Inteligencia Artificial. Revista Iberoamericana de Inteligencia Artificial
ISSN: 1137-3601
revista@aepia.org
Asociación Española para la Inteligencia Artificial
España

Ardaiz, Oscar; Díaz de Cerio, Luis; Freitag, Felix; Gallardo, Antonia; Marqués, Joan Manuel; Messeguer, Roc; Navarro, Leandro; Sanjeevan, K.

Sistemas distribuidos y CSCL

Inteligencia Artificial. Revista Iberoamericana de Inteligencia Artificial, vol. 8, núm. 24, 2004, pp. 13-20
Asociación Española para la Inteligencia Artificial
Valencia, España

ARTÍCULO

Sistemas Distribuidos y CSCL

Oscar Ardaiz, Luis Díaz de Cerio, Felix Freitag, Antonia Gallardo, Joan Manuel Marqués, Roc Messeguer, Leandro Navarro, K. Sanjeevan

Universitat Politècnica de Catalunya
{leandro, marques, oardaiz, felix} @ac.upc.es

Resumen

El diseño de sistemas que faciliten el trabajo o el aprendizaje cooperativo precisa que tanto los participantes humanos como máquinas se organicen de forma que se aprovechen las características, respeten las limitaciones, adapte y optimice el uso de las capacidades de humanos y máquinas trabajando juntos con estrecha interdependencia. Este artículo presenta las dificultades y requisitos para que un sistema distribuido proporcione soporte al aprendizaje o trabajo colaborativo in grupos potencialmente dispersos. Una revisión de arquitecturas propuestas recientemente como las redes de iguales, grid computacional, grid semántico ayudan a introducir la 'arquitectura dispersa', un modelo suficientemente flexible para integrar los requisitos y características de recientes arquitecturas que incluyen agentes de usuario ligeros y componentes que se ejecutan en recursos proporcionados por los miembros del grupo (sus propios PC) o obtenidos bajo demanda del grid computacional. Este modelo precisa de una capa de middleware que proporcione soporte y simplifiquen el desarrollo de aplicaciones colaborativas. Finalmente se presenta el proyecto de investigación en que este trabajo tiene lugar.

Palabras Clave: CSCW, CSCL, P2P, Grid, arquitectura dispersa, middleware.

1. Personas y máquinas

El mundo de las personas y máquinas que colaboran sumando sus habilidades, recursos y trabajo es complejo y de delicado equilibrio. Comprender y diseñar componentes para mejorar el funcionamiento de un sistema así requiere conocimientos y técnicas que vienen de varias disciplinas. También está reconocido ampliamente que el diseño y la introducción de groupware en un grupo requiere tanto de un conocimiento muy detallado de la tarea a asistir, como del impacto global en el sistema social.

El diseño de un sistema complejo en que personas colaboran con la ayuda de aplicaciones informáticas requiere que todos los agentes (tanto personas como aplicaciones) se organicen de forma que aprovechen las características, se respeten las limitaciones, se aprovechen hasta el mínimo detalle posible las capacidades tanto de las personas como de las máquinas. Las aplicaciones colaborativas pueden llevar a un grupo de personas a una situación tanto de extrema efectividad en su tarea, como de extrema incomodidad. Del mismo modo, un grupo de personas que colaboran intensamente pueden llevar a las máquinas y la red a una situación de extrema saturación e incapacidad de asistir, como es en el caso de colaboración utilizando medios de comunicación muy ricos en información.

Es decir, no es efectivo desarrollar aplicaciones que ignoren la forma en que las personas colaboran, o la forma en que los recursos funcionan. Ambos

componentes del sistema socio-técnico tienen limitaciones que no conviene ignorar si se quieren evitar desequilibrios.

Por tanto, es conveniente desarrollar aplicaciones de forma "apropiada" a: la forma de gobierno del grupo, la ubicación de los participantes, las características de sus dispositivos de interacción con las máquinas, de las características de la red, del precio y disponibilidad de los recursos computacionales existentes.

Del dominio de la cooperación para el desarrollo se adopta el término "tecnología apropiada" para referirse a la conveniencia de proponer y construir sistemas que se adapten a los recursos existentes [Fox96], evitando gastos innecesarios, cuidando el coste-beneficio considerando personas y máquinas. Consideramos que es conveniente construir sistemas que asistan, aprovechando la limitada capacidad de las máquinas sin esperar que la ley de Moore [Moore65] nos salve en el futuro.

Desde el punto de vista de la tecnología apropiada, el diseño de software que ofrece un marco de abstracciones puede llevar a abstraer excesivamente [Einstein77] algunos detalles importantes de la infraestructura que hagan que el sistema sea inapropiado en para la tarea en la muchas de situaciones. Esto ocurre cuando se ignoran aspectos generales como la distribución, la presencia de fallos, la escala del sistema u otros aspectos que se detallan a continuación. Creemos que deben proponerse soluciones que se adaptan a las características del complejo sistema distribuido que forman personas, computadores y redes, cuando se organizan para llevar a cabo una actividad colaborativa.

El resto del artículo se estructura así: la sección 2 presenta las dificultades y requisitos que ha de cumplir un sistema de soporte a la colaboración en grupos potencialmente dispersos. A continuación se analizan diversas arquitecturas que aportan soluciones parciales. En la sección 4 se describe la propuesta de arquitectura dispersa que incluye un middleware que facilita la construcción de aplicaciones colaborativas. La sección 5 describe el proyecto CRAC que es el proyecto de investigación en que se realiza esta actividad. Finalmente se presentan unas breves conclusiones.

2. Dificultades y requisitos

La colaboración o el aprendizaje con la ayuda de máquinas pueden resultar complicados en situaciones en que los participantes pertenecer a varias organizaciones o departamentos, en que los recursos están gestionados por distintas autoridades que imponen reglas y limitaciones pensadas para facilitar la gestión y el trabajo interno y el uso individual, con barreras hacia el exterior (firewall). Pero la lista no acaba aquí, pueden aparecer muchos otros inconvenientes que a continuación se clasifican en términos de requisitos que podría cumplir un sistema para poder evitarlos.

Escala Internet del sistema: que esté formado por varios componentes (distribuido) y que tanto los miembros como los componentes puedan estar en cualquier lugar (disperso).

Acceso universal y transparente: que los participantes puedan conectarse desde cualquier ordenador, y que la vista que tengan sea independiente del punto de conexión. Por ejemplo el web lo permite.

Descentralización: que ningún componente tenga asignada de forma exclusiva la tarea de coordinación, ni que sea el único que disponga de alguna información. La centralización lleva a soluciones simples pero con componentes críticos, que condicionan la autonomía de los participantes, el crecimiento y la supervivencia del sistema.

Auto-organización del sistema: que el conjunto (sistema) tenga la capacidad de funcionar automáticamente sin necesidad de intervención externa, que tenga la habilidad de reorganizar sus componentes de forma espontánea, que sea resistente a fallos, que permita el dinamismo (ej. mobilidad, desconexión).

Disponibilidad del grupo: capacidad de que el grupo pueda trabajar con algún componente averiado o no disponible. Los sistemas con un coordinador (centralizado) son especialmente vulnerables si el propio coordinador no se implementa con un componente tolerante a fallos. La replicación es una técnica que introduce la redundancia necesaria para ayuda a aumentar la disponibilidad y la calidad de servicio.

Autonomía individual: que cada miembro del grupo decide libremente qué acciones realizará, qué recursos aportará, cuando estará conectado o desconectado,

Autosuficiencia del grupo: que cada grupo pueda funcionar con recursos limitados, idealmente únicamente los aportados por los miembros del grupo, o que el grupo pueda conseguir exteriormente.

Permitir compartición: que varios componentes puedan utilizar la misma información del grupo (p.ej:

eventos, objetos, estructura) tanto en el acceso como en la representación.

Seguridad del grupo: garantizar la identidad y el acceso selectivo y limitado a la información compartida. (protección de la información, autenticación).

Disponibilidad de recursos: En un sistema abierto como Internet, que un grupo debería disponer de recursos adecuados para llevar a cabo una tarea colaborativa: tanto los recursos que aporten sus miembros (red de iguales), como recursos externos adicionales (grid). (Por ejemplo un grupo podría reunirse en casa de algún miembro, o bien utilizar un espacio público, alquilar un local.) Los recursos pueden obtenerse y utilizarse según la necesidad sin tener que disponer de ellos previamente. (En el símil eléctrico, utilizar los recursos de la red eléctrica frente a tener que comprar pilas).

3. Arquitecturas propuestas

Para asegurar que algunos, varios o todos estos requisitos se cumplen, hay que diseñar adecuadamente la arquitectura del sistema, definir cómo se estructura en componentes y qué mecanismos o algoritmos distribuidos rigen su funcionamiento y organización. A continuación se presentar algunas arquitecturas y mecanismos propuestos recientemente que cumplen con algunos de los requisitos anteriores.

3.1. Arquitectura de iguales

En una red de iguales, en inglés "peer-to-peer" o P2P, formadas únicamente por los PC individuales de cada participante. Todas las máquinas comparten entre sí sus recursos: su capacidad de cálculo, almacenamiento y capacidad de comunicación, es decir actúan a la vez como clientes y proveedores de servicios. Las redes P2P son auto-suficientes y las máquinas se auto-organizan utilizando protocolos distribuidos para realizar búsquedas, y repartir la carga de transferir objetos de forma descentralizada. Debido a que no tiene funciones concentradas en uno o pocos puntos y todos los PC actúan de forma similar, las redes P2P tienen un alto grado de resistencia a fallos, desconexiones o ataques.

Sin embargo, las redes de iguales no crecen bien pues algunos protocolos consumen mucha capacidad de red y carga en los iguales al hacer búsquedas exhaustivas, y además no ofrecen ninguna garantía de la duración de una búsqueda o de la accesibilidad de la información. [Menascé03].

Las redes de aplicación, en inglés "overlay networks", son un tipo de redes de iguales que ofrecen un servicio de red de alto nivel, que conecta iguales entre sí. Utilizando funciones de hash distribuidas o a partir de medidas de la red, se organiza la topología de la red según el contenido de los iguales. La función de hash distribuida o el resultado de las medidas pueden ofrecer un reparto de carga, encaminar consultas, acotar el tiempo máximo para localizar un objeto en la red o reducir el tiempo de transferencia de un objeto. Este tipo de redes entre iguales mejoran bastante el comportamiento de las redes P2P pues hacen que el tiempo de búsqueda sea determinista, reducen la carga de la red y pueden mejorar la transferencia de objetos. [Doval03]

3.2. Arquitecturas grid

El grid computacional es una infraestructura para la computación distribuida pensada para aplicaciones que requiere elevada capacidad de proceso de datos en ciencia y tecnología avanzadas. El nombre se refiere a la metáfora de la red eléctrica (power grid), una compleja infraestructura distribuida de sencillo uso que proporciona potencia según la demanda. Un aparato eléctrico puede usar la cantidad de recursos que necesite con sólo conectarlo a un enchufe.

Por tanto el grid ofrece acceso según se requiera a capacidad de proceso de datos y funciones no disponibles para una o varias máquinas. Esto se consigue con un "middleware" que integra recursos computacionales repartidos geográficamente. La existencia de problemas que precisan una enorme capacidad de proceso de datos (p. ej. los experimentos de física de partículas en el CERN), la aparición de una arquitectura concreta (actualmente basada en Servicios Web) y la disponibilidad de una implementación abierta y estándar (Globus) han hecho del Grid un éxito en la comunidad de usuarios de computación avanzada.

La arquitectura actual basada en servicios Web, la "Open Grid Services Architecture" o OGSA, ofrece funciones para la integración y gestión, incluyendo la creación, gestión del ciclo de vida, introspección y agrupación de servicios, seguridad, registro, políticas, acceso e integración de datos, gestión de servicios y flujos (workflow).

AccessGrid [Stevens03] es una aplicación para que construir entornos de colaboración en forma de salas interconectadas para grupos de investigadores. Ofrece un espacio de trabajo multiusuario, semi-público que ofrece conexiones de audio y video con otros espacios de trabajo, en que los usuarios pueden moverse libremente e interactuar como si estuvieran todos en la misma sala. Esto se consigue utilizando

grandes pantallas de proyección (1,2x5 metros) y sonido bidireccional con micrófonos de ambiente.

Curiosamente AccessGrid combina conceptos previamente expuestos: es una red P2P "overlay" sobre la que circulan canales de audio y video por difusión (IP multicast), y actualmente AccessGrid 2.0 usa el Grid (Globus) para gestionar la seguridad en el transporte (SSL) y en la autenticación (PKI para certificados de identidad) de las sesiones multimedia y multipunto.

3.3. Arquitectura de la información

La estandarización de la estructura de la información permite que además de una persona, un programa pueda aprender lo suficiente sobre el significado de los datos como para procesarlos y organizar la información. Esto se conoce como el Web semántico. El futuro del web y de otras aplicaciones como las colaborativas debe ser rico en información sobre los objetos, como una base de datos global.

Los componentes arquitectónicos incluyen semántica (significado de los elementos), estructura (organización de los elementos), y sintaxis (comunicación). La estandarización en estos componentes en general y en el dominio del aprendizaje y la colaboración puede llevar a mejores formas de facilitar las actividades en grupo.

Algunos ejemplos de estándares en el dominio educativo son IEEE LOM o IMS para describir objetos de aprendizaje con meta-información; IEEE PAPI o IMS LIP para representar datos de estudiantes como su identificación, calificaciones, historia de actividad, competencias adquiridas, etc. Hay también estándares horizontales como AICC o SCORM para combinar contenido didáctico proveniente de diversas fuentes y asegurar la interoperabilidad entre varios sistemas.

Algunos ejemplos de estándares que proporciona el Web Semántico basados en XML: RDF para representar ontologías y anotar documentos web para asociarlos a esas ontologías, y extensiones como DAML, OML, OWL. También existen estándares para la descripción semántica de Servicios Web como DAML-S que proporciona un lenguaje para representar servicios más expresivo que UDDI o WSDL.

3.4. Arquitectura de redes programables

Un diferencia esencial del Web frente a la radio o TV está en que en la radio la potencia de emisión determina la zona de cobertura y se puede llegar a cualquier número de oyentes. En cambio, en el Web la zona de cobertura es global, pero lamentablemente los recursos para servir una página son proporcionales a la audiencia que suele ser un factor imprevisible [Ardaiz01].

Una infraestructura programable está formada por una gran cantidad de máquinas distribuidas por Internet, en las que se pueden instalar a la vez nuevos programas a través de un único interfaz. Eso permite la puesta en marcha o desplegamiento de nuevos servicios en una red de forma sencilla. Este es un modelo cercano al Grid.

Las redes programables permiten automatizar la instalación, puesta en marcha y mantenimiento de aplicaciones o servicios en varias máquinas de la red para adaptarse a la demanda y la dispersión de los usuarios [Ten97].

4. Arquitectura dispersa

Proponemos un modelo de sistema y una arquitectura que pueda incorporar la mayor parte de las ideas expresadas anteriormente. El sistema debe integrar personas que colaboran para realizar una tarea con la ayuda de máquinas repartidas e interconectadas por una red. En pocas palabras, agentes de usuario "ligeros" que accedan a entidades que utilizan recursos computacionales. Esta aproximación es próxima al concepto de Computación Ubicua [Weiser93]

En realidad, esta definición también corresponde a muchas aplicaciones existentes que utilizan un navegador como agente de usuario y un servidor estructurado en una o dos partes, muchas de ellas basadas en el paradigma del "modelo-vista-controlador" [MVC 80]. El navegador mediante páginas web o pequeños programas (applets) proporcionan vista y controlador. El servidor proporciona el modelo en forma de un programa que procesa las peticiones y actúa sobre un modelo que se almacena de forma persistente en una base de datos. Este modelo también puede aplicarse a los sistemas llamados "two-tier" y "three-tier".

Una aplicación que se ejecuta en cualquier máquina de una red y se presenta en un terminal gráfico, utilizando protocolos de presentación gráfica en red como X, ICA o RDP sigue también este esquema.

Se ha estudiado VNC [Rich98] una herramienta para presentación gráfica en clientes ligeros que utiliza el protocolo "Remote FrameBuffer" o RFB que permite visualizar a distancia el interfaz de usuario de cualquier aplicación o de la pantalla completa, desde cualquier lugar en Internet y desde una gran variedad de tipos de máquinas, por ser el protocolo simple, requiere pocos recursos de red y máquina y es

independiente de las características de la máquina donde se visualiza y actúa.

Este sistema y protocolo se ha extendido [Navarro03] para permitir la visualización por parte de un grupo de alumnos en un aula informática en red y se han desarrollado algunas aplicaciones de soporte a la colaboración entre un grupo de alumnos o con el profesor.

4.1.1. Niveles de interacción colaborativa

Para clasificar el tipo de soporte a la colaboración hemos definido varios niveles o superficies de colaboración:

- la pantalla: compartir entre varias personas el contenido de una pantalla completa. Por ejemplo, varios miembros del grupo pueden compartir una única superficie (pantalla) en la que pueden trabajar individualmente o bien colaborar utilizando varios cursores, compartiendo un único cursor y entrada gracias a un mecanismo de control de turnos, o realizando anotaciones sobre la pantalla.

- la ventana (en realidad el interfaz de una aplicación): compartir entre varias personas el interfaz de una aplicación. Por ejemplo una aplicación de navegación web en grupo, en que un miembro del grupo puede dirigir una visita guiada por el web haciendo que el navegador de cada miembro del grupo obedezca a sus órdenes, es decir visite las páginas que selecciona el guía, mientras los demás miembros del grupo pueden leer la página cada uno a su ritmo.

- la aplicación: utilizar una aplicación multi-usuario que pueda interactuar de forma diferenciada con cada usuario y pueda presentarle una vista distinta de la información que comparten los participantes. Un ejemplo puede ser un editor compartido, en que un grupo de personas editan un mismo documento, pero cada uno puede estar viendo y modificando a la vez partes distintas del documento.

Este modelo con agentes de usuario sencillos permite que las personas puedan tener la comodidad de utilizar dispositivos de interacción tan diversos como computadores de sobremesa conectados a una red Ethernet o PDA conectados a la red sin cables. También es conveniente y viable permitir la mobilidad: tanto que el interfaz pueda acompañar a las personas que lo usan, migrando de un dispositivo a otro, o funcionado aunque la persona y el dispositivo vaya cambiando de ubicación: todas las personas y algunos computadores pueden cambiar con frecuencia de ubicación con todos los cambios técnicos y organizativos que eso conlleva. (requisito de acceso universal y transparente)

No obstante, además del interfaz, es necesario disponer de máquinas o recursos computacionales donde residan el resto de componentes del sistema.

4.1.2. Recursos computacionales

Tradicionalmente, los recursos computacionales los aportan (se compran e instalan) previamente las personas u organizaciones de acuerdo a las necesidades de las aplicaciones que vayan a utilizarse.

Dependiendo del mecanismo de funcionamiento de las aplicaciones colaborativas puede ser necesario disponer de un recurso "servidor" que lo proporciona una organización. Esto puede suponer un problema para el grupo porque el servidor puede estar gestionado con una política que no se adapte a lo que el grupo necesita o bien cuando las personas pertenecen a varias organizaciones.

De acuerdo con los requisitos de descentralización, auto-organización y auto-suficiencia expresados anteriormente, proponemos usar formas de organización/gobierno que no requieran componentes centralizados utilizando reglas de funcionamiento que permitan el funcionamiento autónomo, tal como funcionan las redes P2P.

Ocurre en ocasiones que los grupos que colaboran no disponen de los recursos propios necesarios. Para que el grupo sea autosuficiente, debería existir un mecanismo para que el grupo pueda conseguir recursos externos. Este mecanismo lo proporciona el Grid, que debe permitir obtener/contratar recursos computacionales cuando sean necesarios para ejecutar componentes, desplegar servicios, almacenar o publicar información.

La necesidad de reclamar recursos externos adicionales puede surgir de la necesidad de mayor capacidad de cálculo (p.ej. entornos interactivos para grupos numerosos), capacidad de almacenamiento, calidad de servicio (p.ej. en comunicaciones multimedia), para desplegar servicios que se ofrecen a personas fuera del grupo: donde la cantidad de recursos es sensible a la demanda, o al uso, (p.ej. en la redacción de un periódico que ha de servirse a una población grande de lectores).

Más que necesidad de gran cantidad de recursos, se trata de seleccionar los recursos necesarios o adecuados para realizar una tarea según las características del grupo y la actividad a realizar.

Por ejemplo un grupo de personas deciden redactar un documento: a medida que se incorporan personas, han de localizar la sesión de redacción, incorporarse a ella, obtener el documento y los cambios que se van produciendo, mantener la persistencia del documento, proveer servicio de difusión para los lectores interesados en borradores o versiones del documento que se está produciendo, etc.

Desarrollar aplicaciones en este entorno puede resultar demasiado complejo si no se proporciona un marco de servicios o una infraestructura que proporcione servicios y transparencias selectivas en forma de un middleware para la colaboración o el aprendizaje colaborativo.

4.1.3. Necesidad de una infraestructura

Para poder construir aplicaciones en esta arquitectura dispersa será necesario orquestar una serie de servicios, protocolos, componentes arquitectónicos que puedan servir para facilitar la construcción de aplicaciones.

Un paso en esta dirección es la propuesta de LaCOLLA [Marquès03], una arquitectura autónoma y auto-organizada que ofrece un "middleware" que cumple en gran medida con los requisitos expuestos, inspirada por las arquitecturas comentadas en la sección 3, para facilitar la construcción de aplicaciones colaborativas para grupos potencialmente dispersos.

LaCOLLA ofrece a las aplicaciones tres abstracciones: grupos, diseminación de información sobre la actividad, y almacenamiento de objetos generados en el grupo. Estas abstracciones se proporcionan de manera autónoma, distribuida, descentralizada, replicada, autoorganizada y utilizando los propios recursos de los miembros del grupo (auto-suficiencia). Para conseguirlo se ha optado por una arquitectura de iguales (P2P). De esta manera se garantiza la máxima flexibilidad, disponibilidad de la información, dinamismo y tolerancia a fallos. LaCOLLA también facilita que cualquier miembro de un grupo pueda saber inmediatamente qué está ocurriendo y tener acceso a los objetos que se van produciendo.

La arquitectura se ha organizado en tres tipos de componentes que actúan de forma autónoma. Cada igual puede instanciar uno, dos o los tres componentes siguientes:

- Agentes de Usuario: representan a los miembros del grupo en LaCOLLA,
- Agentes de Almacenamiento: almacenan de forma persistente objetos y eventos generados en el grupo.
- Agentes de Administración de Grupos y Presencia, que se encargan de aspectos relacionados con la administración de la información de los grupos y sus miembros.

Estos componentes interactúan autónomamente entre sí. Los mecanismos que utilizan para coordinarse se han agrupado en las categorías siguientes: de objetos, de eventos, de presencia, de ubicación y de mensajería instantánea.

Esta arquitectura y las propiedades más importantes como la capacidad de auto-organización, ha sido validada por simulación, para varios grados de actividad, varios tamaños del grupo, en presencia de fallos y desconexiones. [Marquès03]

5. Una propuesta de trabajo

Este es un trabajo en curso, que tiene lugar en el marco del proyecto CRAC [CRAC02], un proyecto coordinado entre la UVA, UPC y UOC financiado por el MCYT de tres años de duración.

El objetivo general es el estudio y desarrollo de un modelo de entorno de aprendizaje cooperativo que funcione en redes globales sirviéndose de "middleware" basado en modelos de mallas de computación ("grid") y sistemas autónomos descentralizados como las redes "peer-to-peer" o "p2p". Se trata por tanto de estudiar tres aspectos fundamentales:

- Entorno de Aprendizaje Cooperativo (A.C.): definir las características esenciales que ha de incorporar una aplicación informática de soporte al aprendizaje universitario para dar soporte a actividades de A.C.
- Estructura (Middleware): definir los componentes de una infraestructura de computación distribuida, basada en los modelos de redes grid y "peer-to-peer", que proporcione servicios de acceso a recursos, seguridad, organización y coordinación de procesos computacionales, para que una grupo de personas pueda participar en una actividad de A.C.
- Mecanismos: Planificación, optimización y asignación de recursos en un sistema distribuido de componentes autónomos y descentralizados basado en el middleware anterior, utilizando algoritmos basados en sistemas económicos, redes neuronales y sistemas neuro-difusos.

El desarrollo y validación de estos modelos y mecanismos que forman ecosistemas computacionales complejos, interdependientes y parcialmente específicos a una aplicación, requiere la participación coordinada de expertos en varios temas que este proyecto reúne.

La validación de cada objetivo y del conjunto precisa:

- realizar simulaciones para estudiar el funcionamiento del modelo de "middleware" propuesto junto con los mecanismos de asignación de recursos y coordinación,

- construir un prototipo o introducir innovaciones en sistemas existentes para validar las características esenciales de una aplicación de aprendizaje cooperativo,

- realizar experiencias con usuarios reales (estudiantes) y analizar sus interacciones para refinar y comprobar que el modelo es correcto.

Objetivos concretos:

A. Análisis de las interacciones colaborativas.
Análisis de las interacciones colaborativas en un espacio virtual de trabajo compartido durante la elaboración de un proyecto informático, a través de parámetros estadísticos para poder comprender y sistematizar los procesos y efectos cognitivos, sociales y motivacionales que resultan de estas interacciones. Puede aplicarse a estudiar la problemática del trabajo para determinar un tamaño adecuado del grupo, volumen de trabajo adicional de los procesos colaborativos, número adecuado de grupos por asignatura, organización eficiente del espacio compartido que ofrecen las aplicaciones informáticas, definir criterios para que el profesor o consultor pueda hacer un mejor seguimiento, orientación y evaluación del trabajo en grupo.

B. Definición de las características y componentes esenciales de una aplicación de soporte al aprendizaje cooperativo.
A partir del análisis de interacciones y del marco DELFOS [UVA], se trata de proponer un modelo de componentes que incorporen las características esenciales de una aplicación informática para dar soporte al aprendizaje cooperativo. A la vez se propondrá cómo incorporar éste soporte en aplicaciones informáticas existentes.

C. Definición de una infraestructura "middleware" basada en los modelos de "grid" y "p2 p".
Para este objetivo se estudian los modelos de redes que permiten desplegar aplicaciones distribuidas, para seleccionar y particularizar los componentes necesarios de una infraestructura que permita formar "organizaciones virtuales" de personas que usan recursos que participan en actividades de aprendizaje cooperativo. Esta infraestructura formará parte y extenderá o especializará redes de computación global basadas en modelos "grid" y "p2p".

D. Estudio y evaluación de mecanismos para la asignación de recursos y coordinación.
En los sistemas de computación global, con infinidad de componentes autónomos en red, los mecanismos para asignar recursos y coordinar las acciones forman parte fundamental del sistema. Se estudia la problemática de asignación de recursos y coordinación de la actividad para optimizar y mantener el equilibrio del ecosistema que se forma, aplicando algoritmos económicos, redes neuronales y sistemas neuro-difusos. Para ello se desarrolla un simulador que modele nuestra infraestructura y aplicaciones cooperativas sobre un sistema en red con usuarios, recursos y servicios. Con él, se realizarán simulaciones y medidas con los diferentes algoritmos a partir de cargas reales o sintéticas, lo que permitirá comparar y verificar su validez en términos de criterios como el uso de recursos, estabilidad, eficiencia.

E. Desarrollo y experimentación con aplicaciones concretas a partir del sistema propuesto.
Es necesario desarrollar prototipos o extensiones de sistemas existentes para poder refinar y validar la investigación. Estos desarrollos van a permitir realizar experimentos con estudiantes, tanto de nivel universitario como no universitario. Con el apoyo de desarrolladores, organizaciones y centros de formación como UVA (BSCW), UPC+upcnet S.L. ("Campus Digital"), UOC+GECSA ("Campus Virtual"), Edebé-Espais Telemàtics S.L. ("Red didáctica" y "Bitaula"), Rededia S.L. ("edebenet") se desarrollarán prototipos de algunos de los componentes propuestos y se realizarán experimentos de validación con usuarios reales. Estas experiencias permitirán a las empresas colaboradoras incorporar resultados del proyecto en sus productos de aprendizaje a distancia.

6. Conclusiones

El diseño de sistemas que incluyen personas y máquinas que interactúan entre sí de forma intensa requiere poner atención y respetar aspectos sutiles pero esenciales tanto del funcionamiento del grupo humano como de la tecnología.

La tecnología ofrece a la vez limitaciones y oportunidades que si se tienen en cuenta pueden aprovecharse para permitir nuevas oportunidades de

colaboración entre personas que pueden estar dispersas respetando la autonomía del grupo.

La propuesta de arquitectura dispersa cumple la mayoría de requisitos identificados para poder construir aplicaciones colaborativas distribuidas, con componentes dispersos, clientes ligeros y recursos ampliables según demanda. En este entorno complejo resulta muy útil disponer de un "middleware" como LaCOLLA para simplificar la construcción de estas aplicaciones.

El proyecto CRAC ofrece la oportunidad de seguir trabajando en refinar, probar y validar estas ideas.

Agradecemos al grupo de investigación formado por los participantes del proyecto COSACO (TIC2000-1054) y CRAC (TIC2002-04258-C03-01) y a los amigos que participan en los talleres de trabajo, siempre abiertos a nuevas ideas y opiniones.

7. Referencias

Ardaiz, O. *Application Network Deployment in the Internet*, Tesis doctoral (en revisión), 2003.

Ardaiz, O.; Freitag, F.; Navarro, L. *Estimating the Service Time of Web Clients using Server Logs*, (SIGCOMM-LA 2001). Computer Communications Review, Volume 31, number 2, ISSN 0146-4833, April, 2001.

Berners-Lee, T. *Semantic Web Road map*, http://www.w3.org/DesignIssues/Semantic.html, 1998.

Berners-Lee, T.; Hendler, J.; Lassila, O. *The Semantic Web*, Scientific American, May 2001.

Burbeck, S. *Applications Programming in Smalltalk-80: How to use the Model-View-Controller (MVC)*, 1992.

CRAC, Proyecto MCYT, 2002-2005, http://research.ac.upc.es/crac/

Doval, D. *Overlay Networks*, IEEE Internet Computing, v7 #4, 2003.

Einstein, A. frase *Everything should be made as simple as possible, but not simpler*, Reader's Digest, 1977.

Fox, A.;Brewer, E.; Gribble, S.; Amir, E. *Adapting to Network and Client Variability via On-Demand Dynamic Transcoding*. ASPLOS, 1996. http://citeseer.nj.nec.com/fox96adapting.html

Marquès, J. M. *LaCOLLA: una infraestructura autònoma i auto-organitzada per facilitar la col·laboració*, Tesis doctoral, 2003.

Menascé, D. *Scalable P2P Search*, IEEE Internet Computing, v7 #2, 2003.

Moore, Gordon E. *Cramming more components onto integrated circuits*, Electronics, Volume 38, Number 8, April 19, 1965.

Navarro, L. *Canal de Aprendizaje*, Informe interno, 2003.

Richardson, T.; Stafford-Fraser, Q.; Wood, K.R.; Hopper, A. *Virtual Network Computing*, IEEE Internet Computing, Volume 2, Number 1, January/February 1998.

Stevens, R.; Papka, M. E.; Disz, T. *Prototyping the workspaces of the future*, IEEE Internet Computing, v7 #4, Julio 2003.

Tennehouse, D. et al. *A Survey of Active Network Research*, IEEE Communications Magazine, 1997.

Weiser, M. *Some Computer Science Problems in Ubiquitous Computing*, Communications of the ACM, July 1993.