

# Un Modelo para el Desarrollo Cooperativo de Software

Ph.D. Angélica De Antonio Jiménez<sup>1</sup>  
Ing. Marco Villalobos Abarca<sup>2</sup>

## Resumen

En este artículo se hace una descripción de un proyecto de investigación que se está llevando a cabo dentro del contexto del desarrollo de software educativo. Este proyecto está orientado a la construcción de un modelo y de un entorno de desarrollo. Hasta el momento se ha logrado construir un modelo genérico basado en la filosofía del trabajo cooperativo, el cual es discutido en este trabajo. En el futuro, se derivará de éste un modelo especializado para el desarrollo de software educativo, así como un entorno de desarrollo que soporte tal modelo.

**Palabras Clave:** Modelo para el desarrollo de Software, Desarrollo Cooperativo, Software Educativo.

## 1. INTRODUCCIÓN

Las necesidades crecientes de uniformidad, calidad y productividad en los procesos de desarrollo de software educativo (SwE) (Squires *et al.*, 1997) junto con las nuevas oportunidades para los proyectos de entrenamiento de gran escala (Méndez *et al.*, 2001) están

haciendo del desarrollo cooperativo y distribuido uno de los sectores más prometedores en el mercado de la tecnología educativa (Ulloa, 2001). El desarrollo de material educativo a través de grupos cooperativos ha cobrado relevancia en el ámbito europeo en los últimos años. En este sentido, existen varios proyectos e investigaciones dirigidas al desarrollo cooperativo de aplicaciones educativas. Algunos de tales trabajos se pueden encontrar en (Ulloa, 2001, Twidale *et al.*, 1994, Qu *et al.*, 2001, Ateyeh *et al.*, 2000).

En general, el diseño y la producción de SwE de calidad es un proceso de trabajo en grupo que suele ser complejo y costoso. Este proceso es llevado a cabo por un equipo multidisciplinario que incluye expertos de varias áreas, como es el caso de los Diseñadores Instruccionales, Especialistas en Contenido, Especialistas en Informática, Artistas, Especialistas en Comunicaciones, Especialistas en Producción Multimedia, y Especialistas en Pruebas y Control de Calidad. Cada miembro del equipo juega un rol específico, y por tanto hace contribuciones específicas. De esta manera, surge la necesidad de coordinar los diferentes roles y sus contribuciones con una visión amplia de todo el proceso de desarrollo de SwE.

A nuestro juicio existen tres líneas de investigación orientadas hacia el Desarrollo Cooperativo de Software. La primera relacionada con el análisis y descripción del trabajo en grupo, en equipo, o de forma cooperativa para desarrollar software. Algunos trabajos en este sentido pueden ser vistos en (Keil *et al.*, 1995, Humphrey, 2000, Robillard *et al.*, 2000, Seaki, 1995). Una segunda línea de investigación está orientada hacia la construcción de modelos basados en un enfoque cooperativo para desarrollar software; algunos trabajos pueden ser vistos en (Boehm *et al.*, 1994, Bandinelli *et al.*, 1996, Lonchamp, 1994, Saeki *et al.*, 2000, Suzuki *et al.*, 1999, Dewan *et al.*, 1993). Finalmente, una tercera línea de investiga-

1 angelica@fi.upm.es  
Depto. de Lenguajes y Sistemas Informáticos e Ingeniería de Software, Facultad de Informática, Universidad Politécnica de Madrid Madrid - España  
Directora del Laboratorio "Decoroso Crespo" para la aplicación de las Tecnologías de la Información y Comunicaciones en la Educación.

2 marco@gordini.ls.fi.upm.es  
Depto. de Computación e Informática,  
Facultad de Ingeniería, Universidad de Tarapacá  
Arica - Chile  
Doctorando de la Universidad de Tarapacá Chile en la Facultad de Informática, Universidad Politécnica de Madrid España.

ción está dirigida hacia la construcción de entornos y herramientas Groupware (Ellis *et al.*, 1991, Grudin, 1994, Mendes *et al.*, 1997) para dar soporte al desarrollo o diseño de software. Algunos entornos y herramientas de este tipo pueden ser vistos en (De Farias *et al.*, 2000, Bandinelli *et al.*, 1996, Lonchamp, 1994, Saeki *et al.*, 2000, Suzuki *et al.*, 1999, Dewan *et al.*, 1993, Kaplan *et al.*, 1992, Mashayekhi *et al.*, 1993).

En la actualidad se está llevando a cabo un proyecto de investigación dentro del contexto del desarrollo de software educativo, cuyo objetivo es definir un modelo basado en un enfoque cooperativo para el desarrollo de aplicaciones educativas. Además, como parte del objetivo del proyecto, se pretende construir un entorno de desarrollo basado en dicho modelo. Así, el objetivo del proyecto está relacionado con la segunda y tercera línea de investigación mencionadas anteriormente. Esto porque se pretende construir, por un lado, un modelo basado en un enfoque cooperativo para desarrollar software, específicamente SwE, y por otro, un entorno groupware para el desarrollo de SwE que soporte tal modelo.

Para lograr el objetivo de desarrollar un entorno groupware (un sistema CSCW (Ellis *et al.*, 1991, Grudin, 1994)) para dar soporte al desarrollo o diseño de software, existen tres enfoques alternativos (De Farias *et al.*, 2000): desarrollo *ad hoc*, uso de toolkits (equipo de herramientas) de CSCW, y uso de entornos CSCW integrados. El desarrollo *ad hoc* de sistemas se lleva a cabo desde cero, usando poco o nada de elementos elaborados. El uso de toolkits tales como Prospero (Dourish, 1998) y Groupkit (Roseman *et al.*, 1996) apoya la construcción de sistemas sobre la base de un conjunto de bloques predefinidos que pueden ser reusados y combinados de muchas formas. Los entornos CSCW integrados, tal como el sistema Locale Support System Worlds (Fitzpatrick *et al.*, 1995), son plataformas cooperativas que proveen un conjunto de abstracciones y facilidades para el desarrollo de aplicaciones cooperativas.

Sin embargo, dado que las alternativas dos y tres imponen un modelo cooperativo propio que quedará subyacente en el entorno a ser generado, y que el modelo que aquí se propone es específico y está orientado hacia el desarrollo cooperativo de software educativo, nosotros hemos optado por la alternativa uno. Así, en este trabajo se propone primero generar un modelo de desarrollo cooperativo propio, para luego, a partir de este modelo, proponer la construcción *ad hoc* de un entorno que soporte dicho modelo.

De esta manera, las etapas planteadas para llevar a cabo este proyecto fueron las siguientes: (1) Generación de un modelo genérico de desarrollo cooperativo, (2) Especialización del modelo genérico hacia el desarrollo de SwE, (3) Construcción de un entorno de desarrollo que soporte el modelo especializado, y (4) Evaluación del modelo y su entorno asociado. En la actualidad, la primera etapa ha sido concluida, de manera que en el resto de este artículo se describe en forma resumida esta primera etapa del proyecto, así como los resultados obtenidos hasta el momento.

Para lograr la primera etapa del proyecto, primero fue necesario elegir un marco teórico (teoría o modelo) que constituyera un *framework* para capturar los aspectos más importantes concernientes con las actividades cooperativas, y que además, este marco teórico fuera una guía en el desarrollo del modelo, enfocándonos rápidamente en los problemas más relevantes y ayudándonos a estructurar el modelo de una forma coherente. Se observó en la literatura que ya han sido propuestos varios marcos teóricos para modelar aplicaciones cooperativas, y de esta manera se decidió analizar estos marcos teóricos para obtener la información necesaria para ser usada en el desarrollo del modelo propuesto. Posteriormente, y tomando como base dicha información, se generó el modelo considerando algunos conceptos claves de los marcos teóricos analizados. Este modelo se basa en el concepto denominado *Lugar de Actividad* el cual es definido como un espacio donde ocurren y se registran las actividades de un proceso de desarrollo de software en el tiempo.

El resto de este documento está organizado como sigue: La sección 2 describe los marcos teóricos que han sido usados para modelar aplicaciones CSCW. La sección 3 presenta el desarrollo del modelo genérico propuesto. Finalmente en la sección 4 se presentan las conclusiones alcanzadas y algunas ideas para el trabajo futuro.

## 2. MARCOS TEÓRICOS USADOS PARA MODELAR APLICACIONES CSCW

Esta sección presenta un resumen del análisis de varias propuestas de marcos teóricos que han sido usados para modelar aplicaciones CSCW, basado en los aportes encontrados en la literatura relacionada. Hemos analizado 3 propuestas diferentes: la Teoría de la Coordinación (Malone *et al.*, 1990), la Teoría de la Actividad (kuuti, 1991), y la Teoría de la Acción (Strauss, 1993).

## 2.1 Teoría de la Coordinación

La Teoría de Coordinación fue desarrollada por Thomas Malone (Malone *et al.*, 1990) y su equipo en el Centro para la Ciencia de la Coordinación, MIT, Boston, en los inicios de los años noventa. Observando la coexistencia de muchos estudios sobre coordinación en varios dominios, ellos sugieren un nuevo enfoque de coordinación, combinando e integrando disciplinas tan diversas como la informática, la psicología, y la economía.

Malone *et al.* definen *Teoría de la Coordinación* como un cuerpo de principios que describen cómo pueden ser coordinadas las actividades, esto es, describen cómo los actores pueden trabajar juntos armoniosamente.

Los autores identificaron cuatro componentes básicos de coordinación: objetivos, actividades, actores e interdependencias.

## 2.2 Teoría de la Actividad

El origen de este concepto puede encontrarse en los primeros escritos de Lev Vygotsky (1896-1934), quien sugiere que, socialmente, el significado de actividad puede servir como un principio explicativo con respecto a –y es considerado como un generador de– la conciencia humana. Esta teoría fue desarrollada por Leont'ev, un discípulo de Vygotski.

Esta teoría usa el concepto de actividad como la unidad básica de análisis (kuuti, 1991). Las actividades consisten en acciones o cadenas de acciones, que a su vez consisten en operaciones, que son rutinas bien definidas y usadas como respuestas a condiciones que se dan cuando se ejecuta una acción. La primera condición para cualquier actividad es la presencia de una necesidad. La necesidad puede ser suficiente para despertar y estimular las actividades, pero es incapaz de dirigir su particular orientación. De hecho, una actividad está orientada por la objetivación de la necesidad en un objeto. Las actividades son realizadas como acciones individuales o cooperativas, y las cadenas de tales acciones se relacionan unas a otras por el mismo objetivo global. La distinción puede hacerse de una actividad a otra según su objetivo.

Las acciones son componentes fundamentales de las actividades. Ellas están subordinadas a objetivos específicos. El objetivo de una acción es una representación mental consciente del resultado a lograr. Su función es la orientación de la acción. Pueden emprenderse diferentes acciones para lograr el mismo objetivo. Enton-

ces, las actividades se comprenden como acciones orientadas por objetivos.

Las operaciones son formas de ejecución de las acciones. Ellas se corresponden con la forma de lograr el objetivo y están directamente determinadas por las condiciones objetivas sobre las cuales tiene que ser logrado tal objetivo. Las operaciones pueden volverse rutinarias y ejecutarse en forma inconsciente con la práctica.

## 2.3 Teoría de la Acción

Según Fitzpatrick *et al.* (1995) la Teoría de la Acción de Strauss (1993) es un buen candidato como framework para fundamentar un entendimiento del concepto de trabajo por las siguientes razones: (1) este framework maneja un conjunto de abstracciones coherentes y relacionadas, lo cual hace posible que el aspecto social del concepto trabajo (cómo las personas trabajan en forma cooperativa) sea más accesible, y provee un medio para que otros conceptos de CSCW puedan ser relacionados o vistos como complementarios; y (2) provee medios analíticos para los desarrolladores de sistemas, quienes no tienen experiencia o conocimientos de las ciencias sociales o que no cuentan con los servicios de un profesional en el equipo. Los aspectos de la teoría de Strauss que son enfatizados en el trabajo de Fitzpatrick *et al.* son las nociones de acción e interacción, y la noción de mundo social.

Los conceptos de acción e interacción son los principales conceptos de esta teoría. Las acciones están implícitas en las interacciones. Las acciones e interacciones son llevadas a cabo por uno o más interactuantes. Un interactuante puede ser un individuo, una agregación o una colección de individuos. Cada interactuante lleva una particular identidad, biografía y perspectiva de una interacción. Las acciones de los interactuantes pueden ajustar y dirigir el curso de una interacción. El curso de una interacción está compuesto por secuencias de acciones conectadas que pueden ser estáticas o dinámicas.

Las acciones ocurren en un contexto de condiciones estructurales. Las condiciones estructurales pueden restringir o facilitar el curso de una interacción. Así, las condiciones estructurales y las acciones/interacciones se ajustan y ayudan a desarrollarse entre sí. Los cursos de una interacción pueden ser descompuestos en secuencias de acciones conectadas. Cuando hay varios participantes en el curso de una interacción, se requiere una articulación o alineamiento de sus respectivas acciones conectadas.

Una trayectoria es el concepto central para explorar el concepto de trabajo como una interacción. Una trayectoria es el curso de una acción sobre el tiempo. Las acciones e interacciones contribuyen a la evolución de una trayectoria.

La evolución de un curso de interacciones y de una trayectoria es ajustada y llevada a cabo por los interactuantes vía dos tipos de procesos: procesos interaccionales o estrategias –tales como la negociación, persuasión, educación o manipulación– y las acciones o procesos de trabajo, tales como la división del trabajo, supervisión, y ejecución de tareas.

Así, las dos formas principales sobre las cuales Strauss explora y entiende las acciones e interacciones son las estructuras y procesos, es decir, en términos de las condiciones que influyen y son influenciadas por las acciones, y en términos de la respuesta a cambios en estas condiciones sobre el tiempo. Este dualismo existe sólo a nivel teórico, la estructura y el proceso se unen en el momento de la acción.

La segunda noción importante en este modelo es la idea de mundos sociales. Un mundo social es una unidad interactiva que surge cuando un grupo de individuos actúa en forma colectiva, a menudo requiriendo la coordinación de perspectivas separadas y el uso de recursos compartidos. Un mundo social tiene actividades, lugares donde las actividades ocurren, y tecnologías como medios para llevar a cabo las actividades. Un mundo social puede también tener submundos e individuos que pueden ser miembros de varios mundos sociales simultáneamente. El trabajo emprendido por los miembros del mundo social puede ser descrito usando las nociones de acción/interacción.

## 2.4 Resumen

Del análisis de los marcos teóricos que han sido usados en el diseño de aplicaciones CSCW, se puede observar que cuatro conceptos se presentan en las tres teorías (sin considerar las diferencias de nombre). Estos conceptos son: actividad, actor, recurso y herramienta. La tabla 1 resume la terminología empleada por los diferentes enfoques.

Marco Teórico	Actividad	Actor	Recurso	Herramienta
Teoría de la coordinación	Actividad	Actor	Recurso	—
Teoría de la Actividad	Actividad/Acción/Operación	Participante, sujeto	Objeto	Herramienta
Teoría de la Acción	Actividad/Acción	Miembro/Interactuante	—	Tecnología

Tabla 1: Resumen de los términos usados por las teorías presentadas previamente.

- Una actividad representa un procedimiento cooperativo. Esta puede usualmente ser descompuesta en partes pequeñas llamadas sub-actividades o acciones y puede usualmente estar relacionada con otras actividades.
- Un actor, también llamado participante, sujeto, miembro o interactuante, representa una entidad responsable de la ejecución de una actividad. Un actor puede ser un individuo o un grupo.
- Un recurso u objeto representa lo que usa, produce o transforma una actividad. Un recurso puede estar disponible en forma electrónica o en alguna otra forma.
- Una herramienta o tecnología puede ser algo usado para llevar a cabo las actividades. Estas pueden ser herramientas para modelar o herramientas de apoyo.

Las características de cada concepto difieren de una teoría a otra. Por ejemplo:

- Una actividad puede ser descompuesta en pequeñas unidades de acuerdo a la Teoría de la Actividad y la Teoría de la Acción. Sin embargo, una actividad es vista como un concepto atómico en la Teoría de la Coordinación.
- No se hace distinción entre los actores en la Teoría de la Coordinación y la Teoría de la Acción. Sin embargo, en la Teoría de la Actividad pueden haber diferentes clases de actores asociados con una sola actividad.
- Sólo en la Teoría de la Actividad se considera a un actor como parte de una actividad. En las otras teorías un actor está asociado a una actividad con la intención de ejecutar tal actividad.

- Un recurso es compartido por las actividades de acuerdo a la Teoría de la Coordinación, mientras que en la Teoría de la Actividad un recurso es parte de una actividad.
- Un recurso es usado por un actor en la Teoría de la Actividad, mientras que en la Teoría de la Coordinación no se considera esta relación.
- La comunicación entre los diferentes actores involucrados en una actividad no es representada explícitamente en ninguna teoría.

### 3. MODELO PARA EL DESARROLLO COOPERATIVO DE SOFTWARE

En esta sección se describe el modelo propuesto. Este modelo se basa en un concepto denominado *Lugar de Actividad* el cual es definido como un espacio donde ocurren y se registran las actividades de un proceso de desarrollo de software en el tiempo. Iniciaremos esta descripción con las definiciones básicas necesarias para comprender el modelo y su proceso de generación.

#### 3.1 Desarrollo Clásico y Cooperativo de Software

Un *Proceso de Desarrollo de Software* (PDSw) es el proceso que se sigue en el tiempo para construir, entregar y hacer evolucionar un producto de software, desde la concepción de una idea hasta el fin de la vida útil del producto o del sistema en el cual opera. En general, el objetivo de un PDSw bien definido es hacer que el desarrollo sea confiable, predecible y eficiente, permitiendo ventajas tales como la automatización y el uso de componentes y (sub) procesos estándares.

La forma de manejar la complejidad subyacente en un PDSw es a través de abstracciones que llamamos *Modelos*. Así, un Modelo que describe un PDSw es una representación simplificada y práctica de dicho proceso que permite predecir su comportamiento bajo condiciones especificadas. Existen varios Modelos que describen un PDSw (por ejemplo, Cascada, Prototipos, Espiral). De ahora en adelante se les denominará Modelos para Desarrollar Software (MDSw). Todos estos modelos están basados en el reconocimiento de dos hechos:

- El software, como cualquier otro producto industrial manufacturado, tiene un ciclo de vida que se extiende desde su concepción hasta el fin de su vida útil; y
- Este ciclo de vida debe ser anticipado y controlado para lograr un producto con las características deseadas.

Por tanto, los objetivos de los MDSw son dos:

- Determinar el orden en que deberían llevarse a cabo las etapas y las diversas actividades técnicas involucradas en el desarrollo y evolución del software, esto es, responder a la pregunta: ¿qué debemos hacer a continuación?; y
- establecer un marco de referencia para administrar el desarrollo y mantenimiento del producto que nos permita estimar recursos, definir hitos intermedios y criterios de transición para avanzar de una etapa a la siguiente, y controlar el avance, esto es, responder a la pregunta: ¿hasta cuándo debemos realizar la siguiente actividad?

En torno a estos objetivos básicos, se han propuesto varios MDSw. Cada modelo se describe mediante un conjunto de fases (etapas y actividades) genéricas que puntualizan el desarrollo y la evolución del software, y un tipo de estilo de desarrollo que representa la forma en la cual se conducirán las fases de desarrollo (secuencial, iterativo, o recursivo) (Pressman, 1997). Esta es la forma clásica de describir un MDSw. A su vez, esta forma clásica de describir un MDSw ha dado origen a una forma clásica de visualizar un PDSw, es decir, a partir de las necesidades o requisitos y de los recursos existentes, se lleva a cabo un proceso de desarrollo y mantenimiento guiado por una descripción de dicho proceso, obteniéndose un producto software.

La forma clásica de describir un MDSw también ha dado origen a una forma clásica de diseñar y orientar el soporte informático para el PDSw. Es decir, dicho soporte generalmente se ha orientado hacia el proceso de desarrollo mismo, dando origen, por ejemplo, a las técnicas CASE (Computer Aided Software Engineering) que pretenden ser un apoyo informático a la labor de desarrollo de software (Lewis, 1991).

Una desventaja de este enfoque clásico es el énfasis que da a la descripción del PDSw, dejando implícita la intervención del factor humano, que es una característica inherente al desarrollo de software. Dicho de otra manera, el desarrollo es llevado a cabo por equipos humanos multidisciplinarios (Keil *et al.*, 1995, Humphrey, 2000) y cooperativos (Robillard *et al.*, 2000), por lo que la descripción de un PDSw no ocurre por sí misma. Así, el enfoque clásico deja implícito o no considera: (1) la intervención de un equipo humano y los problemas complejos de comunicación y coordinación que esto implica, (2) información sobre la interacción de los miembros del equipo, y (3) la forma de gestionar la información que requiere cada miembro sobre lo que se están

realizando los demás miembros y cómo lo están realizando.

Por lo anterior, fue necesario un cambio fundamental en la forma de describir los modelos de desarrollo de software. Este cambio está basado en las actividades de interacción<sup>3</sup> de las personas que realizan el proceso de desarrollo y no sólo en la descripción de lo que un equipo humano debe hacer. Este nuevo enfoque es el Desarrollo Cooperativo de Software (DCSw) que, en contraste con la forma clásica de describir, visualizar y soportar un PDSw, trata explícitamente y durante todo el proceso de desarrollo, la intervención de un equipo humano y toda la problemática asociada a ello (Mendes et al., 1997, Seaki, 1995).

Este nuevo enfoque ha dado origen a una nueva forma de soporte informático orientado hacia la interacción humana, dando lugar a software que soporta la comunicación, coordinación, registro de la intervención del equipo humano e información del producto y gestión de la información que requiere cada miembro sobre lo que están realizando los demás miembros del equipo y cómo lo están realizando (percepción). Este tipo de software se conoce con el nombre genérico de entorno o herramienta Groupware para el desarrollo o diseño de software (Ellis et al., 1991, Grudin, 1994).

Como fue comentado anteriormente, el objetivo de este proyecto es definir, por un lado, un modelo basado en un enfoque cooperativo para desarrollar software, específicamente SwE, y por otro lado, construir un entorno groupware para el desarrollo de SwE que soporte tal modelo. Se ha comentado que un modelo de desarrollo de software debería estar basado en las actividades de interacción de las personas que intervienen en el proceso de desarrollo. Así, la estrategia que se ha empleado para obtener el modelo cooperativo se basa en: (1) la consideración de un marco teórico fundamentado en las discusiones realizadas en la sección anterior; (2) un análisis de los posibles lugares donde ocurren y se registran las actividades de un PDSw en el tiempo, denominados *Lugares de Actividad*, y (3) una generalización de estos lugares de actividad.

3 Entenderemos por interacción el proceso social por el cual los individuos se relacionan a través del establecimiento de expectativas recíprocas y significativas, con el objetivo de trabajar o actuar juntos.

### 3.2 Derivación del modelo

Un análisis de los MDSw clásicos nos permite visualizar el conjunto de etapas que serán llevadas a cabo por el equipo de desarrollo. Estas etapas ayudarán a su vez a determinar los diferentes lugares de actividad.

#### Definiciones básicas previas

- Definiremos *Lugar de Actividad* como un *espacio* donde ocurren y se registran *actividades* de un PDSw en el *tiempo*.
- Definiremos *Actividad* como las *acciones* realizadas por un *equipo multidisciplinario* que *comunicado, coordinado*, y asistido por una o más *herramientas* lleva a cabo una o más *etapas* de un PDSw, las cuales están compuestas por una o más *tareas*.
- Definiremos *Tarea* como una o varias rutinas bien definidas.

Para dar una definición de *equipo multidisciplinario* nos basaremos en la propuesta de Dyer (1984). Según Dyer, un *equipo* consiste en:

- (a) al menos 2 personas, quienes
- (b) trabajan con una misión/objetivo/meta común, en donde
- (c) cada individuo tiene asignado un rol específico, y tal que
- (d) el cumplimiento de la misión requiere alguna forma de dependencia entre los miembros del grupo.

- Así, definiremos *Equipo Multidisciplinario* como un equipo donde los roles a ser asignados a los individuos que lo componen se definen a partir de las múltiples disciplinas que son necesarias para el desarrollo de una aplicación específica. Por ejemplo, para una aplicación educativa, las disciplinas necesarias son el diseño instructivo, la informática, el arte...

Las características mencionadas como *coordinado, comunicado*, y asistido por *herramientas* significan respectivamente que: el equipo multidisciplinario actúa coordinando sus acciones; actúa intercambiando información asociada con las acciones y sus efectos entre sus miembros; y usa una o más herramientas para ejecutar las acciones, incluyendo herramientas para modelar y herramientas de apoyo.

- En un lugar de actividad las actividades pueden ser realizadas como acciones individuales o en grupo (aunque visto de forma global sigue siendo trabajo en equipo). Así mismo, las acciones en grupo pueden requerir de acciones individuales. De esta manera, se

distinguen dos tipos de lugares de actividad: *Lugar de actividad de grupo*, y *Lugar de actividad individual*.

- Finalmente, en un lugar de actividad no sólo ocurren actividades de un PDSw, sino que, además, se registran tales actividades en el tiempo. Es decir, las actividades/acciones y sus efectos (productos/subproductos) son registrados en un repositorio común denominado *Memoria de Actividad*.

Sobre la base de estas definiciones se puede mostrar en un diagrama las relaciones asociadas al concepto de

Lugar de Actividad, como se muestra en la figura 1. En el diagrama se usa un rectángulo de la forma "□" para denotar que un elemento es estático, en el sentido de que está predeterminado por su definición, y se usa un rectángulo de la forma "◻" para denotar que un elemento es dinámico, en el sentido de que éste queda determinado por las actividades/acciones realizadas por el equipo de desarrollo a lo largo del tiempo. Además, como se observa en la figura 1, para almacenar la memoria de actividad se utiliza un repositorio.

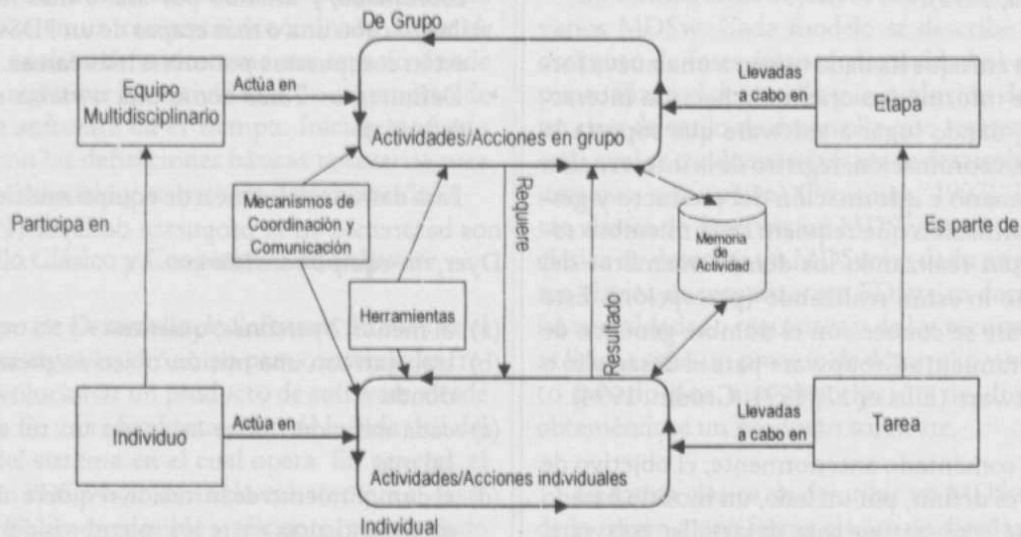


FIGURA 1: Relaciones asociadas al concepto de Lugar de Actividad.

### Etapas de un MDSw que determinan Lugares de Actividad

Un MDSw clásico, visto de forma genérica (Pressman, 1989), posee fases tales como Definición, Desarrollo y Mantenimiento. En la fase de Definición, se pretende establecer el *¿Qué?*, e incluye como subfases el Análisis del sistema, Planificación, y Análisis de requisitos. En la fase Desarrollo se pretende establecer el *¿Cómo?* Y sus respectivas subfases son: Diseño, Codificación, y Pruebas. Finalmente, en la fase de Mantenimiento se pretende tratar los eventuales *¿Cambios?* y las eventuales subfases son: Corrección, Adaptación, y Mejora. A su vez cada subfase o etapa está descrita por una serie de pasos bien definidos.

Dependiendo del estilo de desarrollo que se utilice para conducir las fases (secuencial, iterativo, o recursivo), se definen los MDSw clásicos específicos tales como cascada, cascada iterativo, o por prototipos. Un análisis de estos MDSw clásicos nos permite visualizar un conjunto de lugares de actividad que se establecen en el tiempo.

En primer lugar, aquellas propuestas que se basan en un estilo de desarrollo de tipo secuencial<sup>4</sup> presentan un PDSw basado en descripciones de fases (DF) que son independientes entre sí. En este estilo se determinan dos

4 Una vez que se ha completado un paso, no se retorna nunca a él, o a un paso previo (Eje. Cascada).

tipos de lugares de actividad. El primero, determinado por las actividades del PDSw asociadas a las etapas descritas por las DF. El segundo, determinado por las actividades asociadas a la realización de una transición ha-

cia un nuevo proceso descrito por la siguiente DF. La figura 2a presenta una descripción genérica de las propuestas basadas en este tipo de estilo.

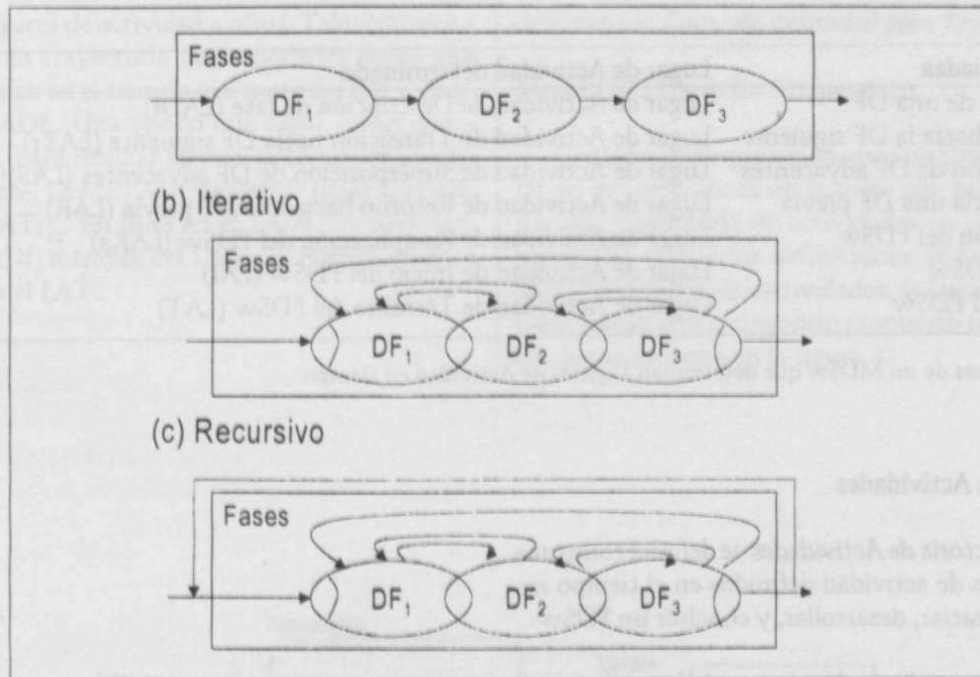


FIGURA 2: Descripción genérica de las propuestas basadas en un estilo de desarrollo secuencial, iterativo, y recursivo (Pressman, 1989)

Las propuestas que reportan un estilo de desarrollo de tipo iterativo<sup>5</sup> presentan procesos basados en DF que no son independientes, y que, por tanto, no sólo determinan los lugares de actividad del enfoque secuencial, sino además otros dos tipos de lugares. Primero, el lugar determinado por las actividades del PDSw asociadas a la superposición<sup>6</sup> entre dos procesos definidos por DF adyacentes. Segundo, el lugar determinado por las acti-

vidades asociadas a la realización de un retorno hacia un proceso descrito por una DF previa. La figura 2b presenta una descripción genérica de las propuestas basadas en este tipo de estilo.

En las propuestas que reportan un estilo de desarrollo de tipo Recursivo<sup>7</sup>, además de presentarse los mismos lugares de actividad determinados para el enfoque iterativo, se determina un tipo de lugar de actividad adicional que surge de las actividades asociadas a la realización de una re-aplicación del proceso de desarrollo. La figura 2c presenta una descripción genérica de las propuestas basadas en este tipo de estilo.

5 Si hay razones, se puede retornar a un paso completado previamente para introducir un cambio, y de esta manera propagar los efectos del cambio hacia atrás en el ciclo de desarrollo (Eje. Cascada iterativo).

6 Naturaleza interactiva de cada fase del modelo, ya que, por ejemplo, un determinado problema podría requerir en su fase de definición algo de diseño, o un producto intermedio, ya sea de la fase de definición o de Diseño, podría requerir mantenimiento por efecto de cambios en los requisitos o el descubrimiento de errores.

7 El enfoque completo puede ser re-aplicado al producto final obtenido (Eje. Espiral, Prototipos).



La tabla 2 presenta un resumen de las etapas de un MDSw que determinan los diferentes tipos de lugares de actividad en el tiempo. En esta tabla se han incluido dos lugares de actividad que están siempre presentes:

Lugar de Actividad de Inicio y Lugar de Actividad de Término. Estos dos lugares están relacionados con las actividades asociadas a la realización de una etapa de inicio y término del proceso de desarrollo, respectivamente.

Etapas asociadas	Lugar de Actividad determinado
Realización de una DF	Lugar de Actividad de Descripción de fase (LADf)
Transición hacia la DF siguiente	Lugar de Actividad de Transición hacia DF siguiente (LATr)
Superposición de DF adyacentes	Lugar de Actividad de Superposición de DF adyacentes (LAS)
Retorno hacia una DF previa	Lugar de Actividad de Retorno hacia una DF previa (LAR)
Re-aplicación del PDSw	Lugar de Actividad de Re-aplicación del PDSw (LARA)
Inicio del PDSw	Lugar de Actividad de Inicio del PDSw (LAI)
Término del PDSw	Lugar de Actividad de Término del PDSw (LAT)

Tabla 2: Etapas de un MDSw que determinan Lugares de Actividad en tiempo.

### Trayectoria de Actividades

Una *Trayectoria de Actividades* se definirá como una serie de lugares de actividad definidos en el tiempo necesarios para iniciar, desarrollar, y concluir un PDSw.

Tomando como base el concepto de lugar de actividad, en la figura 3 se muestra el diagrama de una posible Trayectoria de Actividades que podría haberse configurado en el tiempo.

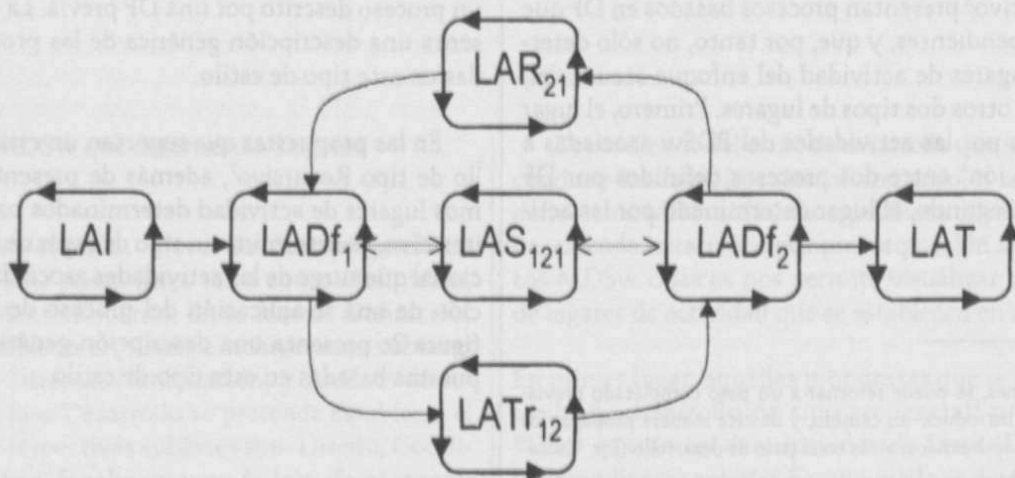


FIGURA 3: Una Trayectoria de Actividades que se ha configurado en el tiempo.

En el diagrama anterior, los subíndices permiten diferenciar entre lugares de actividad del mismo tipo.  $LATr_{12}$  indica el lugar de actividad de transición definido por la transición desde el lugar de actividad de descripción de la fase 1 a la fase 2. De esta manera se indica que en una trayectoria de actividades existen transiciones de unos lugares de actividad a otros. Tales transiciones definen una trayectoria de actividades como una entidad dinámica en el tiempo que parte del LAI y transita hacia el  $LADf_1$ . Una vez en este lugar, es necesario pasar al  $LAS_{121}$  para realizar algunas actividades asociadas a dicho lugar. Desde  $LADf_1$  se decide pasar al  $LADf_2$  a través del  $LATr_{12}$ . Estando en el  $LADf_2$ , es necesario retornar al  $LADf_1$  a través del  $LAR_{21}$ , y posteriormente transitar hacia el LAT.

### Lugar de Actividad para Transiciones

Para que una Trayectoria de Actividades sea posible, es necesario definir un nuevo lugar de actividad asociado a la decisión de qué transición debe llevarse a cabo en un determinado momento. Este nuevo lugar de actividad es denominado *Lugar de Actividad para Transiciones*.

### Modelo de Desarrollo Cooperativo

Considerando que un Modelo de Desarrollo Cooperativo de software es un modelo que permite determinar una trayectoria de actividades en el tiempo, y teniendo en cuenta las definiciones de Lugar de Actividad, Trayectoria de Actividades, y Lugar de Actividad para Transiciones, el modelo propuesto tiene la apariencia que se muestra en la figura 4.

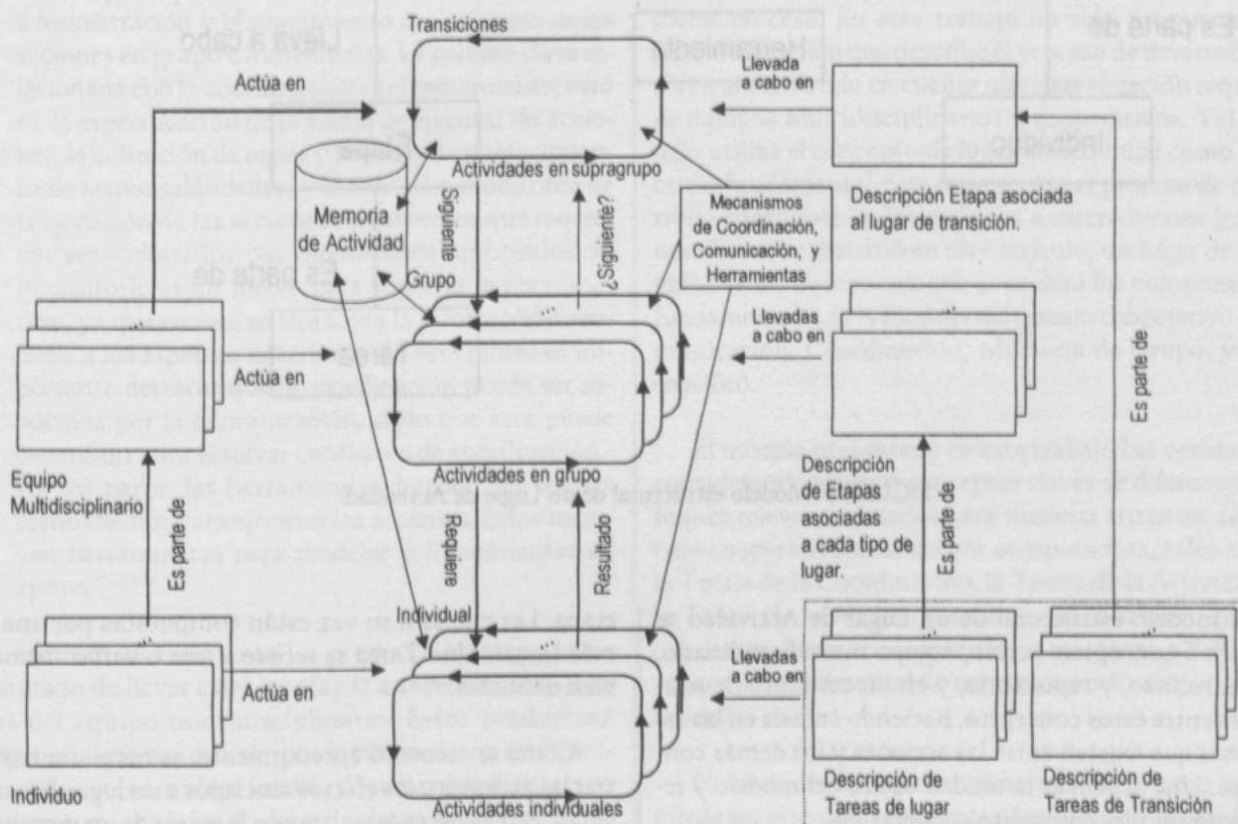


FIGURA 4: Un Modelo de Desarrollo Cooperativo de Software.

### 3.3 Modelo Estructural de un Lugar de Actividad

El modelo estructural asociado al concepto de lugar de actividad que estamos proponiendo es presentado en la figura 5. En este modelo se muestran las relaciones existentes entre los elementos básicos que participan en un lugar de actividad.



FIGURA 5: Modelo estructural de un Lugar de Actividad.

El modelo estructural de un Lugar de Actividad se basa en 5 conceptos: acción, equipo multidisciplinario, etapa, recurso, y repositorio; y en un conjunto de relaciones entre estos conceptos, haciendo énfasis en las relaciones que existen entre las acciones y los demás conceptos. Una acción es la unidad básica del modelo y representa un procedimiento cooperativo.

Una etapa describe el qué hacer en un PDSw, en contraste con las acciones, que son el cómo llevar a cabo esa

etapa. Las etapas a su vez están compuestas por una o más tareas. Una Tarea se refiere a una o varias rutinas bien definidas.

Como se comentó anteriormente, es necesario registrar las acciones y los efectos asociados a un lugar de actividad. Este registro se realiza con la ayuda de un repositorio (Memoria de Actividad) en el cual se almacenan tales acciones, así como los productos/subproductos creados/ usados/modificados por las mismas a lo largo del tiempo.

Un equipo multidisciplinario (compuesto por dos o más individuos) es una entidad a la cual pertenecen las acciones necesarias para llevar a cabo una o más etapas. El equipo puede comunicarse, coordinarse, y hacer uso de herramientas para ejecutar apropiadamente una acción. Cuando se piensa en comunicación, la palabra clave es el enlace.

- No se puede esperar colaboración entre los miembros del equipo si no existe un enlace entre ellos. Así, la comunicación entre los miembros de un grupo depende de la existencia de enlaces. Los enlaces pueden ser definidos como canales técnicos y/o canales de comunicación que ayudan a los miembros a intercambiar información (Keil *et al.*, 1995). Estos canales de comunicación incluyen, por ejemplo, mecanismos de intercambio de mensajes, y conferencias y foros de discusión electrónicos. Fuera de esta comunicación directa, es posible establecer canales de comunicación indirecta a través del Repositorio, donde la información almacenada en éste puede ser considerada como una potencial interfaz de comunicación (Ellis *et al.*, 1991).
- El trabajo cooperativo frecuentemente requiere de la administración y el seguimiento del progreso de las acciones en grupo e individuales. La palabra clave relacionada con la coordinación es el seguimiento, esto es, la especificación de la forma de ejecutar las acciones, la definición de reglas y límites, el establecimiento de responsabilidades, y el control y monitoreo de la ejecución de las acciones son aspectos que requieren ser soportados por mecanismos apropiados. El Repositorio es un medio para soportar la coordinación, ya que en éste se almacena la información asociada a los aspectos anteriores. En este punto es importante destacar que la coordinación puede ser soportada por la comunicación, dado que ésta puede contribuir para resolver conflictos de coordinación.
- Por su parte, las herramientas pueden ser usadas como medios para ejecutar las acciones. Estas incluyen herramientas para modelar y herramientas de apoyo.

Un recurso representa un producto o subproducto resultado de llevar a cabo las etapas a través de las acciones del equipo multidisciplinario. Estos productos/subproductos pueden ser por ejemplo datos electrónicos, tales como código fuente, documentos digitales, o registros en una base de datos; o datos no electrónicos, tales como los documentos no digitales, y son almacenados en el Repositorio. Estos productos/subproductos pueden ser producidos, consumidos, o modificados por

una o más acciones. Por ello se hace necesario gestionar el uso de recursos compartidos. Para gestionar el acceso simultáneo a un recurso o compartir un recurso existen dos alternativas: mecanismos de bloqueo del recurso mismo y mecanismos de control de concurrencia para monitorear el acceso al recurso. En torno a la segunda alternativa se pueden proveer mecanismos de gestión más elaborados tales como el control de concurrencia interactivo (Ellis *et al.*, 1991). En nuestro modelo optamos por el uso de mecanismos de control de concurrencia para compartir los recursos.

Por último, es posible, a partir de los mecanismos de comunicación y coordinación, y del repositorio, obtener la información de percepción que cada miembro del equipo requiera.

#### 4. CONCLUSIONES

El desarrollo de software tradicionalmente se ha llevado a cabo sobre la base de modelos que describen un proceso de desarrollo mediante un conjunto de fases, dejando implícita la intervención del factor humano en dicho proceso. En este trabajo ha sido propuesto un modelo genérico que describe el proceso de desarrollo de software teniendo en cuenta que su realización requiere de equipos multidisciplinarios y cooperativos. Tal modelo utiliza el concepto de lugar de actividad como concepto fundamental para representar el proceso de desarrollo basándose en las acciones e interacciones humanas. Como se discutió en este artículo, un lugar de actividad es un framework que considera los componentes fundamentales de la filosofía del trabajo cooperativo (Comunicación, Coordinación, Memoria de Grupo, y Percepción).

El modelo presentado en este trabajo fue construido considerando algunos conceptos claves de diferentes enfoques relevantes usados para modelar sistemas de trabajo cooperativo asistido por computadora, tales como la Teoría de la Coordinación, la Teoría de la Actividad, y la Teoría de la Acción, discutidas en la sección correspondiente. Estos conceptos claves ayudaron a definir el concepto fundamental de nuestro modelo: Lugar de Actividad.

Nuestro modelo tiene carácter genérico, y por tanto, puede ser orientado a distintas áreas de aplicación (educativas, financieras, etc.), llevando a cabo una especialización del modelo. Tal especialización estará determinada por el área específica de interés, ya que una vez

que el área queda definida, lo mismo ocurrirá con el equipo multidisciplinario y con las etapas/tareas asociadas al proceso de desarrollo de una aplicación. En este sentido, como trabajo futuro se llevará a cabo la especialización de nuestro modelo hacia aplicaciones educativas, el cual será soportado por un entorno groupware para diseñar este tipo de aplicaciones.

## REFERENCIAS

1. Ateyeh, K.; Millie, J.A.; Lockemann, P.C.: "Modular development of multimedia courseware", Proceedings of the First International Conference on Web Information Systems Engineering, vol. 2, pp. 179-187, 2000.
2. Bandinelli S.; Di Nitto E.; Fuggetta A.: "Supporting Cooperation in the SPADE-1 Environment, IEEE Transactions on Software Engineering, vol. 22, no. 12, December, pp. 841-865, 1996.
3. Boehm, B.; Bose, P.: "A collaborative spiral software process model based on Theory W", Proceedings Third International Conference on the Software Process, pp. 59-68, 1994.
4. De Farias, C.; Ferreira, L.; Van Sinderen, M.: "A Conceptual model for the development of CSCW systems", Designing Cooperative Systems. Use of Theories and Models. Proceedings of the 5th International Conference on the Design of Cooperative Systems (COOP'2000). IOS Press, Amsterdam, Netherlands, pp. 189-204, 2000.
5. Dewan P.; Reidl J.: "Toward Computer-Supported Concurrent Software Engineering", IEEE Computer, vol. 26, no. 1, pp. 17-27, 1993.
6. Dourish, P.: "Using Metalevel Techniques in a Flexible Toolkit for CSCW applications", ACM Transactions on Computer-Human Interaction, vol.5, no. 2, pp. 109-155, 1998.
7. Dyer J.: "Team Research and Team Training: a state-of-the-art review", Human Factors Review, The Human Factors Society, Inc., 1984, pp. 286, 309.
8. Ellis C.; Gibbs S.; Rein G.: "Groupware: Some Issues and Experiences", Communications of the ACM, vol. 34, no. 1, January 1991.
9. Fitzpatrick, G.; Tolone, W.; Kaplan S.: "Work, Locales and Distributed Social Worlds", Proceeding of the 1995 European Conference on Computer Supported Cooperative Work (ECSCW '95), pp. 1-16, 1995.
10. Grudin, J.: "Computer Supported Cooperative Work: History and Focus", Computer, May., vol. 27, no. 5, pp. 19-26, 1994.
11. Humphrey W.: "Introduction to the Team Software Process", SEI Series in Software Engineering, Addison Wesley, 2000.
12. Kaplan S.; Tolone W.; Carroll A.; Bogia D.; Bignoli C.: "Supporting collaborative software development with ConversationBuilder", Proceedings of the Fifth ACM SIGSOFT Symposium on Software development environments, December 9-11, Washington, DC United States, pp. 11-20, 1992.
13. Keil M.; Carmel E.: "Customer-developer link in software development" Communications of The ACM, vol. 8, no. 5, May, pp. 33-44, 1995.
14. Kuutti, K.: "The concept of activity as a basic unit of analysis for CSCW research", Proceedings of the second European Conference on Computer Supported Cooperative Work (ECSCW'91), pp. 249-264, 1991.
15. Lewis T.: "CASE. Computer-Aided Software Engineering", Van Nostrand Reinhold, 1991.
16. Lonchamp J.: "A Collaborative Process-Centered Environment Kernel", in Advances Information Systems Engineering (Wijers G.; Brinkkemper S.; Wasserman T. Editors), CAISE'94, Lectures Notes in Computer Science, no. 811, Springer-Verlag Pub., 1994.
17. Malone, T.; Crowston, K.: "What is Coordination Theory and how can it help design cooperative work Systems?", Proceedings of the ACM Conference on Computer Supported Cooperative Work (CSCW'90), pp. 357-370, 1990.
18. Mashayekhi V.; Drake J.; Tsai W.; Riedi J.: "Distributed, collaborative software inspection", IEEE Software, vol. 10, no. 5, September, pp. 66-75, 1993.
19. Mendes R.; De Sousa M.; Da Silva M.: "A Framework for the Classification of Computer Supported Collaborative Design Approaches", Proceedings of CRIWG' 97, Third CYTED-RITOS International Workshop on Groupware, San Lorenzo de El Escorial, Madrid, España, pp. 91-100, 1997.
20. Méndez G.; Antonio de A.; Herrero P.: "PRVIR: An Integration Between an Intelligent Tutoring System and a Virtual Environment", World Multiconference on Systemics, Cybernetics and Informatics", SCI, Orlando, Florida, USA, vol. VIII, pp. 175-180, 2001.
21. Pressman, R.: "Ingeniería del Software un Enfoque Práctico", 4ª edición, McGraw Hill, 1997.
22. Pressman, R.: "Ingeniería del Software un Enfoque Práctico", 2ª edición, McGraw Hill, 1989.
23. Qu C.; Gamper J.; Nejdil W.: "A Collaborative Courseware Generating System Based on WebDAV, XML, and JSP", Proceedings International Conference on Advances Learning Technologies, pp. 197-198, 2001.
24. Robillard P.; Robillard M.: "Types of collaborative work in software engineering", The Journal of Systems and Software, vol. 53, no. 3, September, pp. 219-224, 2000.
25. Roseman, M.; Greenberg, S.: "Building Real Time Groupware with Groupkit, A Groupware Toolkit", ACM Transactions on Computer Human Interaction, vol. 3, no. 1, pp. 66-106, 1996.
26. Saeki M.: "Communication, collaboration and cooperation in software development-how should we support group work in software development?", Proceedings Software Engineering Conference, 1995, Asia Pacific, pp. 12-20, 1995.
27. Saeki M.; Sureerat S.; Tanaka A.: "Supporting Distributed Individual Task in Cooperative Specification Development", International Journal of Software Engineering and Knowledge Engineering, vol. 10, no. 3, pp. 319-344, 2000.
28. Squires D.; McDougall: "Cómo elegir y utilizar software educativo", Morata Ediciones S.L, Madrid, España, 1997.
29. Strauss, A.: "Continual Permutations of action", Akdine de Gruyter, New York, 1993.
30. Suzuki J.; Yamamoto Y.: "SoftDock: A Distributed Collaborative Platform for Model-Based Software Development", Proceedings. Tenth International Workshop on Database and Expert System APP, pp. 672-676, 1999.
31. Twidale M.; Rodden T.; Sommerville I.: "Developing a Tool to Support Collaborative Dialogues and Graphical Representation of Ideas" in Collaborative Dialogues Technologies in Distance Learning, Verdejo M., Cerri S. (Eds.), NATO ASI Series (Vol. F:133), Springer-Verlag, 1994.
32. Ulloa U., "Open System for Collaborative Authoring and Re-use", Design and Production of Multimedia and Simulation-based learning Material, T. de Jong and L. Sarti (Eds), Kluwer Academic Publishers, pp. 83-97, 1994.