal integrated framewor. European Journal of Information systems, **6** (1997) 51-66
3. Fayyad, U., Piatetsky-Shapiro, G., Smyth P. The KDD Process for Extracting Useful Knowledge from Volumes of Data. Comm. of the ACM **39(11)** (1996) 27-34
4. Caputo, K.: CMM implementation guide. Addison-Wesley (1998)
5. El Emam, K., Drouin, J.-N., Melo W. (Eds.): Spice: The Theory and Practice of Software Process Improvement and Capability Determination. IEEE Computer Society Press (1998)
6. PROFES http://www.profes.org/
7. Satpathy, M., Harrison, R., Snook, C., Butler, M.:A Generic Model for Assessing Process Quality. En Proc. of 10th International Workshop on Software Measurement (IWSM'00), LNCS, Springer, (2000) 94-110
8. Paul, R.A., Kunii, T.L., Shinagawa, Y., Khan, M.F.: Software Metrics Knowledge and Databases for Project Management. IEEE Transactions on Knowledge and Data Engineering, **11(1)** (1999) 255-264
9. Ruiz, M., Ramos, I, Toro, M.: A simplified model of software project dynamic. The Journal of System and Software. Elsevier Science Inc, 2001.
10. Ruiz, M.,Ramos, I.: A Dynamic Estimation Model for the Early Stages of a Software Project. En Software Process Simulation Modelling Workshop (Prosim2000)
11. Ramos, I., Aguilar, J., Riquelme, J.C., Toro, M.: A new method for obtaining software project management rules. En Proc. of Software Quality Management VIII (2000) 149-160
12. Quinlan J.R.: C4.5: Programs for machine learning. Morgan Kaufmann, 1993.
13. Riquelme J.C., Aguilar, J.S., Toro M.: Discovering hierarchical decision rules with evolutive algorithms in supervised learning. Int. J. of Computer, Systems and Signals, 1 (1) (2000) 73-84
14 Agrawal, R. Imielisnki, T., Swami, A. : Mining association ruls between sets of items in large databases. Proc. ACM SIGMOD Conf. on Management of Data, (1993) 207-216