

CLEFTCase: Herramienta CASE con soporte colaborativo[†]

Miguel Ángel Conde González¹, Iván Álvarez Navia², Francisco J. García Peñalvo²

¹Departamento de I+D+i CLAY Formación Internacional
C/Hoces del Duratón nº 57, 37008, Salamanca, España
miguelangel@clayformacion.com

²Dpto Informática y Automática – Universidad de Salamanca
Facultad de Ciencias. Plaza de los Caídos S/N 37008, Salamanca, España
{inavia, fgarcia}@usal.es

Resumen. En este artículo se presenta CLEFTCase, una herramienta CASE con soporte colaborativo, actualmente en desarrollo en el Dpto. de Informática y Automática de la Universidad de Salamanca. Consiste, básicamente, en: un servidor con una serie de componentes que gestionan el almacenamiento de diagramas e información adicional generada, información de administración, etc., además de un conjunto servicios y protocolos de discusión que implementan el soporte colaborativo, y un cliente que, accediendo a estos servicios, proporciona las herramientas de diseño que permiten a un equipo de desarrolladores colaborar en la elaboración de un conjunto de diagramas de modelado.

Palabras clave: CSCW, Collaborative Design, CASE Tools.

1 Introducción

Las herramientas CASE (*Computer Assisted Software Engineering*) constituyen uno de los ejemplos más paradigmáticos de la evolución del software en los últimos años. Del objetivo inicial, herramientas software capaces de generar software automáticamente, se ha pasado a un objetivo más realista, proporcionar soporte a los desarrolladores en el análisis, especificación y diseño del sistema software. Dada la importante presencia de estas herramientas en el proceso de desarrollo de software en el ámbito profesional, se hace imprescindible para un ingeniero una formación en los conceptos, técnicas y metodologías que utilizan. Así las sesiones prácticas de la asignatura Ingeniería del Software, impartida en la titulación Ingeniería Técnica en

[†] Este trabajo está parcialmente soportado por el Ministerio de Educación y Ciencia a través del proyecto de investigación KEOPS (TSI2005-00960).

Informática de Sistemas de la USAL, se utilizan diferentes herramientas, tanto comerciales como de libre distribución, algunas desarrolladas dentro del Dpto. de Informática y Automática de la USAL [1].

La utilización de herramientas comerciales supone un gasto considerable que, en algunas ocasiones, resulta difícilmente abordable por una institución académica. De ahí que en dicho Departamento se plantease el desarrollo de este tipo de herramientas con una clara orientación docente. Adicionalmente, el desarrollo a medida permite centrarse en el uso de determinadas metodologías e, incluso, variaciones de las mismas, consiguiendo una mayor adaptación a las necesidades docentes.

Por otro lado, el desarrollo del software es fundamentalmente un trabajo en equipo. Existen múltiples estudios que indican que el trabajo en equipo, en el que unos individuos interactúan y colaboran con otros, ocupa entre un 70% y un 85% del esfuerzo total de desarrollo. En las metodologías modernas de desarrollo del software se pone en manifiesto la necesidad de la presencia y colaboración de diversos expertos en las diferentes fases del desarrollo. Estos expertos deberán compartir información, realizar tareas, tanto independientes como conjuntas, y por supuesto crear un producto único común. En este marco de trabajo se observa como la posibilidad de coordinación del equipo trabajando sobre un único problema simultáneamente, o de creación de múltiples grupos de trabajo que se encargarán de resolver diversos problemas, es un aspecto crítico en las herramientas CASE [5]. Por lo tanto, además de los aspectos relacionados con las técnicas y metodologías, es imprescindible un entrenamiento/instrucción del futuro ingeniero en el trabajo en grupo/equipo. De ahí el esfuerzo realizado en construir una herramienta CASE con soporte colaborativo, cuyos resultados se presentan en este artículo.

La herramienta presentada, CLEFTCase, consiste en una arquitectura abierta que permite el desarrollo de diagramas de modelado que apoyen la fase de diseño en el ciclo de vida de un proyecto software realizado por un equipo de desarrollo. Se ha hecho un uso extensivo de componentes, buscando una mayor flexibilidad y extensibilidad del sistema.

A continuación se hará una introducción a la experiencia previa de desarrollo, antecedentes de CLEFTCase, para seguir con una descripción más detallada de la arquitectura global de la herramienta, describiendo brevemente cada uno de los módulos construidos. Seguidamente se pasará a explicar el protocolo diseñado para soportar una sesión colaborativa. Para terminar con unas conclusiones.

2 Antecedentes de CLEFTCase

Son varias las aplicaciones software que se han desarrollado y que se han venido utilizando en la docencia de la asignatura Ingeniería del Software de tercer curso de la Ingeniería Técnica en Informática de Sistemas de la Universidad de Salamanca. Las primeras herramientas desarrolladas centraban los esfuerzos en cubrir las técnicas de modelado más representativas, surgiendo así las siguientes aplicaciones: ADAM CASE (realización de Diagramas de Flujo de Datos utilizando la notación Yourdon y Diagramas de Clases utilizando la notación UML), ER CASE (diagramas de Entidad-Relación), CRC CASE (Clase, Responsabilidad y Colaboración) y GestCat

(documentación de requisitos utilizando la notación tabular propuesta por el método de elicitación de requisitos de Durán y Bernárdez) [1].

Aunque se han utilizado de forma extensiva en la docencia de la asignatura ya indicada, este conjunto de herramientas adolece de dos problemas fundamentales:

- Cada herramienta es independiente del resto, no hay integración entre las mismas, lo que influyen negativamente a la hora de proporcionar al alumno una visión global del proceso de análisis y diseño del software.
- Añadir nuevas técnicas de modelado implica la creación de nuevas herramientas o la modificación de las existentes, pero sin resolver la ausencia de integración con las existentes.

Por estas razones, y apoyándonos en la experiencia obtenida, tanto en el desarrollo de estas herramientas, como en su utilización en tareas docentes, se plantea la creación de una nueva línea de trabajo consistente en el desarrollo de una plataforma CASE, LEFTCase, constituida por un contenedor y un conjunto de componentes. Dichos componentes se empotran debidamente en el contenedor, implementan distintas técnicas de modelado, y el contenedor permite la integración de todas las tareas de análisis y diseño de una forma natural, proporcionando a la vez una interfaz de usuario coherente y uniforme entre todos los componentes, aparte las diferencias intrínsecas impuestas por las diferentes técnicas de modelado. Además la arquitectura propuesta facilita la creación e integración de nuevos componentes, que añadan nuevas funcionalidades, de una forma sencilla. Los componentes creados son cinco, cada uno ofreciendo el apoyo adecuado para una técnica de modelado: Diagramas de Flujo de Datos, Diagramas Entidad-Relación, Diagramas de Clases UML, Diagramas de Interacción UML y Diagramas de Estados UML.

Dado que la herramienta está claramente orientada a fines docentes, se planteó desde un principio la necesidad de trabajar con entornos abiertos y de libre distribución, tanto para el desarrollo como para la explotación. Así la plataforma escogida es GNU/Linux (aunque, en principio cualquier sistema UNIX que cuente con las herramientas GNU sirve) y como entorno visual se escoge GNOME y su plataforma de desarrollo. GNOME es un entorno de escritorio de libre distribución y que funciona en sistemas UNIX y, fundamentalmente, sistemas GNU/Linux. Este entorno de escritorio incluye una completa plataforma de desarrollo de la que destaca su sistema de componentes, Bonobo, que facilita en gran medida el desarrollo de aplicaciones basada en este paradigma. El sistema de comunicación entre objetos distribuidos está basado en el estándar CORBA, y aunque la implementación utilizada, ORBit, si soporta la comunicación entre objetos en diferentes máquinas, sin embargo, el sistema de componentes no ofrece soporte de servicio de nombre de objetos (obtención de las referencias a objetos remotos).

LEFTCase supuso un importante esfuerzo de desarrollo y una buena plataforma de desarrollo, dado su carácter abierto basado en componentes, además de una excelente experiencia de desarrollo. Se plantea la evolución de la misma en dos líneas:

- Incorporación de nuevos componentes que implementen nuevas técnicas de modelado.
- Aprovechar todo lo aprendido en su desarrollo y centrarse en nuevos paradigmas de trabajo que, por otro lado, tienen especial relevancia en el desarrollo del software: el trabajo en equipo y herramientas CASE con soporte colaborativo.

Dado que la importante experiencia y buenos resultados en la creación de componentes, se opta por centrar los nuevos esfuerzos en la segunda línea, el soporte colaborativo. Esta decisión implica plantear fuertes cambios en la arquitectura de LEFTCase [2], aunque se mantienen elementos fundamentales y se puede reutilizar gran parte del software desarrollado.

Surge así CLEFTCase, cuyo objetivo fundamental es elaboración de una arquitectura abierta para permitir el desarrollo de un conjunto de diagramas de modelado, que apoyen la fase de diseño en el ciclo de vida de un proyecto software realizado por un equipo de desarrollo. Y más concretamente el esfuerzo se ha centrado en el desarrollo de una arquitectura que proporcione a los desarrolladores capacidades de colaboración y consenso, mediante la definición de un protocolo y la construcción de herramientas. De este objetivo fundamental se derivan, de forma muy resumida, las siguientes características funcionales:

- Soporte para la creación de diagramas siguiendo diferentes técnicas de modelado, además de generación automática de la documentación técnica de análisis y diseño.
 - Soporte para el trabajo colaborativo, en diferentes aspectos:
 - Existencia de un repositorio común para todos los diagramas generados, incluyendo política de versionado, además de la posibilidad de recuperar versiones anteriores de un diagrama.
 - Posibilidad de trabajo individual sobre un diagrama, o bien varios usuarios sobre un mismo diagrama o una parte del mismo.
 - Comunicación entre usuarios.
 - Control de concurrencia.
 - Definición de un protocolo de comunicación entre usuarios que posibilite la realización de sesiones de consenso, incluyendo medidas de corrección para situaciones imprevistas (cortes de red, etc.).
 - Infraestructura para la creación de pizarras virtuales que sirvan de base para las discusiones en sesiones de consenso, incluyendo un contexto de sesión que permite mantener información sobre los usuarios participantes. Se hace uso de los servicios del repositorio para hacer persistentes los datos de las pizarras creadas, permitiendo que sean incorporados a los diagramas por los desarrolladores, además de posibilitar la navegación por los diferentes estados por los que pasa una sesión.
 - Gestión de proyectos y labores de administración.
 - Gestión de usuarios, autenticación y sesiones. Seguridad y permisos de acceso.
- A continuación se dará una visión global del diseño de CLEFTCase.

3 Arquitectura de CLEFTCase

Aunque la arquitectura de CLEFTCase hereda partes importantes de los anteriores desarrollos, más concretamente, la orientación a componentes utilizada en LEFTCase, se ha creado una nueva aproximación que resuelve todos los aspectos relacionados con las tareas de colaboración, lo que unido a los importantes cambios introducidos en las nuevas versiones de la plataforma de desarrollo, GNOME 2, ha supuesto partir de cero.

Así se parte de una arquitectura clásica cliente/servidor, en la que las diferentes funcionalidades se distribuyen de la siguiente manera:

- Cliente: representa la interacción con el usuario. Básicamente consta de un contenedor que albergará diferentes componentes, de un manera análoga a como se diseñó LEFTCase, pero al que se le añaden nuevos componentes, encargados de acceder a los servicios del servidor y relacionados con las tareas colaborativas.
- Servidor: es el encargado de proporcionar una infraestructura de colaboración y repositorio centralizado de información, así como todo lo relacionado con la gestión de usuarios, seguridad, gestión de sesiones.

Esta arquitectura aparece reflejada en la Figura 1 mostrando, con un cierto nivel de detalle, los paquetes que implementan las funcionalidades atribuidas a cada parte, cliente y servidor.

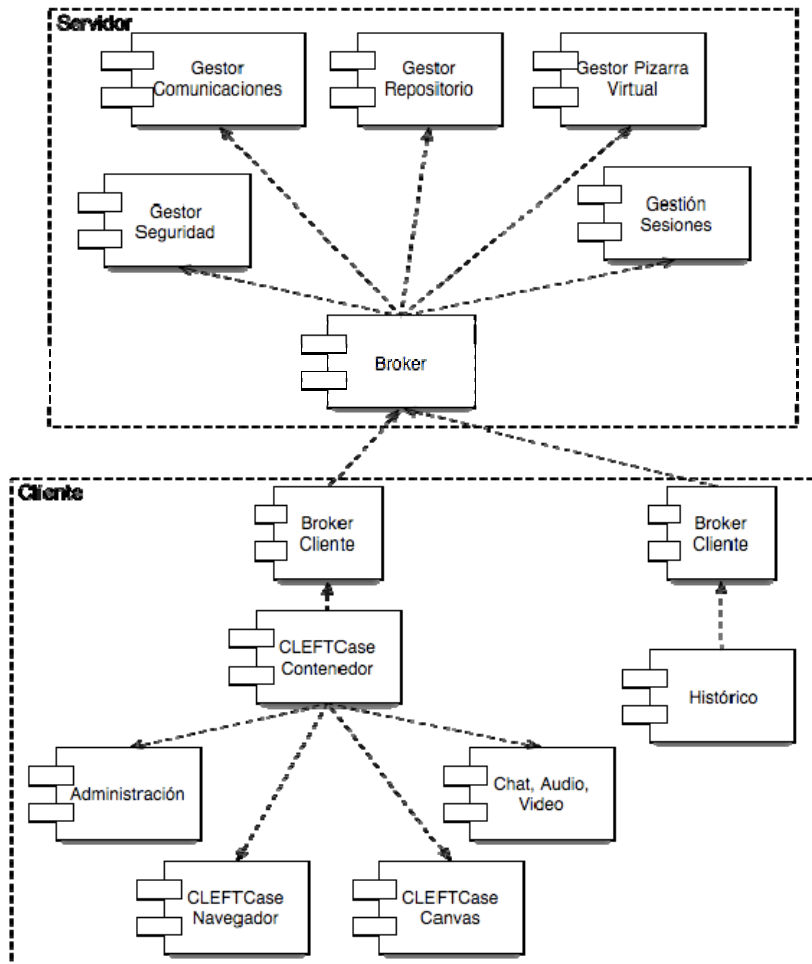


Fig. 1. Visión global del diseño de CLEFTCase

3.1 Soporte colaborativo

Este es el aspecto central del desarrollo y el que ha determinado la arquitectura planteada, por lo que se describirá con un poco más de detalle. Comenzaremos con la definición de los diferentes tipos de usuario (Figura 2):

- **Desarrollador:** todas aquellas personas físicas que toman parte en el desarrollo de uno o más proyectos software mediante el uso de la herramienta CLEFTCase.
- **Administrador:** se encarga de la instalación, mantenimiento y gestión de los usuarios que van a utilizar la herramienta.
- **Jefe de proyecto:** desarrollador parte del equipo que tendrá la responsabilidad de liderar el mismo durante el desarrollo del proyecto, creando, gestionando, planificando, junto con el administrador, el equipo y distribuyendo tareas, dirigiendo las discusiones en las sesiones de consenso, etc.
- **Propietario:** desarrollador que se encarga de la generación de una sección del proyecto y que tiene el control de los permisos de acceso de los elementos que pertenecen a dicha sección.
- **Desarrollador temporal:** desarrollador que obtiene temporalmente los permisos para usar/modificar un elemento de una sección del proyecto de la que no es propietario.
- **Invitado:** usuario con permisos de lectura sobre la pizarra virtual, de manera que puede observar el trabajo que se está realizando.

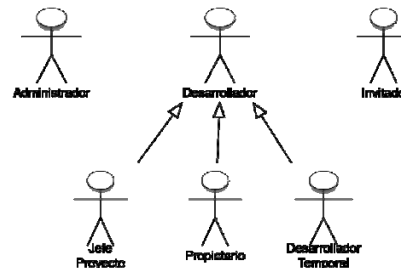


Fig. 2. Tipos de usuario

Una vez definidos los tipos de usuario, se pasa al estudio de los diferentes escenarios de colaboración que se pueden presentar. Como se ha comentado antes, este es el principal objetivo que se persigue: el desarrollo de una plataforma abierta que permita ir incorporando, de forma sencilla, componentes para elaboración de diagramas de modelado con posibilidad de colaboración entre los diferentes miembros de un equipo de desarrollo. Por lo tanto, el esfuerzo de desarrollo se centró en la propia plataforma, dejando un poco de lado los componentes que se encargan de la elaboración de los diagramas. Así la definición de los protocolos de colaboración ha sido objeto de estudio detallado, para conseguir que el mismo contemple todas las posibles situaciones, incluyendo aquellas que resulten imprevistas. Para ello se:

- definió un modelo de negocio con las actividades deseables para el protocolo, utilizando para ello diagramas de actividad,
- se evaluó la evolución de dichas actividades mediante árboles de tareas concurrentes [4], como el que se muestra en la Figura 3, en este caso análisis de las tareas involucradas en el intercambio de turno.
- Finalmente, se definieron los escenarios colaborativos a partir de los diagramas anteriores. En la Figura 4 se puede observar el correspondiente diagrama colaborativo para el escenario de intercambio de turno, elaborado a partir del análisis realizado, como se indica en el punto anterior.

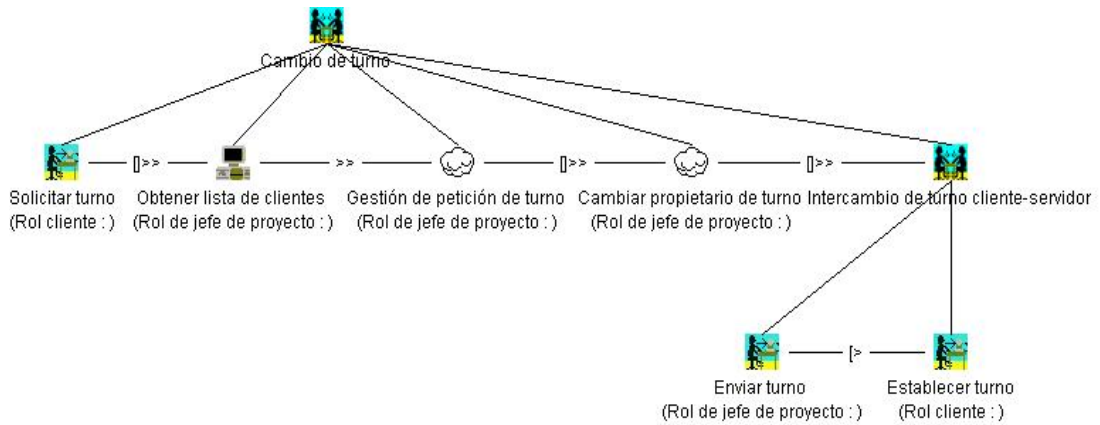


Fig. 3. Árbol de tareas concurrentes de intercambio de turno

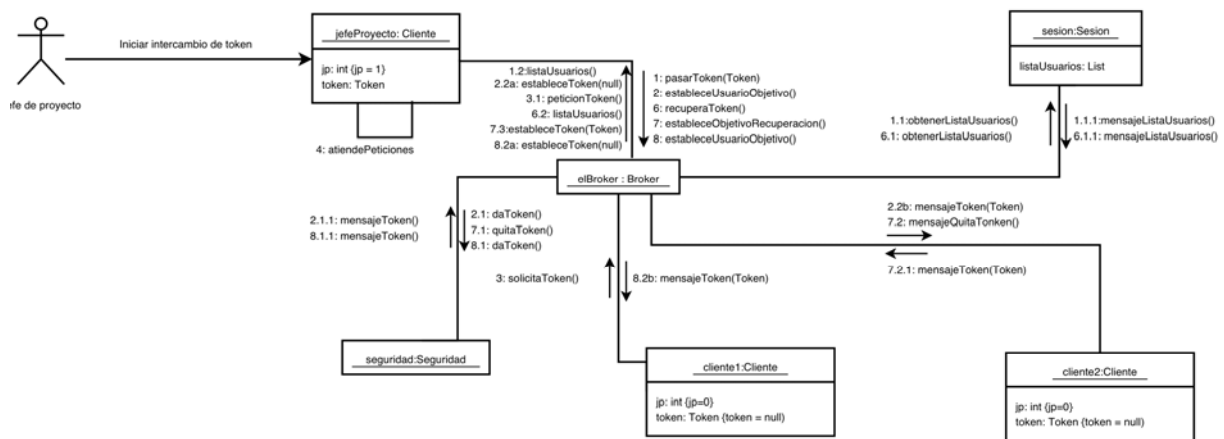


Fig. 4. Diagrama colaborativo de intercambio de turno

Como resultado se obtiene los paquetes y servicios necesarios para proporcionar el soporte colaborativo al resto de componentes del sistema y que se detallan en la Figura 1. Cabe destacar el paquete “Pizarra virtual”, dentro del servidor, encargado de proporcionar el soporte para discusiones y sesiones de consenso. También el paquete “Gestor repositorio” que ofrece servicios de almacenamiento de diagramas, control de versiones, tanto estables de equipo de desarrollo, como privadas de cada desarrollador, y los diferentes estados de una discusión en una pizarra virtual. En el cliente, los paquetes “Navegador” e “Histórico” hacen uso intensivo de los servicios ofrecidos por los dos anteriores, el primero para acceder a diagramas, versiones de los mismos, tanto propios de un desarrollador (Propietario) como a los de otro desarrollador que así lo ha autorizado (Desarrollador temporal), y posibilitar la realización los trabajos de edición de diagramas. De esta parte se encarga el paquete “Canvas”, que proporciona toda la funcionalidad como editor. También hará uso de los servicios del paquete “Pizarra virtual” en las sesiones de discusión y consenso, en

ese momento el editor se convierte en una pizarra compartida por todos los usuarios que participan de la discusión. El paquete “Histórico” es un navegador específico que permite al usuario visualizar los diferentes estados por los que ha pasado una discusión y visualizar los diferentes cambios que se han producido en un diagrama a lo largo de la misma, careciendo de capacidades de edición. Una vez el usuario ha seleccionado un estado, el diagrama correspondiente pasa al editor, convirtiéndose así en el objeto actual de edición y discusión.

4 Descripción de CLEFTCase

CLEFTCase es una herramienta CASE que posibilita la creación, edición, modificación, verificación y almacenamiento de diagramas de modelado por parte de un equipo de desarrolladores.

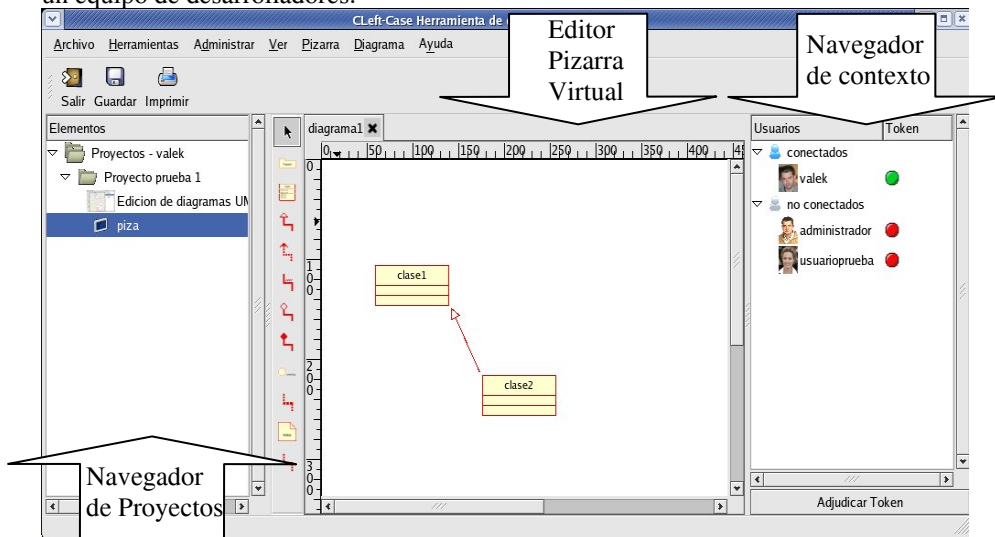


Fig. 5. Sesión de consenso

Por lo tanto, desde un punto de vista funcional, ofrece dos tipos de comportamiento:

- Como editor de diagramas de UML [3], un usuario puede realizar diagramas, de forma individual o de forma conjunta, como miembro de un equipo de desarrollo. Dichos diagramas son almacenados en un repositorio centralizado al que tienen acceso todos los miembros del equipo. El usuario “Jefe de Proyecto” se encarga de organizar las tareas de desarrollo y de dirigir las discusiones.
- Como herramienta de colaboración (ver Figura 5), se ofrece el soporte necesario para que varios usuarios puedan trabajar sobre un mismo diagrama, estableciendo los permisos de acceso necesarios. Además, se ofrece la posibilidad de crear pizarras virtuales para la realización de sesiones de discusión, de manera que todos los participantes en la discusión tengan en el editor el mismo diagrama y, mediante un sistema de intercambio de turno dirigido por el “Jefe de proyecto”, cada uno

pueda hacer las aportaciones y modificaciones que estime oportuno. El “Jefe de proyecto” será el responsable de dirigir dicha discusión y de aceptar o rechazar las diferentes modificaciones hasta alcanzar un resultado final, que podrá pasar al repositorio como “versión estable”.

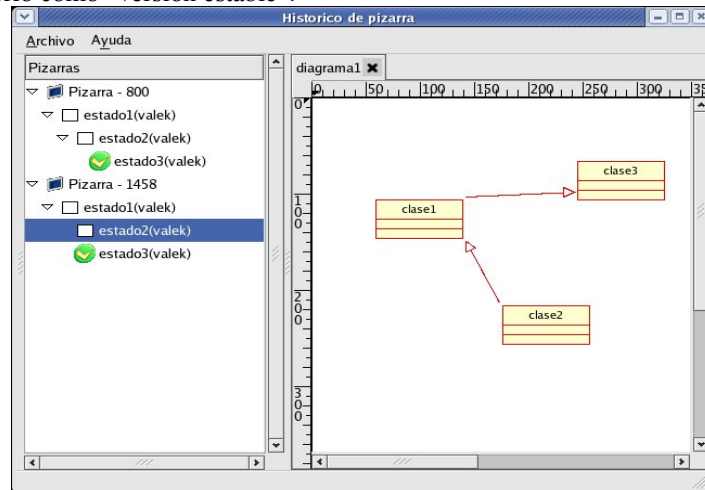


Fig. 6. Histórico de pizarra

Como apoyo a las sesiones de discusión, se proporciona un navegador de histórico de estados de la misma (Figura 6), de manera que los usuarios siempre pueden volver a estados anteriores de la discusión e iniciar una nueva rama de discusión. Este navegador no permite la edición, sólo la navegación y visualización de diagramas. Una vez un usuario selecciona un estado, éste pasa a la pizarra virtual compartida por todos los usuarios y donde si tienen posibilidad de editar los nuevos cambios.

Otro elemento de la interfaz que sirve de apoyo a las sesiones de discusión es el navegador de contexto, que muestra un listado de usuarios participantes en la discusión. Se han planteado dos vistas, una para el “Jefe de proyecto”, en la que se listan todos los usuarios miembros del proyecto, mostrando su estado actual (conectado o no conectado), y que usuario está en poder del turno en cada momento, y por lo tanto, tiene realmente permiso para editar modificaciones en el diagrama. La segunda vista, la del resto de usuarios, sólo proporciona el listado de usuarios realmente conectados, además de indicar que usuario está en posesión del turno. A través de este elemento se podrá, además, solicitar el turno al jefe de proyecto para realizar modificaciones o para iniciar una rama diferente, a partir de otro estado anterior, previamente seleccionado mediante el navegador de histórico de estados. La posesión del turno se representa mediante una metáfora de semáforo, una esfera verde indica posesión del turno, una esfera amarilla indica solicitud del mismo realizada y una esfera roja indica no posesión. Una solicitud de turno no implica que el usuario lo vaya a tener, es el “Jefe de proyecto” quien decide el usuario que estará en posesión del mismo en cada momento.

En todo momento existe un control del sistema sobre las comunicaciones y los usuarios conectados, de manera que es capaz de reaccionar de forma correcta ante situaciones no previstas como, por ejemplo, un corte en la red que deje a uno o varios

usuarios desconectados de la discusión. Automáticamente aparecerán reflejadas estas situaciones en el navegador de contexto, informando así al resto de los participantes.

5 Conclusiones

En este artículo se ha presentado CLEFTCase una herramienta CASE para el desarrollo de modelos UML de forma colaborativa. Sobre la base arquitectónica de una herramienta CASE basada en componentes, se ha desarrollado esta nueva versión que pone el énfasis en la posibilidad de que los miembros de un equipo de desarrollo colaboren para la creación de los diseños.

Más concretamente, en este trabajo se ha puesto el énfasis en presentar no tanto la parte de creación colaborativa como la parte discusión y posterior aceptación de un diseño por parte del jefe de proyecto. Para ello, la herramienta cuenta con una herramienta de pizarra compartida, a la que todos los miembros del equipo pueden sumarse con la opción de participar en la discusión activamente, y con un protocolo de discusión basado en el cambio de turno moderado por el jefe de proyecto (en las figuras 3 y 4 se representa específicamente el escenario de cambio de turno).

Desde un punto de vista docente, las posibilidades que abre esta herramienta son muy amplias. Con el conjunto de herramientas que se había desarrollado, se cubría de una forma más que aceptable el conjunto de competencias, destrezas y habilidades propias del manejo de una herramienta CASE en las fases de análisis y diseño, soportando un determinado lenguaje de modelado, ya fuera orientado a objetos o estructurado. Pero con CLEFTCase, además de lo anterior, se puede formar a los ingenieros de software en habilidades transversales derivadas del trabajo en grupo, el *role playing*, y las sesiones de discusión y contexto, todas ellas competencias imprescindibles y muy valoradas por la industria del software.

Referencias

- [1]García, F. J., Álvarez, I.: Left CASE – A Free Software Component-based CASE Tool for Software Engineering Practice Support. Proceedings of the 33rd ASEE/IEEE Frontiers in Education Conference. IEEE-CS Press. (2003). S2C-7-S2C-12.
- [2]García Peñalvo, F. J., Álvarez Navia, I.: Plataforma CASE Basada en Componentes para la Docencia de Ingeniería del Software. Actas del IE-2002 (2002).
- [3]OMG: Unified Modeling Language: Superstructure. Version 2.0. Object Management Group Inc. Document formal/05-07-04. (2005). <http://www.omg.org/cgi-bin/doc?formal/05-07-04>.
- [4]Paternò, F., Mori, G., Santoro, C.: Support for Developing and Analyzing Task Models for Interactive System Design. IEEE Transactions on Software Engineering, 28 (2002), 797-813.
- [5]Vessey, I., Sravanapudi, A.: CASE Tools as Collaborative Support Technologies. Communications of the ACM 38 (1995) 83-95.