

Universidad de Deusto (tercer ciclo)

Programa: *Ciencias de la Computación e Inteligencia Artificial*

**Sistema de Trabajo Cooperativo
basado en la transmisión de eventos**

Tesis doctoral presentada por don **Jesús M^a Herrero Arranz**

Dirigida por el Dr. **Javier Oliver**

El Director

El Doctorando

Bilbao, 1 de Julio de 1998

Resumen

El Trabajo Cooperativo Soportado por Ordenador aborda cómo los sistemas informáticos pueden ayudar a realizar las tareas que son llevadas a cabo por un grupo de personas. Dentro de este ámbito se incluyen los Sistemas de Conferencia de Documentos que hacen uso de aplicaciones existentes y permiten a varios usuarios trabajar de forma simultánea desde diferentes lugares con las mismas aplicaciones que utilizan de forma individual.

Estos Sistemas de Conferencia de Documentos han sido tradicionalmente desarrollados siguiendo un modelo de arquitectura centralizada, en el que las aplicaciones se ejecutan en un solo nodo. En esta tesis se ha desarrollado un sistema, denominado *DocuLAN*, siguiendo un modelo escasamente utilizado basado en una arquitectura replicada en la que se lanzan las aplicaciones cooperativas en cada nodo y la cooperación se sustenta en la transmisión de los eventos de entrada. Este modelo replicado genera un menor tráfico de red y unos tiempos de respuesta más bajos que el modelo centralizado. En cambio, presenta la dificultad, que ha sido abordada satisfactoriamente en *DocuLAN*, del mantenimiento de la consistencia entre todos los nodos participantes.

Asimismo, el sistema se ha desarrollado sobre el entorno *MS-Windows* lo que facilita en gran medida la cooperación ya que se utiliza un entorno de bajo coste, ampliamente utilizado y que cuenta con un gran volumen de aplicaciones disponibles.

Agradecimientos

Quiero mostrar mi agradecimiento a mi director de tesis Javier Oliver por su paciencia y aliento durante el tiempo de elaboración de esta tesis.

Asimismo agradezco a mi empresa ROBOTIKER por los medios que ha puesto a mi disposición para realizar el presente trabajo.

Por último, quiero agradecer a mi familia y a mis amigos y amigas por su comprensión y ánimo durante este tiempo.

ÍNDICE

Resumen.....	ii
Agradecimientos.....	iii
ÍNDICE.....	iv
Índice de Tablas.....	viii
Índice de Figuras.....	x
1. Introducción.....	1
2. Trabajo Cooperativo.....	6
2.1 Introducción al Trabajo Cooperativo.....	6
2.2 Los Grupos.....	10
2.2.1 Actividades de los grupos.....	12
2.2.2 Papeles y normas.....	13
2.2.3 Ciclo de vida de un grupo.....	14
2.2.4 Ventajas e inconvenientes.....	15
2.2.5 Tareas grupales.....	16
2.3 La coordinación y la comunicación.....	19
2.4 Algunas conclusiones.....	22
3. CSCW y Groupware.....	24
3.1 CSCW.....	24
3.2 Groupware.....	28
3.3 Relación entre CSCW y Groupware.....	30
3.4 El éxito de las aplicaciones Groupware.....	32
3.5 Clasificación de los sistemas Groupware.....	38

3.5.1 El espectro Groupware	39
3.5.2 Tabla espacio-temporal.....	40
3.5.3 Clasificación Funcional	42
3.6 Sistemas de Conferencia de Documentos en Tiempo Real.....	45
3.7 El estándar T.120.....	49
4. Estudio de los Sistemas de Arquitectura Replicada	52
4.1 Introducción.....	52
4.2 Algunas definiciones	53
4.3 Descripción general.....	54
4.3.1 Gestión del testigo	56
4.3.2 Gestión del espacio cooperativo.....	58
4.3.3 Establecimiento de la sesión cooperativa	61
4.3.4 Anotaciones y señalización.....	61
4.4 Ventajas.....	62
4.5 Desventajas.....	65
4.6 Consciencia de la cooperación	72
4.7 Conclusiones	75
5. Descripción del Sistema <i>DocuLAN</i>.....	76
5.1 Introducción.....	76
5.2 Relación con el proyecto europeo MMTCA	77
5.3 Entorno de desarrollo y ejecución	78
5.4 Descripción general.....	80
5.5 Eventos y filtros.....	84
5.5.1 Eventos	84
5.5.2 Filtros.....	86
5.6 Módulo de Gestión de las Aplicaciones	90
5.6.1 Proceso de preparación de la sesión	91
5.6.2 Proceso de arranque de las aplicaciones cooperativas.....	93
5.6.3 Proceso de cooperación	95

5.6.4	Proceso de finalización de la sesión	97
5.7	Módulo de Gestión del Testigo	97
5.8	Módulo de Comunicaciones	99
5.9	Módulo de Gestión de Eventos	100
5.9.1	Mecanismo de eliminación de eventos no cooperativos	101
5.9.2	Mecanismo de pulsar y arrastrar	102
5.9.3	Mecanismo de pulsación doble	104
5.9.4	Mecanismo de control de situaciones indeterministas	105
5.9.5	Mecanismo de gestión de los recursos externos	107
5.9.6	Mecanismo de mejora del rendimiento	107
5.10	Limitaciones de <i>DocuLAN</i>	108
5.10.1	Recursos externos	108
5.10.2	Problemas concretos	109
6.	Evaluación del Sistema <i>DocuLAN</i>	111
6.1	Comparación con sistemas centralizados	111
6.1.1	<i>Farsite Document Conferencing</i>	111
6.1.2	<i>Intel ProShare Personal Conferencing Software</i>	113
6.1.3	Descripción del entorno de pruebas	113
6.2	Resultados de las pruebas	117
6.3	Valoración de los resultados	125
6.3.1	Tráfico de red	126
6.3.2	Tiempo de respuesta	135
7.	Discusión	138
7.1	Comparación con otros sistemas de arquitectura replicada	138
7.1.1	Sistema <i>Rapport</i>	139
7.1.2	Sistema <i>MERMAID</i>	147
7.1.3	Otros sistemas	150
7.2	Consistencia en la arquitectura replicada	151
7.3	Comparación con los sistemas de arquitectura centralizada	158

8. Conclusiones.....	161
8.1 Trabajo futuro.....	163
Referencias	166

Índice de Tablas

Tabla 1.1. Cuota de mercado de Sistemas Operativos a nivel mundial.....	3
Tabla 2.1. Ventajas e inconvenientes de los Procesos Grupales.....	16
Tabla 2.2. Cuadrantes y tipos de Tareas Grupales.....	18
Tabla 2.3. Componentes de la Coordinación.....	20
Tabla 3.1. Recomendaciones de la Serie T.120.....	50
Tabla 5.1. Eventos del entorno <i>Windows 3 v</i>	85
Tabla 5.2. Información del evento en la cola del sistema.....	85
Tabla 5.3. Descripción de los filtros.....	88
Tabla 5.4. Descripción de las <i>funciones filtro</i>	89
Tabla 6.1. Resultados de la Prueba 1.....	117
Tabla 6.2. Resultados de la Prueba 2.....	118
Tabla 6.3. Resultados de la Prueba 3.....	119
Tabla 6.4. Resultados de la Prueba 4.....	120
Tabla 6.5. Resultados de la Prueba 5.....	121
Tabla 6.6. Resultados de la Prueba 6.....	122

Tabla 6.7. Resultados de la Prueba 7.....	123
Tabla 6.8. Resultados de la Prueba 8.....	124
Tabla 6.9. Tráfico total en las 8 pruebas.....	125
Tabla 6.10. Resultados de la Prueba 1 con el tamaño medio de los paquetes.	126
Tabla 6.11. Acciones realizadas por cada Aplicación.	128
Tabla 6.12. Resultados globales de las pruebas.....	131
Tabla 7.1. Mensajes generados en <i>Rapport</i> sin mecanismo de empaquetamiento. ...	144
Tabla 7.2. Mensajes generados en <i>Rapport</i> con mecanismo de empaquetamiento. ..	145
Tabla 7.3. Tiempos de respuesta en segundos en <i>Rapport</i>	146
Tabla 7.4. Ratios de volumen de información en <i>FarSite</i> , <i>ProShare</i> y <i>DocuLAN</i> . ..	146

Índice de Figuras

Figura 2.1. Modelo de Tareas Grupales.....	17
Figura 3.1. Posible impacto de los Sistemas Groupware (valoración de 0 a 10).....	36
Figura 3.2. Disponibilidad de los Sistemas Groupware (valoración de 0 a 10).....	37
Figura 3.3. Espectro Groupware.....	40
Figura 3.4. Tabla espacio-temporal.....	41
Figura 3.5. Clasificación de los Sistemas de Conferencia en Tiempo Real.....	46
Figura 3.6. Sistema de Arquitectura Centralizada.....	47
Figura 3.7. Sistema de Arquitectura Replicada.....	48
Figura 4.1. Esquema de un Sistema Replicado.....	55
Figura 4.2. Secuencia de un evento en una Arquitectura Centralizada.....	64
Figura 4.3. Secuencia de un evento en una Arquitectura Replicada.....	64
Figura 5.1. Módulos del sistema <i>DocuLAN</i>	82
Figura 5.2. Pantalla de un usuario ejecutando <i>DocuLAN</i>	83
Figura 5.3. Llamada y encadenamiento de las <i>funciones de filtro</i>	88
Figura 5.4. Procesos del Módulo de Gestión de Aplicaciones.....	90

Figura 7.1. Versión <i>MULTI</i> del sistema <i>Rapport</i>	140
Figura 7.2. Versión <i>SINGLE</i> del sistema <i>Rapport</i>	141
Figura 7.3. Secuencia de un evento en la versión <i>MULTI</i>	142
Figura 7.4. Secuencia de un evento en la versión <i>SINGLE</i>	143
Figura 7.5. Esquema de control de las aplicaciones cooperativas en <i>MERMAID</i>	148

1. Introducción

Durante la década de los 80 los ordenadores personales se afianzaron como una potente herramienta ampliamente utilizada en las empresas. Durante ese tiempo, estos equipos se utilizaban fundamentalmente de forma individual [Rodden91b]. Pronto surgieron la tecnología de red y aplicaciones como el correo electrónico que permitían establecer una comunicación y una interacción entre los usuarios, pero su uso durante esa década fue muy limitado [Beaudouin92]. Existía, por tanto, una carencia importante en el soporte informático a las tareas que requerían la participación de varias personas.

A comienzos de la década de los 90 las empresas tuvieron que afrontar la creciente globalización del mercado mundial. Este hecho propició una organización de las empresas más distribuida en el espacio y con más puntos de contacto repartidos por todo el mundo [Reinhard94] [Watabe90].

Ante esta nueva realidad, se debe también superar el uso individualizado de los sistemas informáticos y afrontar el reto de crear sistemas que permitan dar soporte al trabajo cooperativo, a aquellas tareas que son llevadas a cabo por varias personas independientemente de su localización física. Estas personas necesitan reunirse, intercambiar información, consultar y discutir documentos, realizar informes, etc. Para poder aplicar la informática a estas tareas surge el Trabajo Cooperativo Soportado por Ordenador (*Computer Supported Cooperative Work, CSCW*) [Bannon91].

El CSCW es un amplio campo multidisciplinar que cubre todos los aspectos relacionados con el trabajo cooperativo y con el soporte informático que se le puede prestar. Una tarea fundamental es investigar el modo en que trabajan las personas cuando forman parte de un grupo. Para llevar a cabo esta labor se requiere la participación de expertos en las ciencias sociales. Al mismo tiempo, y teniendo siempre en cuenta los estudios anteriores, se deben desarrollar sistemas informáticos que faciliten las tareas en las que están involucradas dos o más personas.

En general, estos sistemas informáticos de soporte al trabajo cooperativo se denominan sistemas *groupware*. Algunos ejemplos característicos son el correo electrónico, los productos de videoconferencia, los sistemas de gestión de procesos (sistemas *workflow*) o las agendas electrónicas para grupos.

En este trabajo se van abordar los Sistemas de Conferencia de Documentos en tiempo real que hacen uso de aplicaciones existentes [Ellis91]. Estos sistemas permiten a varios usuarios trabajar de forma simultánea desde diferentes lugares físicos con las mismas aplicaciones que utilizan habitualmente de forma individual. Asimismo, previamente, se va a realizar una introducción de los conceptos fundamentales relacionados con el CSCW.

Los Sistemas de Conferencia de Documentos han sido tradicionalmente desarrollados siguiendo un modelo de arquitectura centralizada en el que únicamente se lanzan las aplicaciones cooperativas en un nodo y la cooperación se basa en la transmisión del estado de las ventanas de salida al resto de participantes. En esta tesis se ha abordado un método escasamente desarrollado hasta el momento que consiste en hacer uso de una arquitectura replicada en la cual las aplicaciones cooperativas se lanzan en cada uno de los nodos que participan en la conferencia. La cooperación se lleva a cabo por medio de la transmisión de los eventos de entrada desde el nodo que los origina al resto. Este modelo presenta importantes ventajas como su reducido tráfico de red y su menor tiempo de respuesta, pero cuenta con el importante inconveniente de la dificultad para mantener la consistencia entre todos los nodos participantes. Si las

aplicaciones cooperativas de cada nodo no mantienen el mismo estado, la cooperación se rompe [Lauwers90].

Hasta el momento las únicas experiencias en el desarrollo de sistemas replicados se han realizado en el entorno *UNIX* [Greenberg90]. Este entorno, como se muestra en la Tabla 1.1, cuenta con una escasa cuota de mercado y las previsiones futuras se mantienen en la misma línea. Los datos de la tabla provienen de la consultora *Dataquest* e incluyen nuevas adquisiciones y actualizaciones de los sistemas [Computing97].

Sistemas Operativos	1996	1998	2000
Windows 3.x, 9x	81%	55,2%	54%
Windows NT	9%	36%	38%
UNIX	0,4%	0,3%	0,3%
MacOS	6,1%	6,2%	6,3%
Otros	4,4%	2,6%	1,9%

Tabla 1.1. Cuota de mercado de Sistemas Operativos a nivel mundial.

Los Sistemas de Conferencia de Documentos aspiran a facilitar tareas grupales que realizan habitualmente las personas. Por tanto, resulta apropiado proporcionar sistemas de arquitectura replicada para aquellos entornos mayoritariamente utilizados y, de esta forma, superar la limitación del escaso volumen de usuarios a los que se llega con los sistemas actuales sobre *UNIX*. Además los Sistemas de Conferencia de Documentos con arquitectura replicada no requieren un alto grado de formación ya que posibilitan establecer una cooperación haciendo uso de las aplicaciones que se utilizan de forma individual. Es decir, los usuarios de aplicaciones individuales son usuarios potenciales de los Sistemas de Conferencia de Documentos con arquitectura replicada.

Asimismo, los sistemas *UNIX* ofrecen un coste claramente superior a los sistemas *Windows 3.x* o *9x* o incluso *Windows NT*. En un estudio realizado por *Deloitte & Touche* se afirma que *Windows NT* ofrece una reducción respecto a *UNIX* del 39% en el coste total de la propiedad (*total cost of ownership, TCO*). Los componentes del modelo TCO incluyen los costes iniciales de hardware y de software, los costes de mantenimiento y soporte y los costes de oportunidad [Computing98].

El objetivo del presente trabajo es la construcción y evaluación de un Sistema de Conferencia de Documentos en tiempo real bajo un modelo de arquitectura replicada que se ejecute en un entorno de uso masivo y de bajo coste. El entorno seleccionado para el desarrollo de este sistema ha sido *MS-Windows*, que cumple los requisitos mencionados [Petzold90]. De esta forma, los sistemas replicados acceden a un entorno muy extendido que además cuenta con un gran volumen de aplicaciones susceptibles de ser utilizadas en la cooperación.

La evaluación del sistema va a consistir de una medición del tráfico de red y del tiempo de respuesta y de una valoración global del grado de consistencia en la cooperación que se puede alcanzar. El objetivo último es determinar si se puede obtener un sistema lo suficientemente estable que pueda ser utilizado por los usuarios aprovechando las ventajas potenciales de este modelo.

En la primera parte de la tesis se presentan los conceptos relacionados con esta área de investigación como son el Trabajo Cooperativo, *CSCW* y *Groupware*. A continuación se introduce un estudio de los Sistemas de Conferencia de Documentos que siguen un modelo replicado, indicando sus ventajas (tráfico de red y tiempo de respuesta) y desventajas (mantenimiento de la consistencia) al ser comparados con los tradicionales sistemas centralizados.

En una segunda parte, se describe de forma detallada el sistema de arquitectura replicada propuesto, denominado *DocuLAN*, desarrollado sobre un entorno de bajo coste. Posteriormente se va a medir de forma cuantitativa el comportamiento de *DocuLAN* en lo referente al tráfico de red y al tiempo de respuesta. A continuación, se

va a realizar una evaluación global del sistema tomando como referencia los sistemas ya existentes, tanto los centralizados como los replicados. Por último se presentan las conclusiones obtenidas tras la realización de esta tesis, así como algunas posibles líneas de trabajo futuro.

2. Trabajo Cooperativo

2.1 Introducción al Trabajo Cooperativo

En cualquier organización es habitual que las tareas, y sobre todo las de mayor trascendencia, sean llevadas a cabo por un grupo de personas. En la actualidad nadie trabaja de forma completamente independiente y cada individuo forma parte, consciente o inconscientemente, de uno o varios grupos que varían a lo largo del tiempo y cuya composición responde a diversos criterios como la localización física, el proyecto en el que está involucrado, el departamento al que pertenece, etc. Entre las personas que componen estos grupos existe una comunicación, un intercambio de información y en muchos casos un objetivo común.

El término Trabajo Cooperativo tiene una larga trayectoria en las ciencias sociales. Fue utilizado por primera vez en la mitad del siglo XIX en el campo de la economía para referirse al trabajo en el que intervienen varios actores [Ure1835] [Wakefield1849]. Años después fue definido formalmente por Marx como:

varios individuos trabajando conjuntamente de una forma planificada en el mismo proceso de producción o en otro proceso relacionado [Marx1867].

En este siglo, el término ha sido usado en el mismo sentido por varios autores [Miller64] [Thompson67] y especialmente por la escuela alemana de la sociología del trabajo [Popitz57] [Bahrtdt58].

El campo que abarca el término Trabajo Cooperativo no está claramente definido. Para algunos autores todo trabajo es esencialmente cooperativo en el sentido de que siempre depende de otros agentes externos para su correcta realización [Ehn88] [Kling91]. Siguiendo este razonamiento, el adjetivo *Cooperativo* resulta redundante ya que no añade ninguna información nueva. Kling apunta que hasta en el caso extremo de un prisionero encadenado que es trasladado por su guardián existe algún tipo de cooperación entre ambos ya que en caso contrario el guardián tendría que estar continuamente arrastrando al prisionero para poderle conducir de un lugar a otro.

Por otra parte, en el otro extremo se sitúan quienes como Sørgaard imponen una serie de condiciones muy restrictivas para calificar un trabajo como cooperativo [Sørgaard87]. Estas condiciones que determinan el carácter cooperativo del trabajo, según Sørgaard, son las siguientes:

1. Las personas trabajan conjuntamente debido a la naturaleza de la tarea. Puede ser por pura necesidad, como en la manipulación de un tornillo y una tuerca separados por una barrera física, o para mejorar la calidad del producto final, como en la formación de un grupo para desarrollar un proyecto determinado.
2. Las personas involucradas deben compartir objetivos comunes y no competir entre ellos.
3. El trabajo es llevado a cabo bajo una organización horizontal, poco jerarquizada.
4. El trabajo desarrollado debe ser relativamente autónomo, ya que el control externo reduce la naturaleza cooperativa del trabajo.

El mismo Sørgaard reconoce que esta definición de Trabajo Cooperativo es muy estricta, propia de una situación ideal difícil de encontrar. Estas condiciones definen un tipo específico de trabajo y no su forma habitual. Sørgaard concluye que la

cooperación puede ser un aspecto más del trabajo que aparece de forma simultánea con otros aspectos, como la existencia de una jerarquía.

Sørgaard también realiza una interesante distinción entre Trabajo Cooperativo y Trabajo en Colaboración, en base a la definición de ambos términos. Cooperar significa *obrar juntamente con otro para algún fin*, mientras que colaborar es *trabajar con otra u otras personas en una misma tarea*. El término Trabajo Cooperativo introduce el concepto de un objetivo o fin compartido lo que le hace más restrictivo.

Otros autores se sitúan en un nivel intermedio entre la calificación de todo trabajo como cooperativo y la imposición de una serie de condiciones muy restrictivas, pero sigue predominando el carácter general y amplio del término. Así Bannon define el término Trabajo Cooperativo como:

una referencia general a varias personas trabajando conjuntamente para generar un producto o un servicio [Bannon91].

Este estilo de trabajo no implica una forma específica de interacción u organización, como una relación de compañerismo, identidad de grupo, igualdad de posición o formación, etc.

Partiendo de esta definición general, Bannon diferencia varios tipos de Trabajo Cooperativo. Unos se basan en una *comunicación o interacción indirecta*, por medio del cambio de estado de un proceso de transformación y otros se basan en una *comunicación directa*, en la que existe una comunicación interpersonal. Para Sørgaard únicamente estos últimos se pueden considerar como Trabajo Cooperativo.

Se toma como ejemplo una línea de montaje en la que cada trabajador realiza una parte del proceso. No se podrá hablar de Trabajo Cooperativo, tal y como lo concibe Sørgaard, por la naturaleza de la tarea y por no existir una comunicación directa entre los trabajadores. En cambio para Bannon, la línea de montaje entraría dentro del

concepto de Trabajo Cooperativo al estar elaborándose un producto con la participación de varias personas y existir un entorno de trabajo compartido.

A lo largo de esta tesis, se entenderá el Trabajo Cooperativo como:

trabajo realizado por un grupo de personas para alcanzar un objetivo compartido; entre estas personas existe una comunicación y una coordinación.

En esta definición se establecen una serie de requisitos para poder considerar un trabajo como cooperativo:

1. La existencia de un grupo o un colectivo de personas.
2. La existencia de un objetivo compartido.
3. La existencia de una comunicación y una coordinación entre las personas del grupo.

A pesar de introducir estas condiciones, se mantiene un concepto general y abierto del término Trabajo Cooperativo al considerar estos requisitos en su sentido más amplio. Así, el concepto grupo hace referencia a un conjunto de personas involucradas en una tarea. Incluso para algunos autores estas condiciones serían redundantes ya que los conceptos de objetivo compartido, coordinación y comunicación estarían incluidos en la definición de grupo. No obstante, se incluyen todas estas condiciones para cubrir aquellas situaciones en las que la noción de grupo es más débil entre los participantes, por ejemplo debido a la lejanía física, pero se cumplen el resto de características que definen al Trabajo Cooperativo.

Asimismo, se incluyen en esta definición aquellas situaciones en que la cooperación se puede dar de una forma distribuida tanto en el espacio como en el tiempo. Hay una cooperación distribuida en el espacio cuando los miembros del grupo no están en el mismo lugar físico y hay una cooperación distribuida en el tiempo cuando existe una comunicación asíncrona entre los miembros del mismo, por ejemplo a través

de fax o correo. También se incluyen aquellas situaciones en las cuales la comunicación se realiza de forma indirecta según el modelo de Bannon.

A continuación se analizan algunos de los elementos que forman parte de esta definición de Trabajo Cooperativo como son los grupos, la coordinación y la comunicación.

2.2 Los Grupos

El concepto de grupo también ha sido objeto de un amplio estudio por parte de las ciencias sociales y existen diferentes definiciones del mismo. Así Cole define el término grupo como:

dos o más personas que interactúan y comparten un propósito común [Cole92].

Por tanto, establecen como requisitos imprescindibles para poder hablar de grupo la existencia de un propósito común y una comunicación. Por ejemplo, un conjunto de personas esperando en una parada de autobús no forman un grupo, pero si empieza a llover y una persona abre su paraguas y lo ofrece a algunos más se establece un grupo al darse una comunicación (ofrecimiento del paraguas) y un propósito común (refugiarse de la lluvia).

En los informes *DATAPRO* se ofrece una definición similar, pero haciendo hincapié en la información. Así el grupo es definido como:

un conjunto de individuos que trabajan para resolver un problema común usando la misma información y recursos, independientemente de su localización [Datapro94].

En cambio Stodosky se centra en los aspectos relacionados con la comunicación y define el grupo como:

dos o más personas que interactúan con otras de forma que cada persona influye y se ve influenciada por otra persona [Stodolsky95].

Todas estas definiciones se adaptan al concepto propuesto de Trabajo Cooperativo, en el que se consideraba como uno de los requisitos la existencia de un grupo.

Bahrtd proponía en 1984 un definición también muy general, pero que no se ajusta al concepto propuesto de Trabajo Cooperativo. Bahrtd considera que:

existe un grupo cuando sus miembros se perciben a sí mismos como un nosotros [Bahrtd84].

Esta definición se identifica con la idea más común del término grupo en cualquier entorno. No obstante, el término Trabajo Cooperativo intenta abarcar algo más, al incluir también aquellas situaciones en la que las personas involucradas no se consideren a sí mismas como un grupo, pero cumplen el resto de requisitos como la existencia de un objetivo compartido o la existencia de algún tipo de comunicación. Además, esta comunicación se puede dar sin la necesidad de compartir el mismo lugar físico y por tanto sin la necesidad de que los diferentes participantes se conozcan, con lo que la consciencia de estar formando un grupo disminuye.

La definición de Bahrtd, como la mayoría de los estudios sobre los grupos, se ha realizado lógicamente sin tener en cuenta el soporte informático que se puede dar a ese trabajo en grupo. La consciencia de estar formando un grupo, de percibirse como un nosotros, es clara cuando se produce un contacto directo cara a cara entre los miembros, pero esa consciencia disminuye cuando los miembros no están en contacto o se produce alguna comunicación de forma indirecta. Lo importante es que el soporte informático

introduzca importantes ventajas para paliar esa posible pérdida de consciencia de estar formando un grupo.

Los grupos pueden ser clasificados siguiendo diversos criterios. Hay grupos formales o informales, centralizados o descentralizados, estructurados o no estructurados [Marca92]. Los grupos formales están claramente definidos y los componen quienes están involucrados en un mismo proyecto, bajo la dirección de una misma persona. Los grupos informales se rigen bajo otros criterios más abstractos como son la proximidad física o la realización de funciones similares en diferentes proyectos o departamentos.

El comportamiento de los grupos depende de múltiples y complejas variables que son fundamentalmente humanas, es decir, no totalmente predecibles. Por tanto los grupos no pueden ser descritos en base a fórmulas lineales de causa y efecto. Únicamente se podrán describir una serie de conceptos y comportamientos que aparecen [Cole92]. Esta característica esencialmente humana y no predecible es la principal dificultad para que los sistemas informáticos puedan proporcionar un soporte eficiente a las actividades grupales.

2.2.1 Actividades de los grupos

Dentro de estas actividades hay que distinguir entre el contenido y el proceso. El contenido se refiere a *lo que hace* el grupo, es decir, al trabajo en sí, los análisis, los desarrollos, las decisiones, etc. El proceso hace referencia a *cómo lo hace*, es decir, a la forma en que se desarrolla el trabajo, cómo se toman las decisiones, o el nivel de participación.

En el contenido hay que distinguir entre dos tipos de actividades: las tareas propias del grupo y las actividades de mantenimiento. Estas últimas corresponden, por ejemplo, al establecimiento de los objetivos, establecimiento de prioridades, coordinación de los esfuerzos individuales, evaluación de la actividad del grupo, etc.

Todas ellas resultan fundamentales para asegurar el éxito del grupo y su descuido puede ser la razón del fracaso de una actividad grupal. Estas actividades, por su carácter generalizable a todos los grupos, son las que pueden verse optimizadas con la ayuda de sistemas informáticos [Cole92].

2.2.2 Papeles y normas

Otro aspecto importante dentro de los grupos es el de los papeles que adoptan cada uno de los miembros. Estos papeles son actitudes o posiciones adquiridas para actuar durante un espacio de tiempo que no tienen relación con el cargo oficial de cada individuo [Cole92]. Además son variables e intercambiables a lo largo del ciclo de vida del grupo. Kantor distingue cuatro papeles entre los miembros de un grupo: activo, opositor, seguidor y espectador [Kantor75].

1. El papel activo corresponde a quien conduce la actividad del grupo proponiendo objetivos, centrando las cuestiones o haciendo sugerencias.
2. El opositor es el que continuamente plantea discrepancias a las acciones propuestas.
3. El papel del seguidor es el que refuerza las propuestas del que tiene un papel activo u opositor.
4. El espectador es quien observa sin participar en las actividades.

Asimismo, los grupos cuentan con una serie de normas, que son criterios, actitudes, puntos de vista que mantienen todos los miembros del grupo. Funcionan como un conjunto de estándares para el grupo. Estas normas se van elaborando a lo largo de la vida del grupo y se ven influenciadas por las propias actividades que se desarrollan.

Por último, el liderazgo es otro aspecto clave para el éxito de un grupo. El líder es el responsable de establecer los objetivos, hacer un seguimiento continuo de los

avances y tiene que dirigir las actividades propias del grupo y las actividades de mantenimiento. Debe tener en todo momento un papel activo.

2.2.3 Ciclo de vida de un grupo

En el ciclo de vida de los grupos se pueden distinguir cinco etapas: de formación, de lanzamiento, de establecimiento de las normas, de desarrollo y de finalización [Cole92].

1. En la etapa de formación, los miembros tratan de orientarse y habituarse a la nueva situación, clarificando los objetivos del grupo y conociendo al resto de componentes. En esta fase, el grupo depende en gran medida del líder.
2. En la etapa de lanzamiento, el grupo empieza a adquirir su propia identidad. Es una fase delicada porque se pueden encontrar discrepancias entre las personas o entre subgrupos establecidos sobre cuestiones como el liderazgo, el poder o el control interno.
3. Una vez superada esta etapa se puede hablar de un equipo real de trabajo y se entra en la fase de establecimiento de las normas. Estas normas deben posibilitar la coordinación y la integración de todos los miembros.
4. Posteriormente, en la etapa de desarrollo todos los esfuerzos individuales se centran en la consecución del objetivo grupal. Aquí el papel del líder no es tan activo y debe dejar que los miembros desarrollen sus propias capacidades.
5. Los grupos entran en la fase de finalización porque se han alcanzado los objetivos previstos o se ha producido una reestructuración en la organización que ha decidido la supresión de ese grupo. Esta fase debe ser dirigida activamente por el líder.

2.2.4 Ventajas e inconvenientes

El trabajo en grupo no es siempre mejor o más efectivo que el trabajo individual, pero algunas tareas, y normalmente las más complejas, únicamente pueden ser llevadas a cabo con la aportación de diferentes conocimientos individuales que se aglutinan para alcanzar un objetivo común. Por tanto el trabajo grupal no es una opción sino que muchas veces es una necesidad. En este sentido, Nunamaker distingue procesos grupales que se ven mejorados respecto a los individuales y otras que se ven perjudicados, tal y como se observa en la Tabla 2.1 [Nunamaker92].

Procesos mejorados:	
1. Más información.	Un grupo como tal tiene más información que cualquier miembro individual.
2. Sinergia.	Cada miembro utiliza la información de diferente forma a como lo haría de manera individual porque dentro del grupo posee más posibilidades.
3. Evaluación más objetiva.	Los grupos tienen más capacidad para evaluar y evitar errores que un individuo
4. Estimulación.	El trabajar dentro de un grupo puede estimular el trabajo de cada miembro.
5. Aprendizaje.	Los miembros pueden aprender de los más expertos.

Procesos perjudicados:	
1. Fragmentación del tiempo.	En una reunión, el tiempo se divide entre los participantes.
2. Bloqueo de producción.	No se realizan nuevos comentarios o aportaciones porque no se consideran importantes, porque se concentran en recordar los comentarios que van a realizar o porque se está constantemente escuchando a los otros.
3. Fallo al recordar.	Los miembros no prestan atención a los otros y no recuerdan sus contribuciones.
4. Presión de conformidad.	Los miembros huyen de realizar críticas a los otros por educación o para evitar represalias.
5. Aprensión de evaluación.	Una evaluación negativa provoca que los miembros no aporten nuevas ideas o comentarios.

6. Conducción/Paseo libre.	Un miembro no participa al advertir que sus compañeros lo hacen muy satisfactoriamente.
7. Inercia del conocimiento.	La discusión se realiza según el razonamiento de una persona ya que los demás miembros se abstienen de realizar comentarios.
8. Socialización.	Las discusiones sobre cuestiones no relacionadas con los objetivos del grupo reduce la eficiencia del mismo, aunque por otra parte, resulta necesario este tipo de comunicación dentro de un grupo.
9. Dominación.	Algunos miembros del grupo ejercen una fuerte influencia sobre los otros y monopolizan ineficazmente el tiempo del grupo.
10. Sobrecarga de información.	La información es presentada muy rápidamente y no puede ser procesada.
11. Problemas de coordinación.	El grupo carece de una apropiada organización que impide la integración y contribución de todos los miembros y puede dar lugar a decisiones erróneas.
12. Utilización incompleta de la información.	No se accede a toda la información necesaria o se utiliza de forma incompleta.
13. Análisis incompleto de las tareas.	Se realiza un análisis incompleto de la tarea a realizar lo que puede provocar discusiones superficiales.

Tabla 2.1. Ventajas e inconvenientes de los Procesos Grupales.

Estos procesos varían considerablemente dependiendo de las situaciones, y en algunos casos puede que no existan. Por ejemplo, en una reunión de un grupo, las pérdidas debidas a la fragmentación del tiempo dependen fundamentalmente del número de personas que forman dicho grupo. En general, la introducción de sistemas informáticos para facilitar el trabajo en grupo debe mejorar alguno de estos procesos grupales menos eficaces.

2.2.5 Tareas grupales

McGrath propone un modelo interesante de clasificación de las tareas que llevan a cabo los grupos [McGrath84]. Tal como se observa en la Figura 2.1, McGrath

diferencia cuatro procesos generales que se distribuyen en cada uno de los cuadrantes: generar alternativas, elegir alternativas, negociar y ejecutar.

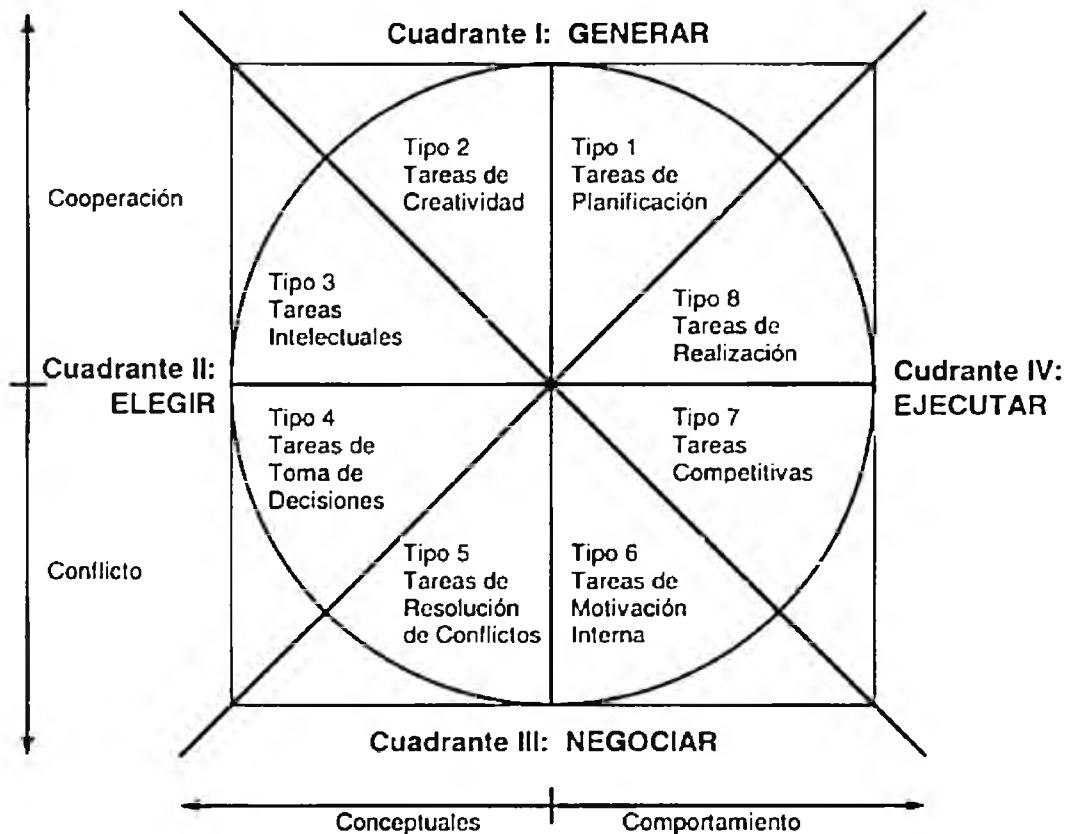


Figura 2.1. Modelo de Tareas Grupales.

Cada uno de estos procesos generales o cuadrantes se divide a su vez en dos tipos de tareas, como se describe en la Tabla 2.2.

Este modelo de tareas tiene dos dimensiones. La dimensión horizontal muestra a la derecha las tareas que implican acciones y a la izquierda las tareas conceptuales o intelectuales. La dimensión vertical recoge en la parte superior las tareas realizadas bajo cooperación o acuerdo y en la parte inferior las caracterizadas por la existencia de algún tipo de conflicto.

Cuadrante I	GENERAR
Tipo 1	Tareas de planificación, de generación de planificaciones.
Tipo 2	Tareas de creatividad, de generación de ideas.

Cuadrante II	ELEGIR
Tipo 3	Tareas intelectuales encargadas de encontrar la solución correcta a un determinado problema.
Tipo 4	Tareas de toma de decisiones en las que el grupo selecciona una alternativa o solución según los criterios del propio grupo, ya que no hay una única solución correcta.

Cuadrante III	NEGOCIAR
Tipo 5	Tareas de resolución de conflictos debidos a las diferentes preferencias de los miembros del grupo.
Tipo 6	Tareas de motivación interna orientadas a establecer coaliciones o negociaciones entre los miembros del grupo.

Cuadrante IV	EJECUTAR
Tipo 7	Tareas competitivas, en las que un grupo compite con otros en la consecución de un determinado objetivo.
Tipo 8	Tareas de realización, que no incluyen competencia, en las que el grupo se esfuerza por alcanzar unos determinados objetivos.

Tabla 2.2. Cuadrantes y tipos de Tareas Grupales.

En la misma línea, Nunamaker describe los procesos que llevan a cabo los grupos de la siguiente forma [Nunamaker91]:

- Generación de ideas. Analizar el problema y generar diversas ideas o alternativas.
- Organización de ideas. Identificar, sintetizar y formular las diversas propuestas.
- Ordenación. Seleccionar una determinada propuesta.

- Desarrollo. Llevar a cabo las acciones decididas.

La dinámica que se describe sobre el comportamiento de los grupos se puede trasladar a toda una organización o una empresa. Las dinámicas descritas a nivel del grupo (las normas, la comunicación, el liderazgo, los tipos de tareas, etc.) existen también a nivel de toda la organización [Cole92]. Por tanto, se puede concluir que la mejora del comportamiento de los grupos con la ayuda de sistemas informáticos significa una mejora del funcionamiento de toda la organización o de toda la empresa.

2.3 La coordinación y la comunicación

Otro de los requisitos para poder hablar de Trabajo Cooperativo es la existencia de algún tipo de comunicación y coordinación entre las personas participantes en esa tarea.

El concepto de coordinación es otro término que ha sido objeto de análisis en diversas disciplinas. En la práctica, en muchas situaciones se puede detectar la existencia de algún tipo de coordinación, por ejemplo, en una línea de montaje o en un equipo de baloncesto, pero resulta más fácil ser consciente de este concepto cuando se produce algún fallo. Por ejemplo, si en una línea de montaje una persona no realiza su labor en el momento oportuno, el trabajo de todos los demás se puede ver afectado.

En uno de los artículos más importantes sobre la coordinación, realizado por Malone, se parte de una definición general de coordinación :

la acción de trabajar conjunta y armoniosamente [Malone90].

Los autores también incluyen en esta amplia definición aquellas situaciones en las que aparece algún conflicto. Un grupo de trabajo puede tener importantes conflictos de intereses o diferentes puntos de vista, pero puede producir resultados que un observador externo puede considerar como buenos y obtenidos en armonía. Por

ejemplo, las divisiones de una compañía frecuentemente compiten entre sí por la obtención de mayores recursos económicos y de personal. Esta competencia puede resultar positiva y contribuir en la obtención de mejores resultados para la empresa.

La palabra *armoniosamente* implica que las actividades que se llevan a cabo conjuntamente no son independientes y que unas deben favorecer, y no entorpecer, la correcta realización de las otras. La relación que se establece entre estas actividades se denomina *interdependencia*. Estos autores han definido una *Teoría de la Coordinación*, que incluye un conjunto de principios sobre la forma de coordinación de las actividades, es decir, sobre cómo las personas pueden trabajar conjunta y armoniosamente.

Malone identifica cuatro componentes de la coordinación a los que asocian sus correspondientes procesos, tal como se observa en la Tabla 2.3.

Componentes	Procesos
Objetivos	Identificar los objetivos.
Actividades	Asignar actividades a los objetivos.
Actores	Asignar actores a las actividades.
Interdependencias	Gestionar las relaciones de dependencia entre las actividades.

Tabla 2.3. Componentes de la Coordinación.

Por ejemplo, una empresa fabricante de automóviles tiene varios objetivos (ej. producir diferentes modelos) y un conjunto de actores (ej. los empleados) que desarrollan actividades de cara a conseguir los objetivos. Estas actividades pueden tener diferentes tipos de interdependencias como el uso de un mismo recurso (ej. la línea de montaje) o la necesidad de realizar las tareas en un determinado orden (ej. el diseño del modelo debe realizarse antes que su construcción).

Esta definición general incluye casi todas las tareas que se llevan a cabo cuando se realiza un trabajo entre varias personas como son el establecimiento de los objetivos,

la selección de los actores, etc. Por esta razón el propio Malone estableció una definición más restringida que se centra en el elemento más específico de la coordinación que es la interdependencia de tareas. Así define coordinación como:

la acción de gestionar las interdependencias entre las actividades llevadas a cabo para conseguir un objetivo [Malone90].

Normalmente siempre que exista una coordinación van a estar involucrados varios actores, pero esta definición se centra en la relación de interdependencia de las actividades, sin tener en cuenta el que éstas sean llevadas a cabo por una o varias personas.

Malone analiza las interdependencias entre las actividades en base a *los objetos comunes* que intervienen en las acciones que se coordinan. Por ejemplo, en las actividades de diseño y fabricación de un objeto está involucrado el *diseño detallado* del mismo. En la actividad de diseño se crea el objeto *diseño detallado* y en la actividad de fabricación se utiliza. Entre ambas actividades se establece una dependencia de tipo *pre-requisito*. Esta dependencia queda establecida a través del objeto *diseño detallado*, que es necesario para poder comenzar la segunda actividad. Un mejor entendimiento de la coordinación debe ayudar a construir unas herramientas más eficientes de Trabajo Cooperativo.

Otro aspecto importante es la comunicación. Ya se ha mencionado previamente que dentro del ámbito del Trabajo Cooperativo algunos autores consideran un concepto amplio de comunicación que incluye tanto la comunicación directa interpersonal, como la comunicación indirecta que se realiza por medio del cambio de estado de un proceso de transformación.

La comunicación directa, que es como tradicionalmente se percibe el término comunicación, también ha sido objeto de numerosos estudios. El objetivo básico de cualquier comunicación es hacer llegar un mensaje desde un emisor a un receptor. Pero además hay otros elementos importantes relacionados con cualquier proceso de

comunicación como son la capacidad de influencia y los objetos que se comparten. Short llega a definir la comunicación como:

las señales físicas por las cuales un individuo puede influenciar el comportamiento de otro [Short76].

Cole realiza una interesante distinción entre intercambio de información y comunicación. El intercambio de información es un requisito para la comunicación, pero además la comunicación requiere que las partes involucradas tengan una comprensión común de la información [Cole92].

Por otra parte, Clark definió un modelo de comunicación basado en el concepto de base o sustrato común, en el que se relaciona la comunicación con la coordinación como se refleja en el siguiente ejemplo. Tomamos dos artistas que van a tocar conjuntamente una pieza de piano. Ambos tendrán que coordinar el contenido y el proceso de lo que van a hacer. En la coordinación del contenido, tendrán que llegar a un acuerdo sobre la pieza que van a tocar. En la coordinación del proceso, tendrán que sincronizar la entrada, la velocidad o el tono. Pero ambos artistas no podrán comenzar la coordinación del contenido sin asumir previamente gran cantidad de información que se denomina base o sustrato común, que correspondería con un conocimiento mutuo o una similar base musical. Asimismo, para poder mantener una coordinación del proceso necesitan actualizar continuamente ese sustrato común, lo que permitiría una reacción rápida de ambos ante cualquier anomalía. Por tanto, Clark concluye que todas las acciones colectivas se construyen tomando como referencia un sustrato común y sus continuas actualizaciones [Clark90].

2.4 Algunas conclusiones

Hasta este momento se ha presentado una breve introducción de lo que es el Trabajo Cooperativo, sin entrar en consideraciones sobre las posibilidades que los

sistemas informáticos tienen para mejorar estos procesos. Además el estudio de este concepto de Trabajo Cooperativo comenzó a realizarse antes de la propia existencia de los ordenadores.

Como se podrá observar en el desarrollo de esta tesis, el estudio del Trabajo Cooperativo sigue resultando una necesidad básica en sí misma, y también es necesario para poder desarrollar sistemas informáticos eficientes de ayuda al trabajo en grupo. Cuestiones como en qué se diferencia el trabajo cooperativo del trabajo individual, cómo trabajan los grupos, cómo trabajan los individuos dentro de un grupo, cómo se lleva a cabo la coordinación y la comunicación grupal necesitan todavía ser objeto de un mayor estudio. Los conocimientos así adquiridos facilitarán la creación de sistemas informáticos que ayuden a la realización de estos procesos grupales.

Para llevar a cabo esta tarea se necesita un esfuerzo multidisciplinar en el que tienen que participar ciencias como la antropología, la sociología, la psicología y la informática. Esta cooperación entre expertos de diferentes ciencias es un reto y al mismo tiempo puede servir de evaluación de los modelos de cooperación y de los posibles sistemas informáticos de apoyo al Trabajo Cooperativo que se vayan implementando.

3. CSCW y Groupware

3.1 CSCW

Durante años las herramientas y aplicaciones informáticas han sido utilizadas por una sola persona, sin establecerse relación con otras personas cercanas que empleaban los mismos sistemas u otros similares [Rodden91b]. Este aislamiento empezó a romperse con la aparición de la tecnología de red que interconectaba varios equipos informáticos y permitía la compartición por un grupo de personas de una serie de recursos como ficheros, programas, impresoras, etc. Posteriormente surgieron aplicaciones como el correo electrónico que facilitaban la comunicación grupal. Estas tecnologías son los primeros ejemplos del soporte que los sistemas informáticos pueden dar al trabajo en grupo.

Según datos recogidos en los informes *DATA PRO*, muchos procesos de trabajo son todavía muy lentos a pesar del grado de automatización existente [Datapro94]. Algunos expertos afirman que durante la década de los años 80 la industria americana no ha mejorado significativamente su productividad y sin embargo ha doblado sus inversiones en elementos hardware y software. Como ejemplo se presenta una aplicación de contratación de seguros de vida que tarda 22 días en generar la póliza. De ese tiempo, el proceso de aceptación o rechazo del seguro dura 17 minutos. El resto del

tiempo se invierte en procesos de interacción y comunicación entre el grupo de trabajo y personas externas [Datapro94]. Las herramientas informáticas individuales no pueden mejorar estos procesos, los cuales deben ser abordados por herramientas que den soporte al trabajo en grupo y permitan que una tarea sea llevada a cabo por varias personas. La tendencia en la década de los 90 es pasar del concepto de la productividad personal al concepto de la productividad de la organización [Marinak94].

Desde mediados de los años 80 empieza a utilizarse el término *Computer Supported Cooperative Work (CSCW)* (Trabajo Cooperativo Soportado por Ordenador) para definir el campo de investigación que aborda cómo los sistemas informáticos pueden dar soporte al trabajo grupal. Esta área intenta superar la barrera del ordenador personal y aproximarse a los procesos reales de trabajo en los que existe una interacción continua entre diferentes personas.

El término CSCW fue utilizado por primera vez en 1984 por Greif del MIT y Cashman de DEC, quienes organizaron un seminario al que asistieron unas 20 personas de diferentes disciplinas interesadas en analizar cómo se organizan los grupos y cómo llevan a cabo sus trabajos. Dos años después, en 1986, ya se celebró la primera conferencia sobre CSCW en Estados Unidos y en 1989 tuvo lugar la primera conferencia europea [Grudin94].

Como todo nuevo campo de investigación su ámbito de trabajo, sus límites, su definición e incluso su propio nombre son objeto de discusión. Greif, creadora del término, afirma que lo escogieron para poder referirse a los aspectos relacionados con un grupo de personas trabajando conjuntamente con la ayuda de sistemas informáticos. Y reconoce también que no esperaban que se hiciera tanto hincapié en cada una de las palabras que forman el término y sus posibles implicaciones. Esto ha sucedido en parte por la ausencia de una única interpretación del significado de CSCW.

En este sentido, Bannon presenta el término CSCW como un concepto amplio y únicamente lo describe como:

el campo que cubre todo lo relacionado con el soporte informático de las actividades en las cuales está involucrada más de una persona [Bannon91].

Bannon presenta este campo como un foro amplio en el que deben participar investigadores de diferentes especialidades. Este carácter multidisciplinar del CSCW es aceptado por la mayoría de los autores [Grudin94] [Kling91]. Incluso algunos establecen una diferencia entre campo de investigación y foro. Kling considera que el CSCW es únicamente un foro de encuentro en el que personas con experiencia en diversos campos de investigación se reúnen para mostrar su trabajo y aprender del realizado por los demás, pero mantienen una identidad propia e independiente en cada uno de los campos de los que provienen.

Pero también se puede considerar al CSCW un área de investigación, tomando como referencia la definición del sociólogo Whitley, quien considera que un área de investigación se define únicamente por la existencia de un problema en unas determinadas circunstancias [Whitley74]. Siguiendo esta definición, el CSCW es el área que trata de resolver el problema del soporte informático al trabajo realizado por un grupo.

Por otra parte, Ellis propone otra definición general, pero poniendo más hincapié en el aspecto sociológico del trabajo cooperativo, así afirma que:

el CSCW investiga cómo trabajan los grupos e intenta descubrir la forma en que la tecnología y especialmente los ordenadores pueden facilitar ese trabajo [Ellis91].

La conjunción de la potencia de los ordenadores y las nuevas formas de comunicación electrónica deben mejorar, según Ellis, la interacción entre las personas y posibilitar la creación de sistemas que integren el procesamiento de la información y las actividades de comunicación. No obstante, Ellis ya avanza que la principal dificultad para desarrollar sistemas eficientes va a estar en los aspectos sociales y de organización.

Es decir, en la asunción de dichos sistemas por parte de quienes realizan tareas en colaboración o que requieran la participación de más de una persona.

En la misma línea, Kling critica las definiciones del CSCW que se centran en exceso en el aspecto tecnológico. En este sentido, considera que el CSCW es una conjunción de una serie de tecnologías, un conjunto de usuarios (normalmente un grupo de profesionales) y un entorno de trabajo [Kling91]. El CSCW no puede ser definido en base únicamente a las tecnologías aplicadas, ya que se sitúa al nivel de diseño de aplicaciones y por tanto es posible la utilización de cualquier tipo de tecnología como Inteligencia Artificial, sistemas distribuidos, comunicaciones, etc. Lo que caracteriza a las aplicaciones del CSCW es que satisfacen los requisitos de trabajo cooperativo.

Si se analizan las palabras del término CSCW se pueden diferenciar dos conceptos. Por una parte, el concepto de *Trabajo Cooperativo (CW)* que ha sido ya presentado en el capítulo anterior. El CW es en sí mismo un área de investigación que, como ya se ha afirmado, resulta clave para garantizar el éxito del CSCW. En estos momentos, la principal dificultad para diseñar sistemas informáticos eficientes que den soporte al trabajo cooperativo radica todavía en la falta de comprensión de cómo trabajan las personas y cómo funcionan los grupos y las organizaciones.

La otra parte del término, *Soportado por Ordenador (CS)*, hace referencia al apoyo que con los sistemas informáticos se pretende dar al Trabajo Cooperativo. En estos momentos hay todavía pocos sistemas completos y eficientes que den soporte a este tipo de trabajo. Todavía al principio de los 90 se llegaba a afirmar que la primera labor de los ordenadores en las actividades cooperativas era la de no molestar [Bannon91]. Esta es una afirmación exagerada pero que demuestra la dificultad de esta área para producir sistemas comerciales ampliamente utilizados.

La conjunción de estas dos partes forman el término CSCW, que integra dos tareas fundamentales y complementarias como son el seguir analizando el trabajo cooperativo, es decir, cómo trabajan las personas y los grupos, y el ir desarrollando

prototipos y sistemas informáticos que intentan facilitar los procesos de trabajo realizado por varias personas.

3.2 Groupware

Relacionado con el término CSCW aparece otro concepto: Groupware. No existe una única definición del mismo y tampoco existe una asunción universal sobre cuál es la relación entre CSCW y Groupware.

El término Groupware fue utilizado por primera vez por Johnson-Lenz en 1982, y por tanto antes incluso de la aparición del concepto CSCW. Hacía referencia a los procesos y los procedimientos que llevaban a cabo los grupos de trabajo junto con los sistemas informáticos utilizados [Johnson-Lenz82]. Posteriormente Johansen en 1988 restringió la definición únicamente a los sistemas informáticos que daban soporte a un grupo de trabajo [Johansen88]. Este último planteamiento es el seguido por Ellis, cuya definición es hoy en día la más utilizada dentro de la comunidad que estudia el trabajo cooperativo:

sistemas que dan soporte a grupos de personas involucradas en una tarea común (o en un objetivo común) y que proporcionan una interfaz a un entorno de trabajo compartido [Ellis91].

En esta definición, los elementos *tarea común* y *entorno de trabajo compartido* resultan claves a la hora de determinar si un sistema se puede considerar como producto groupware. Por ejemplo, no se consideran productos groupware los sistemas multi-usuario como las bases de datos ya que quienes los utilizan no comparten un objetivo común. Por otra parte, esta definición incluye sistemas de correo electrónico en los que el concepto de cooperación es más débil.

Stodolsky presenta otra definición similar que excluye la condición de *objetivo común* y se centra en los procesos con un claro carácter de cooperación:

Groupware es el software y hardware para un entorno de trabajo compartido e interactivo [Stodolsky95].

El término *entorno de trabajo* incluye el software y el hardware que establece el contexto para la interacción. El término *interactivo* indica que el aspecto temporal está controlado por el sistema, ya sea síncrona o asíncronamente. Asimismo, el término *compartido* refleja que dos o más participantes interactúan entre sí de forma que cada persona se ve influenciada por los demás. Stodolsky suprime la condición de *objetivo común* para no poner en duda la etiqueta de groupware de algunos sistemas que se caracterizan por la existencia de algún tipo de conflicto entre los participantes, por ejemplo los sistemas de votación.

Otras definiciones hacen más hincapié en la información, así Sawyers define groupware como:

los sistemas que facilitan el procesamiento y uso de información entre grupos de trabajo que pueden estar geográficamente u organizativamente dispersos [Sawyers94].

Otros autores introducen el concepto de productividad y definen groupware como:

aquellos sistemas que mejoran la productividad de un grupo de personas [Marshak94].

Marshak también propone una serie de condiciones que deben cumplir las aplicaciones para que sean consideradas como sistemas groupware:

- Las aplicaciones groupware están diseñadas para ser usadas por un grupo de personas y no son versiones de red de herramientas personales.
- Las aplicaciones groupware tienen poco o ningún valor como sistemas individuales usados por una sola persona.

- Las aplicaciones groupware mejoran la productividad del grupo sin disminuir la productividad de cada uno de sus miembros.
- Las aplicaciones groupware están disponibles a todos los miembros del grupo independientemente de su localización y su plataforma hardware.

Según Marshak, las dos últimas condiciones no son estrictamente necesarias, pero su cumplimiento puede garantizar que se trate de un buen producto groupware.

3.3 Relación entre CSCW y Groupware

Los términos CSCW y Groupware son para algunos autores dos nombres de un mismo campo de investigación, que además también recibe otras denominaciones como *Coordination Technology*, *Group Decision Support Systems (GDSS)*, *Electronic Meeting Systems (EMS)*, *Collaboration Technology*, etc. [Palmer94] [Kremer91].

El término Groupware fue utilizado por primera vez por Johnson-Lenz en 1982 para hacer referencia a los procesos grupales y los sistemas informáticos que les daban soporte. Esta definición puede aplicarse actualmente al término CSCW. La definición de Groupware más extendida en estos momentos es la propuesta por Johansen y Ellis que incluye únicamente los sistemas informáticos que dan soporte al trabajo en grupo.

Se puede entonces considerar al CSCW como el área de investigación que aborda el trabajo cooperativo y su soporte informático, y al Groupware como el área que estudia y desarrolla sistemas informáticos para dar soporte al trabajo cooperativo [Kling91] [Marshak94] [Kremer91]. De forma clara y escueta esta relación queda descrita en el título del libro de Marca y Bock: "*Groupware: Software for Computer Supported Cooperative Work*" [Marca92].

Por otra parte, Stodolsky, siguiendo su definición de Groupware, presenta ambos términos como el nombre genérico de diferentes aplicaciones algunas de las

cuales pertenecerían a ambas categorías y otras sólo a una. Por ejemplo, la capacidad de interacción entre los usuarios sería una característica únicamente de los sistemas Groupware, o con otro ejemplo los sistemas de correo electrónico únicamente pertenecerían a la categoría de CSCW [Stodolsky95].

Siguiendo la relación más universalmente aceptada entre CSCW y Groupware, Kremer distingue tres subáreas de investigación dentro del área general de CSCW [Kremer91]:

1. Estudio de los grupos de trabajo. Se encarga de profundizar en cómo los grupos de personas se organizan y llevan a cabo una tarea, teniendo en cuenta los factores que intervienen: la coordinación, la comunicación o la colaboración. Es un área que requiere la participación de diversas disciplinas como la sociología, la psicología o las teorías de organización. Esta área se ha descrito brevemente en el capítulo anterior.
2. Desarrollo de herramientas que dan soporte al trabajo cooperativo. Es el concepto de Groupware que se va a utilizar en este trabajo.
3. Evaluación de los sistemas. Estudia los criterios necesarios para evaluar los sistemas de ayuda al trabajo en grupo. Se debe analizar asimismo el impacto que provoca su introducción en una organización y los cambios que es necesario llevar a cabo en los hábitos de trabajo adquiridos.

El campo de investigación del CSCW requiere un esfuerzo interdisciplinar. Ellis utiliza la palabra interdisciplinar en lugar de multidisciplinar para resaltar que los análisis y las contribuciones de las diferentes disciplinas, incluyendo los usuarios finales, deben ser integradas y no únicamente tenidas en cuenta [Ellis91]. El éxito de los sistemas Groupware no depende en este momento tanto del desarrollo de nuevas tecnologías como de la comprensión del funcionamiento de los grupos de trabajo y del diseño de aplicaciones que satisfagan realmente los requisitos de los usuarios que forman parte de estos grupos para tratar de mejorar las tareas grupales que llevan a cabo.

En este trabajo se van a adoptar las definiciones más generales de CSCW y Groupware, con la relación anteriormente descrita. En un campo de investigación todavía bastante reciente resulta conveniente seguir manteniendo estas generalidades para no establecer exclusiones significativas en las propias definiciones. Este hecho implica que hay que realizar un mayor esfuerzo de clasificación de los sistemas para que se puedan diferenciar claramente las características de cada uno de ellos.

3.4 El éxito de las aplicaciones Groupware

La aparición de una nueva área de investigación o una nueva tecnología informática crea habitualmente grandes expectativas y la esperanza de contar en poco tiempo con aplicaciones innovadoras. Asimismo es habitual que tras un periodo inicial de euforia aparezca un cierto desencanto al comprobar que no se consiguen de forma rápida los resultados esperados. Esta situación se ve agravada si el nombre de esa nueva área es un tanto confuso o puede dar lugar a diferentes interpretaciones. Estas circunstancias están ocurriendo en este momento con el CSCW o el Groupware, como en el pasado puede haber ocurrido con otras áreas como la Inteligencia Artificial.

Por otra parte, según afirma Rogers, cualquier idea nueva necesita un cierto tiempo para que pueda ser asimilada por las personas [Rogers83]. Además en este caso estamos hablando de un campo que presenta serias dificultades entre las que destacan las siguientes:

- El CSCW es un área que en su aplicación práctica modifica la forma en que habitualmente trabajan las personas. Esta tarea obviamente no resulta fácil y es necesario que se lleve a cabo siguiendo un determinado proceso para que los destinatarios puedan asimilar y aceptar estos cambios. En muchas ocasiones no se consiguen los objetivos deseados.
- El CSCW intenta dar soporte informático al trabajo cooperativo o trabajo en grupo. Pero este campo requiere todavía una mayor profundización para poder dar mejores respuestas a preguntas del tipo: ¿cómo trabajan los grupos y las

organizaciones?, ¿cómo es el comportamiento y el trabajo de una persona dentro de un grupo?, etc. Esta investigación corresponde a ciencias como la sociología, la psicología o la antropología, cuyos resultados se deben integrar con los obtenidos por el estudio de las tecnologías existentes que puedan dar soporte al trabajo en grupo.

Realizando un breve repaso de los estudios elaborados sobre el éxito de las aplicaciones cooperativas se pueden detectar algunas ideas significativas. En 1988 Grudin presentaba en la Conferencia de CSCW en Portland un artículo citado frecuentemente titulado "*Why CSCW Applications Fail...*" [Grudin88]. En este artículo se afirmaba que normalmente resulta desproporcionado el esfuerzo y las inversiones realizadas en un sistema Groupware frente a los resultados obtenidos, ya que no se suelen alcanzar los objetivos iniciales.

En la siguiente Conferencia sobre CSCW en Los Angeles en 1990, Bullen y Bennett presentaron otro trabajo que recogía las conclusiones de un estudio sobre el uso de varias aplicaciones Groupware en 25 empresas. Concluían que estas herramientas estaban en su fase inicial y que todavía no eran consideradas como herramientas que pudieran mejorar el trabajo grupal [Bullen90].

Argumentos semejantes son defendidos por otros autores como Ellis, Bannon y Kling en artículos presentados en 1991. Ellis afirma que aunque los objetivos de los sistemas Groupware parecen claros, existe una historia caracterizada por los fracasos de estas aplicaciones [Ellis91]. Otros se cuestionan la palabra *Support* dentro del término CSCW y manifiestan que todavía el primer objetivo de la tecnología respecto al trabajo cooperativo debe ser el de no entorpecerlo (*not disrupt*) [Bannon91]. Por su parte, Kling compara el gran número de prototipos existentes, muy útiles en las investigaciones y estudios realizados, con los escasos productos comerciales disponibles [Kling91].

Aunque todos estos autores concluyen que en líneas generales las aplicaciones Groupware no han alcanzado todavía un mínimo grado de madurez y aceptación, todos defienden la necesidad y la conveniencia de seguir profundizando en este campo.

En estos momentos ya se empiezan a percibir algunos cambios. Así en un estudio realizado por la *City University Business School* de Londres y presentado en el congreso *Groupware'95* en Londres se afirma ya que muchas compañías están obteniendo importantes ventajas por el uso de aplicaciones Groupware, independientemente del tipo y tamaño de la organización. Asimismo se concluye que hay grandes posibilidades de rentabilizar la inversión realizada al introducir estos nuevos sistemas [Turrell95]. Por tanto, según este estudio se estaría empezando a resolver una de las principales dificultades observadas por Grudin en 1988, que consistía en que las inversiones no se estaban rentabilizando. No obstante, hay que constatar que el estudio de la *Business School* de Londres estaba basado en la utilización del producto *Lotus Notes*, uno de los más consolidados en el mercado. En otros sistemas de soporte al trabajo en grupo la rentabilidad no es tan palpable y por tanto siguen existiendo las dificultades apuntadas por Grudin.

El propio Grudin también afirma que la situación está modificándose al presentar en la Conferencia *ECSCW'95* en Estocolmo un artículo titulado "*Why Groupware Succeeds...*" [Grudin95], en clara referencia a su artículo presentado en 1988 "*Why CSCW Applications Fail...*" [Grudin88]. En el nuevo artículo se realiza un estudio sobre la utilización de agendas electrónicas para planificar las reuniones en dos grandes organizaciones. Los resultados demuestran que estas aplicaciones son satisfactorias y ampliamente utilizadas, y además de forma voluntaria.

En la realización de todos estos estudios, los autores han identificado una serie de dificultades, que condicionan la adopción por parte de una organización de un sistema Groupware. Los más destacados son los siguientes:

1. En algunas ocasiones, estas aplicaciones incrementan el trabajo que realizaban los usuarios antes de su introducción. En otras ocasiones, se necesita que algunas personas realicen un trabajo nuevo adicional y estas personas no son quienes se benefician del uso de estos sistemas [Grudin88].
2. En la etapa de diseño existe una falta de experiencia en aplicaciones multi-usuario. Se tiende a realizar una analogía, a veces inconscientemente, entre las

aplicaciones mono-usuario y las nuevas aplicaciones multi-usuario que no produce buenos resultados [Grudin88].

3. Resulta difícil evaluar estas aplicaciones y por consiguiente también resulta complicado ir aprendiendo de las experiencias existentes [Grudin88].
4. Una de las principales dificultades radica en la falta de conocimiento a priori de la utilidad de los sistemas Groupware. En una encuesta realizada en 1992 a investigadores, desarrolladores y usuarios se indicaba que éste era el principal obstáculo de cara a la introducción de los sistemas groupware [Butterfield92]. En este sentido es fundamental demostrar las posibilidades de estos sistemas con procesos reales de la organización en donde se estudia su implantación.
5. Cualquier sistema Groupware debe ser introducido siguiendo un proceso previamente definido. La forma en que inicialmente se instala un nuevo sistema resulta clave en la asunción definitiva de dichas aplicaciones. Esta implantación supone un proceso de cambio tecnológico, pero también un proceso de cambio social ya que se pueden ver modificadas las posiciones y las relaciones que mantienen los usuarios.
6. Las aplicaciones se deben adaptar inicialmente a la cultura existente en la organización. Posteriormente ya se podrán abordar cambios más profundos [Turrell95], incluso aprovechando la introducción de los nuevos sistemas. En este sentido, Orlikowski afirma que el uso de la tecnología Groupware facilita la introducción de los cambios en la organización [Orlikowski95]. Estos cambios pueden ser en la distribución del trabajo, la forma de colaboración, la coordinación entre departamentos, etc. Pero asimismo indica que hay que ser conscientes de que junto a los cambios planificados pueden aparecer otros cambios que van surgiendo durante el proceso de adaptación.

Resulta difícil establecer conclusiones generales sobre el éxito de las aplicaciones Groupware debido a la diversa naturaleza de cada una de ellas. Asimismo, la disponibilidad de los sistemas y su posible impacto es valorado de forma muy diversa

como se puede comprobar en la encuesta realizada por Butterfield en 1992 [Butterfield92].

Este estudio fue realizado mediante encuestas recibidas por correo electrónico entre investigadores, desarrolladores y usuarios finales de sistemas informáticos. En la Figura 3.1 se valora el posible impacto de los sistemas groupware, es decir, su grado de utilidad. La valoración se realiza considerando el 0 como un impacto nulo y el 10 como un producto de gran utilidad.

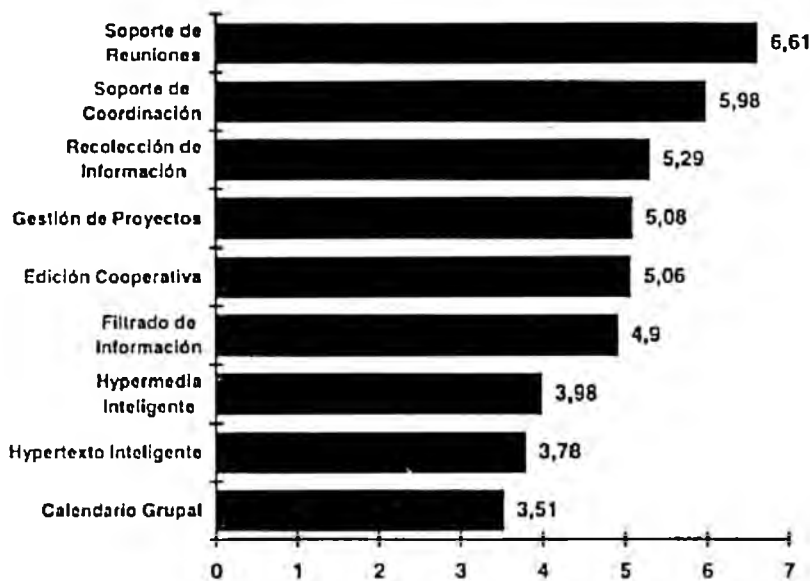


Figura 3.1. Posible impacto de los Sistemas Groupware (valoración de 0 a 10).

En la Figura 3.2, en cambio, se recoge la opinión sobre la disponibilidad de los sistemas en ese momento, en 1992. Una puntuación de 0 indica que no existían productos disponibles y una puntuación de 10 refleja una amplia disponibilidad de los mismos.

En general, las puntuaciones de la Figura 3.1 (posible impacto) son más bajas que las de la Figura 3.2 (disponibilidad), es decir, se considera que el impacto de estos

sistemas es bajo, lo que indica que existían bastantes dudas sobre su viabilidad. Se observa como las aplicaciones de calendario grupal son escasamente valoradas (3,51) a pesar de contar con un fácil acceso a ellas (7). El hecho de que las más disponibles sean las menos valoradas es un reflejo del desencanto existente. En el extremo contrario, se sitúan las aplicaciones de soporte a las reuniones. Se valora en gran medida su utilidad (6,61), y se considera que ya existen aplicaciones que respondan a esa demanda de los usuarios (5,88).

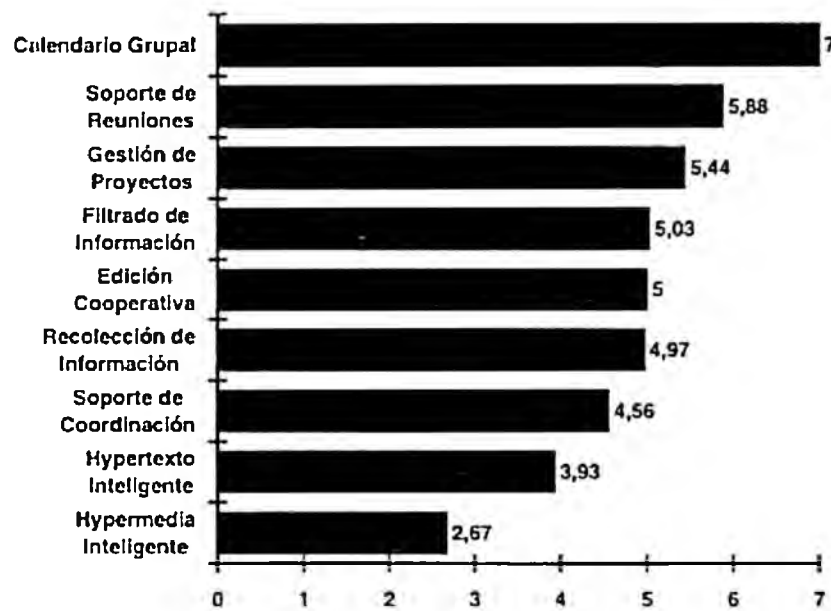


Figura 3.2. Disponibilidad de los Sistemas Groupware (valoración de 0 a 10).

Estas aplicaciones de calendario grupal o agendas electrónicas grupales no han conseguido sustituir a las agendas tradicionales porque no ofrecen toda la funcionalidad de estas últimas. Cualquier agenda tradicional además de contener las citas diarias incluye anotaciones personales de todo tipo: números de teléfonos, notas escritas, papeles añadidos (los tradicionales papeles amarillos), etc. Estos elementos se incorporan con dificultad en una agenda electrónica. Por ejemplo, las notas recogidas en papeles externos en algunas ocasiones no se incluyen en la versión electrónica de la agenda, por lo que ésta no contiene la información más actualizada [Bullen90].

Igualmente si estas aplicaciones se quieren utilizar para planificar una reunión entre varias personas, todas ellas deben hacer uso del sistema para que resulte eficaz. Con esta condición, según apuntaba Grudin, se crea un problema importante ya que quien más se beneficia del sistema, quien convoca la reunión, no es quien debe realizar un mayor esfuerzo, aquel que se ve obligado a utilizar el sistema sólo para poder participar en esa reunión. Estos últimos usuarios deben realizar un trabajo al que no están habituados y lo perciben como un esfuerzo adicional [Grudin88].

Aunque actualmente estas aplicaciones de planificación grupal de reuniones están bastante lejos de ser universalmente adoptadas, se empieza a ver algunos cambios. Según el último trabajo de Grudin, está creciendo la implantación de las Agendas Electrónicas Grupales en las grandes compañías de ingeniería e informática [Grudin95]. Grudin realizó su estudio en las empresas *Microsoft Corporation* y *Sun Microsystems*. Las razones para este éxito relativo son las siguientes:

- La infraestructura con la que cuentan estas organizaciones.
- La mejora de la funcionalidad de estas aplicaciones.
- La mejora de la interfaz de usuario.

3.5 Clasificación de los sistemas Groupware

De la misma forma que no existe una única definición de los conceptos Groupware o CSCW tampoco existe una clasificación única de los sistemas que pertenecen a esta área de investigación.

Los criterios de división más utilizados son los siguientes:

- El espectro Groupware.
- La tabla espacio-temporal.
- La clasificación funcional.

3.5.1 El espectro Groupware

Según este criterio de división, los sistemas Groupware se colocan dentro de un amplio espectro multidimensional [Ellis91]. Las dimensiones de este espectro corresponden a los requisitos propuestos en la propia definición de Ellis:

sistemas que dan soporte a grupos de personas involucradas en una tarea común y que proporcionan una interfaz a un entorno de trabajo compartido [Ellis91].

Los requisitos son la existencia de una *tarea común* y un *entorno de trabajo compartido*. Los sistemas ocuparán un lugar en este espacio dependiendo del grado de cumplimiento de estos requisitos.

Respecto a la dimensión *tarea común*, un sistema de control horario ocupará una posición baja en el espectro. Estos sistemas permiten que varios usuarios introduzcan de forma independiente los datos de las tareas que realizan y su duración. Esta forma de trabajo implica poca relación entre los usuarios y por tanto dentro de la dimensión de *tarea común* se situará en una posición baja.

Mientras que un sistema de evaluación grupal del software ocupará el extremo opuesto. Este sistema permite que un grupo de desarrolladores evalúen de forma simultánea un módulo de software. Estas personas se concentran en alcanzar un mismo objetivo al mismo tiempo y con una estrecha comunicación. Este sistema se situará en una posición alta dentro de la dimensión *tarea común*.

Por otra parte, los sistemas también se pueden clasificar atendiendo a la dimensión del *entorno compartido*. De esta forma los sistemas de correo electrónico ocuparán una posición baja en esta dimensión ya que cada usuario mantiene su propio entorno de trabajo. Para algunos autores estos sistemas incluso quedarían fuera de este espectro ya que no los consideran como sistemas Groupware [Stodolsky94] [Sawyer94].

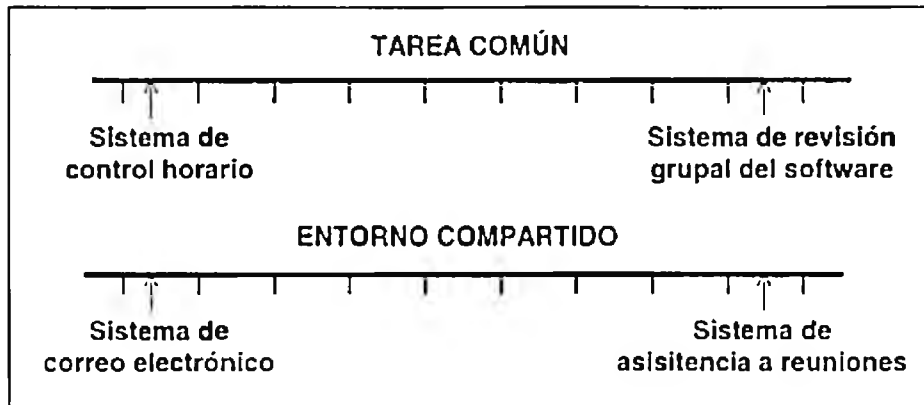


Figura 3.3. Espectro Groupware.

En cambio, un sistema de soporte a reuniones ocupará un lugar alto en el espectro. Por medio de estos sistemas, varios usuarios pueden compartir información, recibir una conferencia, realizar preguntas o ser conscientes del resto de participantes de la reunión. Se crea un espacio compartido al que pueden acceder los usuarios y por tanto se situará en una posición alta dentro de la dimensión de *entorno compartido*. En la Figura 3.3 se realiza una representación de estas dimensiones en la que se han colocado los ejemplos descritos anteriormente.

3.5.2 Tabla espacio-temporal

La tabla espacio-temporal ideada por Johansen en 1988 es un método ampliamente utilizado en la clasificación de los sistemas Groupware [Johansen88] [Johansen91]. Esta división se basa en la forma en que trabajan los grupos y no tanto en aspectos tecnológicos. En dicha tabla se refleja el momento y el lugar en que se realizan las actividades grupales. Es decir sus coordenadas son:

1. El tipo de interacción: síncrona (al mismo tiempo) o asíncrona (en diferentes momentos). La cooperación es síncrona cuando los participantes están actuando simultáneamente, por ejemplo, en la edición conjunta de un documento. Estos sistemas también reciben el nombre de Groupware en Tiempo Real. La cooperación es asíncrona cuando se lleva a cabo en diferentes momentos a lo largo del tiempo, como ocurre en los sistemas de correo electrónico.
2. La localización física: local (mismo lugar) o remoto (diferente lugar). La cooperación es local si los usuarios están en el mismo lugar físico, como en los sistemas para el soporte de reuniones. La cooperación es remota si estos usuarios se encuentran en diferentes lugares, como ocurre en el correo electrónico. En la actualidad, la división en función de la localización no depende tanto de la cercanía física de los usuarios como de su accesibilidad a través de diversos medios de comunicación.

Diferente Lugar	<i>Videconferencia</i>	<i>Correo electrónico</i> <i>Gestión de procesos</i>
Mismo Lugar	<i>Sistemas electrónicos de apoyo a reuniones</i>	<i>Compartición de ficheros</i>
	Mismo Tiempo	Diferente Tiempo

Figura 3.4. Tabla espacio-temporal.

De esta forma, como se observa en la Figura 3.4, se crean cuatro cuadrantes en una matriz espacio-temporal que serán ocupados por los sistemas Groupware. Estos cuadrantes son los siguientes:

1. Mismo Tiempo / Mismo Lugar. Este cuadrante contiene los sistemas que dan soporte a las reuniones. Se denominan *Sistemas Electrónicos de Apoyo a Reuniones (Electronic Meeting Support Systems, EMS)* e incluyen funcionalidades como mecanismos de votación, de recogida de ideas, de ayuda a la toma de decisiones, etc.
2. Mismo Tiempo / Diferente Lugar. Incluye los sistemas que dan soporte a un grupo de personas que trabajan simultáneamente desde diferentes lugares, como por ejemplo los sistemas de videoconferencia.
3. Diferente Tiempo / Diferente Lugar. Este cuadrante incluye los sistemas que dan soporte a la cooperación asíncrona. Algunos ejemplos son las aplicaciones de correo electrónico o los sistemas de gestión de procesos (*workflow*).
4. Diferente Tiempo / Mismo Lugar. Este cuadrante incluye los sistemas que permiten acceder a diferentes usuarios, en cualquier momento, a una información común. Los sistemas de compartición de ficheros se situarían en este cuadrante.

Los sistemas Groupware más interesantes son aquellos que ocupan más de un cuadrante de esta matriz. Lo ideal es que pudieran dar soporte a cualquier actividad grupal, es decir, que pudieran encajarse bajo la etiqueta *Cualquier Tiempo / Cualquier Lugar*. En la actualidad no existen estos sistemas universales y cada uno ofrece unas determinadas funcionalidades. Por tanto, además de esta clasificación basada en la forma en la que trabajan los grupos también es interesante clasificar los sistemas por la funcionalidad que ofrecen.

3.5.3 Clasificación Funcional

Esta división responde a las tareas que realizan los sistemas Groupware. Se distinguen los siguientes grupos:

a) *Sistemas de mensajes*

Como se ha mencionado anteriormente, algunos autores no otorgan el calificativo de Groupware al correo electrónico [Stodolsky94] [Sawyer94]. Pero para la gran mayoría, se trata de uno de los sistemas Groupware de mayor éxito y utilización hasta el momento [Ellis91] [Kling91]. Se pueden distinguir los sistemas tradicionales de correo electrónico (ej. *cc:Mail*, *Microsoft Mail*) en que el destinatario es un usuario específico (o una lista de usuarios) y las conferencias sobre diferentes temas (ej. *News*, *BBS*). En los primeros se mantiene un buzón individual por cada usuario donde es depositada la información mientras que en los últimos, el usuario se conecta a una repositorio central para acceder a la información.

b) *Sistemas de conferencia en tiempo real*

Estos sistemas permiten a un grupo de usuarios que se encuentran en diferentes lugares interactuar simultáneamente a través del ordenador. En la actualidad esta comunicación puede ser a través de diferentes medios como el vídeo, el audio y los documentos. Algunos ejemplos son: *MMConf* [Crowley90], *VConf* [Lantz86], *Dialogo* [Lauwers90], *Rapport* [Ahuja88] o *MERMAID* [Watabe90]. Estos sistemas van a ser analizados detalladamente en esta tesis y en concreto aquellos que permiten la creación de documentos, que pueden ser modificados en tiempo real por cualquier usuario que participa en la conferencia.

c) *Sistemas de soporte a reuniones*

En general se trata de sistemas que apoyan las tareas de organización y desarrollo de una reunión. Se pueden distinguir dos tipos:

1. *Sistemas de agenda electrónica*. Se encargan de determinar la hora óptima para desarrollar una reunión gestionando las diferentes agendas individuales de las personas involucradas. Existen en el mercado gran variedad de estos sistemas como *Network Scheduler* [Cummings94] o *SCHEDULE+* de *Microsoft* [Grudin95].

2. Sistemas de soporte al desarrollo de las reuniones. Estos sistemas son conocidos como *Sistemas Electrónicos de Apoyo a Reuniones (Electronic Meeting Systems, EMS)* y se instalan en una sala que normalmente cuenta con una pantalla de proyección y unos equipos informáticos conectados entre sí. Proporcionan diferentes ayudas para el desarrollo de la reunión como sistemas de votación, de recogida de propuestas, o de ayuda a la toma de decisiones. Estos últimos reciben el nombre específico de *Sistemas de Apoyo a la Toma de Decisiones (Group Decision Support Systems, GDSS)*. Algunos ejemplos son: *CoLab* desarrollado por *Xerox* [Rodden91] o el proyecto *NICK* del *MCC* [Rodden91].

d) Sistemas de creación compartida de documentos

Estos sistemas permiten la cooperación asíncrona en la creación de un documento, gestionando la participación de varias personas en distintos momentos. Normalmente permiten revisar y hacer anotaciones sobre el documento de trabajo. El ejemplo más característico lo constituye el sistema *Quilt* desarrollado por *Bell* [Fish88].

e) Sistemas workflow

Estos sistemas se encargan de dividir los procesos en una secuencia ordenada de tareas, encaminan el flujo de información de una tarea a otra y determinan el papel que deben tomar las personas involucradas en cada tarea. El sistema *Action Workflow System* de *Action Technologies* es un ejemplo característico de este grupo [Cummings94].

f) Sistemas generales de compartición de información

Bajo este epígrafe se incluyen sistemas que engloban varias de las funcionalidades descritas anteriormente. Entre ellos destaca *Lotus Notes* [Lotus98], que es una potente herramienta que aporta una gestión documental, un planificador de tareas o un sistema de mensajería. En general, proporciona una solución completa para

permitir organizar y compartir la información que se genera en una organización. En este grupo también se sitúa *Microsoft Exchange* [Microsoft98], que integra el correo electrónico con diversos productos de *Microsoft*. Ambas herramientas se centran fundamentalmente en la cooperación asíncrona que permite a los usuarios acceder a la información en diferentes momentos.

3.6 Sistemas de Conferencia de Documentos en Tiempo Real

Los Sistemas de Conferencia de Documentos en Tiempo Real son un tipo específico, dentro de la clasificación funcional, del grupo de Sistemas de Conferencia en Tiempo Real. Abordan la cooperación simultánea sobre los mismos documentos. Su definición es la siguiente:

los Sistemas de Conferencia de Documentos en Tiempo Real permiten a un grupo de usuarios, que pueden estar en diferentes localizaciones físicas, interactuar simultáneamente a través del uso de las mismas aplicaciones informáticas.

Por medio de la utilización compartida de estas aplicaciones se puede crear, visualizar o modificar un documento, se puede editar conjuntamente un gráfico o se puede discutir y tratar de corregir en grupo un código fuente que presenta errores.

En la actualidad es habitual que estos productos se integren en los denominados *Sistemas de Conferencias de Sobremesa (Desktop Conferencing Systems)*, que proporcionan una comunicación de vídeo, audio y documentos a través de diferentes redes [Ellis91]. Algunos ejemplos de estos productos son: *ProShare Video System* de Intel, *PCS 100* de *PictureTel*, *Person to Person* de IBM, *Desktop Conferencing* de Fujitsu, *Communique!* de Insoft y *Vistium Personal Video* de AT&T [Gold94] [Labriola95] [Taylor95].

Dentro de los Sistemas de Conferencia de Documentos en Tiempo Real existe un primer nivel de clasificación que distingue entre los sistemas de cooperación consciente y los sistemas de cooperación transparente, como se observa en la Figura 3.5.

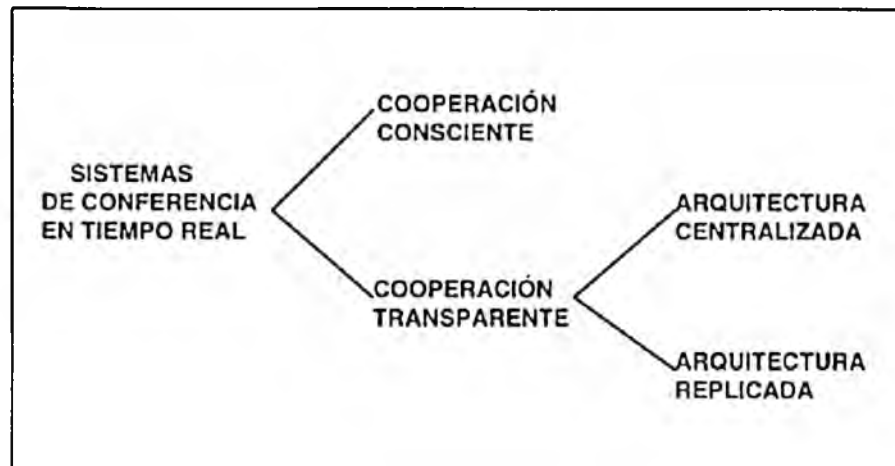


Figura 3.5. Clasificación de los Sistemas de Conferencia en Tiempo Real.

1. *Sistemas de cooperación consciente.*

Se trata de aquellos sistemas que utilizan aplicaciones diseñadas específicamente para soportar la cooperación entre un grupo de personas. Los usuarios son conscientes de que están trabajando conjuntamente con otras personas porque están usando una aplicación diseñada específicamente para realizar dicha tarea que no utilizarían nunca en el trabajo individual. Los ejemplos más comunes son aquellas aplicaciones que permiten la edición simultánea de un documento o un gráfico como el sistema *GROVE* [Ellis91].

2. *Sistemas de cooperación transparente.*

Son aquellos sistemas que hacen uso de las numerosas aplicaciones existentes mono-usuario. Por medio de un módulo externo, estas aplicaciones pueden ser utilizadas simultáneamente por varias personas. Se denominan de cooperación transparente porque el usuario no encuentra sustanciales diferencias entre

trabajar de forma individual o colectiva ya que en ambos casos se dispone de la misma interfaz y prácticamente la misma funcionalidad. Esta circunstancia supone una de las principales ventajas ya que los usuarios no tendrán que aprender a utilizar nuevas aplicaciones para trabajar cooperativamente. Por el contrario, una importante dificultad es que las aplicaciones mono-usuario no fueron diseñadas para ser utilizadas de forma cooperativa, por lo que presentan algunas limitaciones.

Entre los sistemas de cooperación transparente se establece también una división atendiendo a la arquitectura de los mismos distinguiendo entre sistemas de arquitectura centralizada y sistemas de arquitectura replicada, como se observa en la Figura 3.5.

1. *Sistemas de Arquitectura Centralizada.*

En estos sistemas se ejecuta sólo una copia de la aplicación cooperativa. Los eventos de entrada generados por cualquier usuario son enviados al nodo donde se encuentra la aplicación. Este nodo genera la salida la cual se distribuye al resto de usuarios (Figura 3.6). La cooperación se basa en la distribución de las salidas. Estas salidas son las ventanas que reflejan el estado de la aplicación en cada momento. Por este motivo, estos sistemas también son conocidos como *Sistemas de Compartición de Ventanas.*

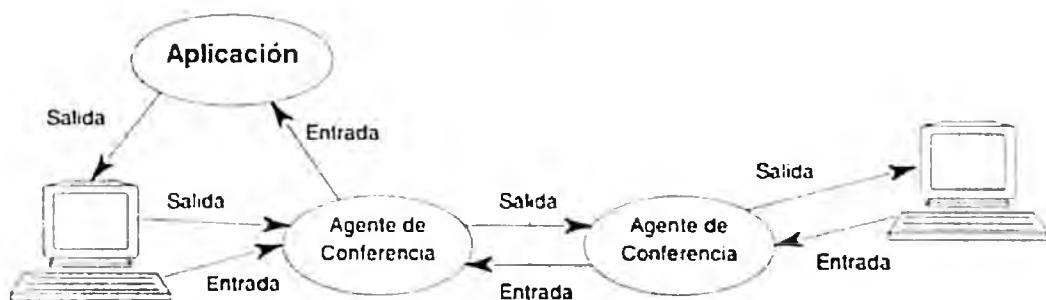


Figura 3.6. Sistema de Arquitectura Centralizada.

La principal ventaja de este modelo es su consistencia, ya que al distribuirse constantemente las ventanas que reflejan el estado de la aplicación se garantiza que todos los usuarios visualizan siempre la misma información. Pero, a su vez, presentan importantes inconvenientes como el alto tráfico de red que generan y el tiempo prolongado de respuesta que presentan.

En estos momentos existen bastantes aplicaciones comerciales que siguen este modelo. Algunos ejemplos son: *FarSite* de *DataBeam*, *ProShare* de *Intel*, *TalkShow* de *Future Lab* y *Smart 2000* de *Smart Technologies* [Labriola95]. En la actualidad destaca el producto *NetMeeting* de *Microsoft* [Byte97b]. En algunos casos estas aplicaciones se presentan como productos independientes y en otros se integran en sistemas de videoconferencia de sobremesa.

2. Sistemas de Arquitectura Replicada.

Estos sistemas se caracterizan porque se ejecuta una copia de la aplicación en cada nodo que participa en la conferencia. Los eventos de entrada generados en cualquier nodo son distribuidos al resto de usuarios participantes. Las salidas son reproducidas en cada nodo a partir de las entradas recibidas (Figura 3.7). Este modelo basa la cooperación en la transmisión de las entradas y el procesamiento de las mismas en cada nodo.

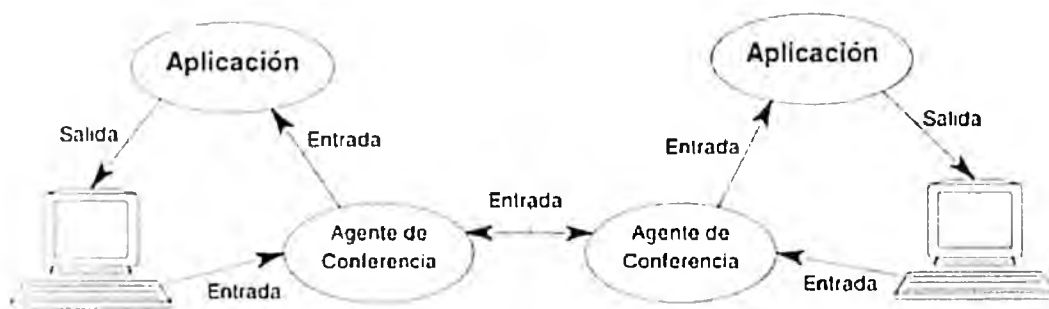


Figura 3.7. Sistema de Arquitectura Replicada.

La principal ventaja de estos sistemas comparados con los anteriores es el reducido tráfico de red que generan. También el tiempo de respuesta es menor ya que la salida se obtiene localmente. Por otra parte, resulta difícil mantener la consistencia ya que hay que garantizar que la secuencia de entradas que se procesan en cada nodo produce siempre las mismas salidas. Este hecho presenta serias dificultades. Además el tratamiento erróneo de un solo evento provoca ya que esa aplicación no mantenga el mismo estado que el resto, por lo que la cooperación se puede ver seriamente afectada.

Estas dificultades son las razones por las que estos sistemas han sido desarrollados hasta la fecha por equipos de investigación y no han pasado todavía el umbral de la comercialización. Algunos de los proyectos que han desarrollado este tipo de sistemas son: *MMConf* [Crowley90], *VConf* [Lantz86], *Dialogo* [Lauwers90], *Rapport* [Ahuja90] y *MERMAID* [Watabe90].

Existe un continuo debate dentro de esta área sobre la viabilidad de los sistemas replicados. Algunos autores se decantan de manera preferente por los sistemas centralizados por su consistencia [Ahuja90] [Patterson90], pero otros consideran necesario seguir profundizando en el estudio de los sistemas replicados para poder aprovechar las importantes ventajas que presentan [Lauwers90] [Greenberg94].

3.7 El estándar T.120

En la actualidad, algunas herramientas de Trabajo Cooperativo están reguladas por un conjunto de estándares agrupados bajo las siglas T.120. Las recomendaciones de la serie T.120 definen un conjunto de servicios y protocolos de comunicación de datos dirigidos a los entornos de conferencia multimedia. El estándar T.120 fue aprobado en Julio de 1996 por el *International Telecommunication Union (ITU)*, la agencia

americana especializada en el establecimiento de estándares de telecomunicaciones [ITU96].

Esta serie incluye un conjunto de recomendaciones comprendidas entre la T.121 y la T.127, cada una de ellas abarca algún aspecto de la conferencia de datos. En la Tabla 3.1 se especifican cada uno de los protocolos de la serie. Como se observa algunas normas sólo son orientativas y no es necesario seguirlas para adquirir la compatibilidad general con el estándar T.120.

Ref.	Fecha	Obligato- riedad	Nombre completo
T.121	1996	No	<i>Plantilla de aplicación genérica.</i>
T.122	1993	Sí	<i>Servicio de comunicación multipunto para la definición de los servicios de conferencia audiográfica y de conferencia audiovisual.</i>
T.123	1994	Sí	<i>Pilas de protocolos para aplicaciones de teleconferencias audiográficas y audiovisuales.</i>
T.124	1995	Sí	<i>Control genérico de conferencia.</i>
T.125	1994	Sí	<i>Especificación de protocolo del servicio de comunicación multipunto.</i>
T.126	1995	No	<i>Protocolo para imágenes fijas y anotaciones multipunto.</i>
T.127	1995	No	<i>Protocolo de transferencia multipunto de ficheros binarios.</i>

Tabla 3.1. Recomendaciones de la Serie T.120.

Este estándar garantiza independencia respecto de la red y de la plataforma utilizada [Davis95]. Asimismo, permite que puedan participar dos o más usuarios y que se lleven a cabo varias conferencias simultáneas en un mismo nodo.

En la actualidad, este estándar incluye las pizarras electrónicas, la transferencia de ficheros y el intercambio de imágenes estáticas. Las pizarras electrónicas son aplicaciones específicas conscientes de la cooperación que simulan el comportamiento

de una pizarra tradicional. Permiten a los usuarios realizar anotaciones, escribir texto o incorporar imágenes de forma simultánea sobre una ventana.

En Mayo de 1997 todavía no se había incluido el protocolo que abarcaba la compartición de aplicaciones. Esta recomendación está en fase de desarrollo y tomará el nombre de T.128. El borrador de Marzo de 1997 [Romano97] presenta un estado muy avanzado por lo que será aprobado definitivamente en breve plazo. Esta recomendación sólo incluye el modelo de arquitectura centralizada.

4. Estudio de los Sistemas de Arquitectura Replicada

4.1 Introducción

La adopción de un modelo basado en la replicación de las aplicaciones puede mejorar significativamente el rendimiento del sistema, como inicialmente sugirió Sarin en sus trabajos a mediados de los años 80 [Sarin84] [Sarin88]. Esta razón ha sido y continúa siendo hoy la principal motivación para profundizar en este tipo de arquitectura.

Los primeros sistemas aparecieron a finales de los 80. En 1986, Lauwers y Lantz presentaron el sistema *VConf* creado en la Universidad de Stanford [Lantz86]. Posteriormente los mismos autores desarrollaron el sistema *Dialogo* en el centro de investigación de *Olivetti* en California [Lauwers90]. Paralelamente la empresa *BBN* (*Bolt Beranek and Newman Inc.*) de Massachusetts creó el sistema *MMConf* [Crowley90] y *Xerox PARC*, dentro del proyecto *Colab*, también adoptó este modelo en alguna de sus herramientas. Por su parte, los laboratorios *AT&T Bell* desarrollaron diferentes versiones del sistema *Rapport* siguiendo inicialmente el modelo centralizado y posteriormente el modelo replicado [Ahuja90]. Asimismo, en Japón *NEC* creó el sistema de arquitectura replicada *MERMAID* [Watabe90].

El principal problema que presentan estos sistemas es el mantenimiento de la sincronización de todos los nodos que participan en una conferencia o sesión cooperativa. La cooperación se basa en la transmisión de los eventos de entrada. Por tanto, una alteración del orden de estos eventos o la incorrecta transmisión de uno de ellos provoca que a partir de ese momento el estado de las aplicaciones cooperativas no sea equivalente, y en consecuencia se interrumpa la cooperación. Este problema es sin duda el inconveniente y al mismo tiempo el reto de estos sistemas. En la actualidad, el objetivo de las investigaciones debe ser la consecución de sistemas suficientemente robustos que garanticen el mantenimiento del sincronismo y que aprovechen las ventajas potenciales que presenta esta arquitectura como ha sido apuntado por los autores que han realizado estudios en este campo [Lauwers90] [Crowley90].

En general, los sistemas desarrollados según esta arquitectura están ideados para hacer uso de aplicaciones existentes. Por tanto, una característica importante consiste en la posibilidad de poder trabajar con estas aplicaciones de la misma forma en la que se utilizan individualmente. Esta circunstancia no va a ser posible en algunos casos, como se verá posteriormente, ya que va a resultar imprescindible modificar ligeramente la funcionalidad para poder mantener el sincronismo.

4.2 Algunas definiciones

A continuación se describen algunos términos que van a ser utilizados en la descripción de los sistemas replicados.

- **Sesión Cooperativa o Conferencia:** proceso por el que varios usuarios desde diferentes lugares trabajan de forma simultánea sobre las mismas aplicaciones y documentos.
- **Aplicaciones Cooperativas:** aquellas que se utilizan en una Sesión Cooperativa.
- **Moderador:** usuario que inicia la Sesión Cooperativa.

- **Participantes:** usuarios que participan en una Sesión Cooperativa, exceptuando el Moderador.
- **Testigo:** elemento cuya posesión determina el usuario que puede escribir sobre las aplicaciones cooperativas.
- **Usuario Activo:** usuario que en un momento dado posee el Testigo, lo que le posibilita escribir sobre las aplicaciones cooperativas.
- **Usuario Pasivo:** usuario que no posee el Testigo, y por tanto, no puede actuar sobre las aplicaciones cooperativas.
- **Consistencia:** (*duración, estabilidad; firmeza, solidez*) en el ámbito de los sistemas replicados hace referencia al mantenimiento del mismo estado de las aplicaciones cooperativas en todos los nodos que participan en la sesión.
- **Sincronismo:** (*circunstancia de ocurrir, acaecer, o efectuarse dos o más cosas a un tiempo*) en el ámbito de los sistemas replicados indica el procesamiento simultáneo de los eventos en los nodos que participan en la sesión.

Con frecuencia la pérdida de la consistencia se origina por una pérdida del sincronismo, por lo que ambos términos van a estar muy relacionados en este trabajo.

4.3 Descripción general

Siguiendo la terminología propuesta por Lauwers, un sistema replicado está formado por un Agente de Conferencia que se intercala entre el sistema de ventanas y las aplicaciones de usuario, tal como se observa en la Figura 4.1 [Lauwers90]. Este modelo implica que el sistema replicado es de propósito general y, por tanto, independiente de las aplicaciones que se utilizan en una conferencia.

La función principal del Agente de Conferencia es la gestión de toda la sesión cooperativa, que comprende acciones como el establecimiento de la sesión, la gestión del espacio cooperativo, la distribución de los eventos cooperativos o la reproducción

local de los mismos en cada uno de los nodos que participa en la conferencia. El Agente de Conferencia del nodo donde se producen los eventos funciona como un multiplexor cuya salida se distribuye a cada uno de los nodos que participa en la sesión.

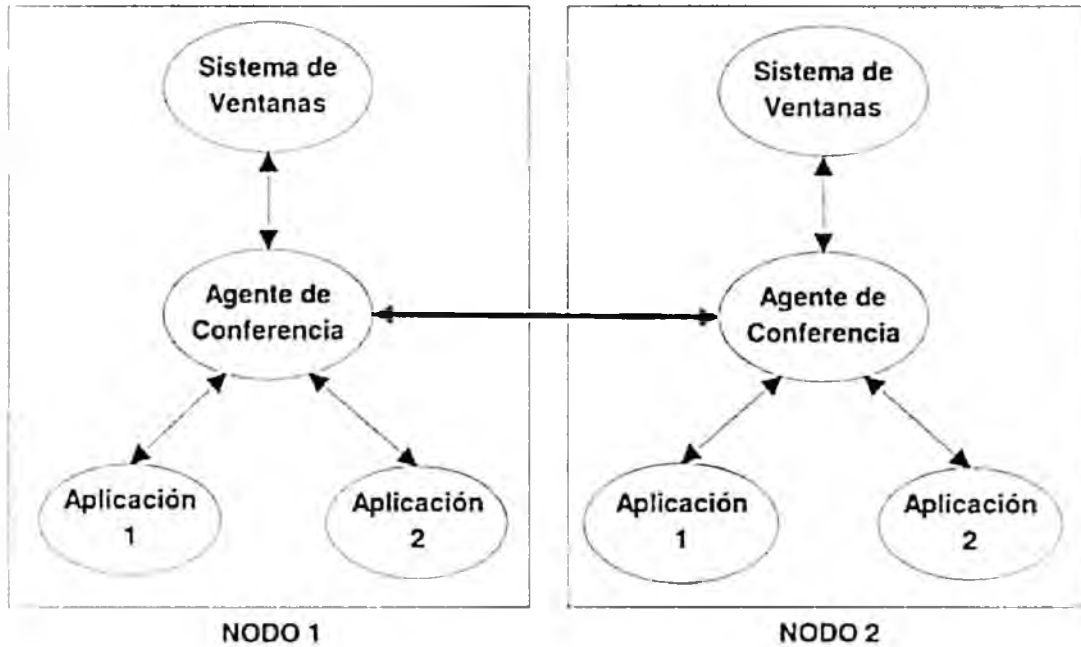


Figura 4.1. Esquema de un Sistema Replicado.

Los estudios realizados hasta el momento sobre la arquitectura replicada se han basado en prototipos desarrollados sobre el entorno *UNIX* [Kernighan84] y principalmente para el entorno de ventanas *X-Window* [Scheifler90]. En esta tesis se detallará posteriormente un sistema desarrollado sobre *MS-Windows*, pero por el momento la mayoría de las consideraciones presentadas son comunes a todos los sistemas, independientemente del entorno de ventanas utilizado. A continuación se describen las acciones que lleva a cabo el Agente de Conferencia.

4.3.1 Gestión del testigo

La gestión del testigo es un área de continuo debate en los estudios de los sistemas de cooperación en tiempo real. La posesión del testigo permite a un usuario generar entradas sobre las aplicaciones cooperativas en un momento dado. Los usuarios que tienen esta capacidad se denominan usuarios activos, mientras que el resto de usuarios que no pueden generar entradas sobre las aplicaciones cooperativas se denominan usuarios pasivos.

Las políticas de gestión del testigo definen quién puede poseer el testigo y cómo éste es intercambiado entre los participantes de una sesión. Estos métodos varían desde permitir sólo a una persona controlar todo el espacio cooperativo hasta permitir a todos los usuarios generar entradas de forma simultánea.

Como indica Lauwers, una política de gestión del testigo se define en tres aspectos [Lauwers90b]:

- El número de testigos, que puede variar desde uno para toda la sesión, uno por aplicación o uno por ventana.
- El número de personas que pueden poseer el testigo en un mismo instante.
- La forma de intercambiar el testigo entre los usuarios.

Como ya se ha indicado anteriormente, los sistemas de arquitectura replicada utilizan aplicaciones mono-usuario ya existentes. Por ello, estos sistemas únicamente contarán con un testigo, que permitirá a un solo usuario generar entradas en cada instante. Si existieran varios testigos, la cooperación resultaría muy difícil, ya que, por ejemplo, los usuarios podrían estar intentando desplazarse a una determinada posición de un documento sin conseguirlo, porque sus acciones serían contrarrestadas por las acciones de otro usuario.

Asimismo, si existieran varios testigos, los eventos se podrían procesar en diferente orden en cada nodo. Por ejemplo, si se generan varios eventos

simultáneamente, cada nodo podría reproducir primero su evento local y a continuación el resto de eventos en su orden de llegada [Crowley90].

Las políticas de intercambio del testigo más habituales son las siguientes [Reinhard94] [Crowley90] [Abe93] [Watabe90]:

1. Política dirigida por el moderador. El usuario que ha iniciado la sesión cooperativa toma el papel de moderador de la sesión y decide en cada momento quién es el usuario que posee el testigo.
2. Política dirigida por el usuario activo. El usuario activo, que posee en ese momento el testigo, determina quien es el nuevo poseedor del testigo.
3. Política con cola de peticiones. Los usuarios que desean poseer el testigo realizan una petición que es almacenada en una cola ordenada por el tiempo en que se produjo la petición o por unas prioridades asignadas a cada usuario. Cuando el usuario activo se desprende del testigo, éste pasa al primer usuario de la cola.
4. Política libre. Cada usuario puede tomar el testigo en cualquier momento. Crowley realiza una distinción entre petición implícita y explícita. En la primera un usuario realiza una acción sobre una aplicación cooperativa y si no posee el testigo, se le asigna inmediatamente y la acción es reproducida. Por tanto la asignación del testigo es totalmente transparente al usuario, pero presenta el inconveniente de que esa primera acción se va a reproducir con un retraso que puede provocar que el usuario intente repetir la citada operación. En cambio, en la petición explícita el usuario debe realizar una determinada acción para tomar el testigo y una vez lo haya tomado puede generar entradas sobre las aplicaciones cooperativas. Este segundo modelo evita el problema del retardo. La política libre de intercambio del testigo es únicamente aplicable de forma eficaz cuando existe un medio auxiliar de comunicación como el audio, a

través del cual se realizan las peticiones y las asignaciones del testigo. Si no existiese este medio auxiliar, los usuarios podrían entrar en una lucha constante por la posesión del testigo que impediría una cooperación eficaz [Crowley90].

La introducción de sistemas informáticos para dar soporte a estos procesos de cooperación puede mejorar los tradicionales protocolos sociales de petición del testigo o de petición de la palabra que aparecen en cualquier reunión. Por ejemplo, se puede asegurar que todas las peticiones van a ser recogidas, se pueden ordenar esas peticiones según diversas prioridades, se puede garantizar que todas las peticiones van a ser atendidas controlando el tiempo de posesión del testigo o cediéndolo cuando no haya actividad [Lauwers90b].

La observación del desarrollo de las reuniones de trabajo indica que no se puede establecer una política general de intercambio de testigo óptima y que en cada momento resulta apropiada una política determinada. Por ejemplo, si se quiere realizar una *lluvia de ideas (brainstorming)* resulta conveniente una política libre en la que cada usuario pueda aportar sus ideas en cualquier momento. Pero esta política resulta inaceptable en otras situaciones, especialmente cuando hay un gran número de usuarios participando en la sesión y no se cuenta con un medio auxiliar de audio o cuando el usuario activo está realizando una acción crítica durante la cual no debe ser desposeído del testigo.

Por tanto, lo más apropiado es contar con una selección dinámica de la política de gestión del testigo que permita en cualquier momento de la sesión seleccionar la política deseada [Lauwers90b].

4.3.2 Gestión del espacio cooperativo

Este módulo se encarga de identificar y controlar las ventanas que forman parte de una conferencia de documentos. Este control incluye la posición, el tamaño y el contenido de las ventanas y debe facilitar en la medida de lo posible la cooperación.

Como indica Lauwers los tradicionales administradores de ventanas ofrecen una información limitada para gestionar el espacio cooperativo. Por ejemplo, un gestor del espacio cooperativo debería poder suministrar la siguiente información [Lauwers90b]:

- Una diferenciación clara entre las ventanas cooperativas y las ventanas locales.
- La identificación de las ventanas que forman parte de una conferencia.
- La identificación de la conferencia a la que pertenece una determinada ventana cooperativa, especialmente si existen varias conferencias paralelas.

Alguna de estas informaciones se puede facilitar mediante señas visuales. Por ejemplo, mediante el empleo de diferentes colores en los bordes de las ventanas locales y las ventanas cooperativas.

La gestión del espacio cooperativo también está relacionada con el modo de cooperación visual. En algunos sistemas, todas las ventanas cooperativas presentan exactamente la misma apariencia, la misma posición, el mismo tamaño y el mismo contenido. Esta forma de cooperación visual se denomina *WYSIWIS estricto* (*What You See Is What I See, lo que tu ves es lo que yo veo*) [Stefik87]. Este modelo fortalece el concepto de entorno de trabajo compartido ya que todos los usuarios están viendo la misma información. En cambio, presenta el inconveniente de su rigidez y la limitación de contar sólo con un testigo cooperativo. Con este modelo, no es posible que dos usuarios generen simultáneamente entradas cooperativas en la misma conferencia.

El sistema *Dialogo* sigue este modelo *WYSIWIS estricto*. Se implementa mediante la ejecución de un administrador de ventanas como una aplicación cooperativa más, que garantiza que las ventanas cooperativas mantengan el mismo estado en todos los nodos [Lauwers90b].

Otros sistemas permiten un mayor control de las aplicaciones cooperativas en cada nodo y adoptan el modelo *WYSIWIS relajado* [Stefik87]. Esta *relajación* permite

que existan variaciones en la disposición externa de la ventana, en su tamaño, posición, o apariencia.

El sistema *MMConf* presenta un modelo *WYSIWIS relajado* en la posición de las ventanas cooperativas, las cuales pueden situarse en diferentes posiciones en cada nodo. En cambio, todas las ventanas cambian de tamaño simultáneamente para garantizar, de forma sencilla, la consistencia. Si el tamaño de las ventanas fuese diferente, la manipulación de la barra de desplazamiento provocaría variaciones desiguales en el contenido de las ventanas de los distintos nodos [Crowley90].

El sistema *MERMAID* ofrece un mayor nivel de *relajación* en la visualización y permite a cada usuario variar de forma independiente la posición y el tamaño de las ventanas cooperativas [Watabe90].

Todos los sistemas anteriores pertenecen al grupo de cooperación transparente, que se basan en la utilización de aplicaciones mono-usuario, y que fueron descritos en el apartado 3.6. Este tipo de sistemas está estrechamente ligado al modelo de cooperación visual *WYSIWIS* y el nivel de relajación permitido no incluye en ningún caso los contenidos de las ventanas cooperativas [Reinhard94].

Los sistemas de cooperación consciente, descritos en el apartado 3.6, sí permiten una variación de los contenidos de las ventanas, lo que posibilita que varios usuarios puedan generar entradas simultáneamente a través de la existencia de varios testigos. Esta posibilidad puede ser importante y necesaria en algunas circunstancias, pero también provoca una pérdida en la sensación de cooperación, ya que el usuario es menos consciente del trabajo que están realizando el resto de participantes [Lauwers90b]. Una *relajación* en los contenidos la ofrece el sistema *GROVE* que permite a los usuarios visualizar diferente información en las ventanas cooperativas [Ellis91].

4.3.3 Establecimiento de la sesión cooperativa

Este módulo recoge las acciones de inicialización y finalización de la sesión cooperativa. La forma concreta de implementar estos módulos depende en gran medida del sistema de ventanas de los equipos utilizados: *X-Window*, *NeWS*, *MS-Windows*, etc.

En general, los sistemas deben proporcionar mecanismos poco rígidos que permitan establecer la cooperación de forma sencilla e inmediata, sin necesidad de realizar un proceso complejo. Por ejemplo, sería deseable que un usuario pudiera seleccionar una aplicación con la que desea cooperar simplemente pulsando el ratón sobre ella.

Asimismo, estos sistemas deben admitir una reconfiguración dinámica de la sesión cooperativa, permitiendo que nuevos usuarios se incorporen a una sesión ya iniciada. Esta posibilidad presenta importantes dificultades en los sistemas de arquitectura replicada ya que el nuevo usuario debe reproducir todos los eventos generados hasta ese momento desde el inicio de la sesión para poder presentar las aplicaciones cooperativas en el mismo estado que en el resto de nodos que ya participaban en la sesión [Lauwers90b] [Crowley90]. La reconfiguración dinámica también debe posibilitar que se añadan nuevas aplicaciones y documentos a una sesión ya iniciada [Chung93].

4.3.4 Anotaciones y señalización

Sistemas como *Dialogo*, *MMConf* o *Rapport* incluyen funciones para señalar diferentes objetos o para realizar anotaciones gráficas (líneas, círculos, cuadrados, etc.) sobre las ventanas cooperativas. Dichas funciones se realizan mediante mecanismos un tanto complejos cuya fiabilidad es cuestionada en algunos casos [Lauwers90b]. Estas tareas se podrían llevar a cabo de una forma más sencilla si los sistemas de ventanas soportaran ventanas transparentes las cuales se situarían encima de las aplicaciones cooperativas y sobre las que se podrían realizar libremente las anotaciones. Este modelo

simularía la acción de poner una transparencia encima de un documento impreso para señalar parte del mismo sin alterar el propio documento. En la actualidad, los sistemas disponibles no soportan ventanas transparentes [Crowley90].

Otra característica importante de estos sistemas es la posibilidad de proporcionar múltiples punteros. Según Crowley, la opción más acertada es la inclusión de dos punteros: el puntero cooperativo y otro propio para cada usuario. El sistema *MMConf* sigue este modelo [Crowley90]. La existencia de un único puntero limita bastante el trabajo privado sobre aplicaciones no cooperativas. En este caso, las acciones privadas no se pueden realizar mientras haya algún usuario que esté generando eventos cooperativos. Por otra parte, en el sistema *MERMAID* se ha optado por incluir un puntero por cada uno de los usuarios [Watabe90]. Esta circunstancia puede resultar confusa, si el número de usuarios participantes es elevado.

4.4 Ventajas

Las principales ventajas de los sistemas replicados están relacionadas con el rendimiento que ofrecen. A continuación se detallan estas ventajas tomando como referencia los sistemas centralizados [Lauwers90].

a) Menor tráfico de red.

Los sistemas replicados generan un menor tráfico de red ya que únicamente son distribuidas las entradas entre los nodos participantes, mientras que en los sistemas centralizados es necesario transmitir las entradas y las salidas [Ahuja90]. Las entradas corresponden generalmente a los eventos provenientes del teclado y del ratón. Las salidas están formadas por las ventanas que componen una aplicación.

En el sistema centralizado de la Figura 3.6 se observa que existen dos flechas entre ambos nodos que corresponden a las entradas y a las salidas. En cambio en el sistema replicado de la Figura 3.7 sólo existe una flecha que corresponde a las entradas.

Además de contar con un único flujo de datos, el provocado por las entradas, la reducción del tráfico se favorece porque el volumen de los datos de salida es en general considerablemente mayor que el volumen de datos generados por las entradas [Lauwers90]. Esta diferencia se suele incrementar en las aplicaciones que manipulan gráficos, ya que generan un gran volumen de salida.

b) Menor tiempo de respuesta.

Los sistemas replicados presentan un menor tiempo de respuesta que los sistemas centralizados ya que las salidas son generadas localmente en cada nodo que participa en la sesión cooperativa.

En una arquitectura centralizada, únicamente se generan localmente las salidas en el nodo que ejecuta la aplicación. El resto de nodos deben esperar la transmisión de las salidas a través de la red [Lauwers90].

En una arquitectura replicada, los usuarios participantes deben esperar a recibir las entradas para poder procesar las salidas localmente. Este retraso no será muy significativo. Dependerá de la propia red y del volumen de las entradas que es relativamente pequeño.

El tiempo de respuesta es especialmente crítico cuando el usuario posee el testigo, ya que dicho usuario es consciente de las entradas que ha generado y por tanto espera ver la respuesta lo antes posible. Este retraso es un factor muy negativo en los sistemas centralizados cuando el nodo activo no es el nodo que ha lanzado la aplicación. En este caso, la secuencia de transmisión de datos desde la generación de la entrada hasta la visualización de la salida se prolonga excesivamente. En algunas ocasiones, este retraso puede provocar que el usuario repita la acción realizada al no ver su resultado inmediato lo que entorpece considerablemente el trabajo con dichas aplicaciones

En la Figura 4.2, se observa la secuencia de un evento cooperativo en una arquitectura centralizada cuando el nodo activo, que genera la entrada, no coincide con

el nodo que contiene la única copia de la aplicación que se ejecuta. El evento de entrada se genera en el nodo activo (1) y se transmite al nodo que contiene la aplicación (2). En este nodo se procesa (3) y se visualiza la salida (4). Esta salida se captura (5) y se envía al nodo que originó la entrada (6) y al resto de participantes. En cada nodo se visualiza el resultado por la pantalla (7). Se observa como el resultado de la entrada (1), sólo se puede visualizar tras todas estas etapas que incluyen dos transferencias de datos entre los nodos (2) y (6). Un usuario puede generar una entrada y por tanto ser consciente de ella, pero debe esperar a que se realice todo el proceso anterior para poder ver el resultado.

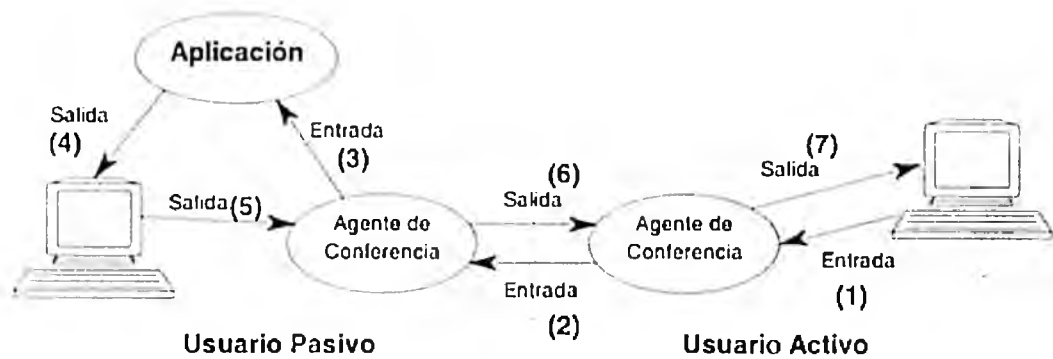


Figura 4.2. Secuencia de un evento en una Arquitectura Centralizada.

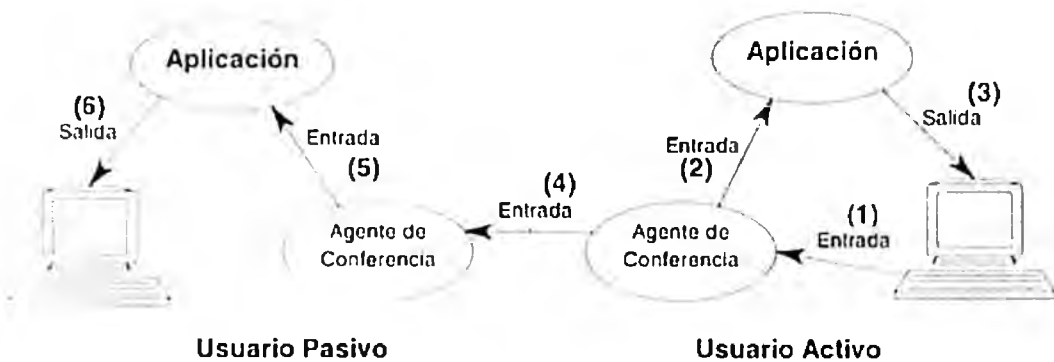


Figura 4.3. Secuencia de un evento en una Arquitectura Replicada.

En la arquitectura replicada, no existe este retraso. Como se observa en la Figura 4.3, en el nodo del usuario activo la entrada (1) se procesa inmediatamente (2) y se obtiene la salida (3). Los usuarios pasivos reciben dicha entrada (4), la procesan (5) y se obtiene la salida (6). Se observa que el proceso es más sencillo y sólo existe una transferencia de datos entre los nodos (4). Además, tanto en el nodo activo como en el pasivo la salida siempre se genera localmente.

c) Mejor adaptación a las variaciones de latencia

Los sistemas replicados no dependen tanto de las variaciones de latencia de la red como lo hacen los sistemas centralizados [Crowley90] [Greenberg94]. La latencia mide el tiempo que permanece un paquete de datos en la red [Acampora94].

Como se observa en la Figura 4.2, en los sistemas centralizados una cooperación eficaz depende directamente del tiempo que tarde el paquete de entrada (2) en recorrer la red desde el nodo activo hasta el nodo que ejecuta la aplicación y del tiempo que tarde el paquete de salida (6) en ir desde el nodo de la aplicación hasta el nodo activo. Por tanto en redes lentas, en las que este tiempo es elevado, el comportamiento de un sistema centralizado será muy poco eficiente.

En los sistemas replicados, como se refleja en la Figura 4.3, únicamente se transmiten las entradas (4), que generalmente implican un volumen reducido. Por tanto, estos sistemas se adaptan con menos problemas a aquellas redes lentas que cuentan con una latencia alta.

4.5 Desventajas

El principal problema de los sistemas basados en una arquitectura replicada es el mantenimiento del mismo estado en todas las aplicaciones cooperativas. Se debe

garantizar en todo momento el sincronismo de la transmisión y la reproducción de los eventos en los nodos que participan en una sesión. Si ocurre cualquier circunstancia que provoca diferentes estados de las aplicaciones cooperativas, se perderá el sincronismo y se interrumpirá la cooperación. A partir de ese momento no se puede garantizar que las mismas entradas vayan a producir idénticas salidas en todos los nodos y por lo tanto no se puede garantizar que los participantes en la sesión vean y manipulen la misma información.

Lauwers analiza este problema de la pérdida de la cooperación afirmando que los sistemas replicados deben mantener una consistencia en las entradas, en las salidas y en el estado inicial. Todas las copias que participan en una conferencia deben iniciarse en el mismo estado y deben recibir entradas equivalentes. Lauwers introduce el término *equivalentes* en lugar de *idénticas* ya que las entradas pueden ser adaptadas en cada nodo a su propio entorno sin por ello perder la sincronización [Lauwers90].

El mantenimiento de la consistencia o del sincronismo es por tanto el principal reto con que se encuentran estos sistemas. En algunas situaciones se introducen mecanismos especiales para garantizar este sincronismo, pero en otras no se puede ofrecer una solución totalmente satisfactoria.

A continuación se detallan las principales dificultades de los sistemas de arquitectura replicada.

a) Consistencia en el estado inicial de la aplicación

Una primera restricción de estos sistemas es la necesidad de que cada nodo disponga de las aplicaciones sobre las que se va a cooperar y que todas las copias correspondan a la misma versión. Algunos sistemas como *MMConf* incluyen un mecanismo de comparación de versiones [Crowley90]. En cambio, en los sistemas centralizados únicamente el nodo que inicia la sesión necesita contar con las aplicaciones cooperativas.

Asimismo, hay que considerar que cada copia de la aplicación cooperativa pueda tener diferentes configuraciones en cada nodo. Este hecho provoca que la aplicación se abra con diferentes estados. Esta circunstancia no se puede permitir ya que originaría la pérdida de la consistencia desde el momento inicial [Abe93]. La solución habitual consiste en tomar como configuración de referencia las opciones del nodo que inicia la sesión. Estas opciones se transfieren al resto de participantes para que todas las copias se abran con el mismo estado.

b) Indeterminismo

En general, el problema del sincronismo puede ser abordado siempre que las aplicaciones cooperativas sean deterministas. Una aplicación es determinista cuando partiendo del mismo estado inicial y la misma secuencia de eventos de entrada, se obtienen siempre las mismas salidas y el mismo estado final. Asimismo, las salidas de una aplicación determinista no dependen del momento en que se generan las entradas [Lauwers90].

Una situación indeterminista ocurre, por ejemplo, cuando un programa ejecuta el cuerpo de un bucle hasta que se produzca un determinado evento de entrada. Después de cada iteración del bucle, la aplicación consulta la cola de entrada para comprobar si ha llegado dicho evento. Entre los nodos que ejecutan ese programa no existe ninguna sincronización en la verificación de la cola y por tanto, el número de iteraciones del bucle dependerá de la capacidad del procesador de cada nodo. Esta situación se produce cuando se mantiene pulsada una flecha de una barra de desplazamiento. Tras esta acción la aplicación se habrá movido a una posición diferente dependiendo del número de mensajes que se hayan podido generar en cada nodo durante el tiempo en que se ha mantenido pulsada la flecha. De esta forma, partiendo de una misma secuencia de entrada se han obtenido diferentes salidas.

Sería una restricción muy drástica la imposibilidad de utilizar aplicaciones que generen situaciones indeterministas, por lo que la mayoría de los sistemas replicados admiten las situaciones descritas anteriormente. La solución general utilizada para

resolver el problema consiste en desactivar las acciones que provocan las situaciones indeterministas [Crowley90].

c) Gestión de los recursos externos

En cualquier sistema los mensajes no provienen únicamente de las acciones del usuario, sino que pueden estar relacionados con diferentes recursos o dispositivos como el sistema de ficheros o una impresora. El tratamiento de los eventos relacionados con estos recursos externos es una tarea difícil en los sistemas replicados ya que la secuencia de eventos de entrada puede provocar la pérdida del sincronismo y las salidas pueden provocar el acceso redundante a un mismo dispositivo, como en la impresión múltiple de un documento.

Cada nodo que participa en una sesión puede acceder a una serie de dispositivos locales o privados a los que no tienen acceso el resto de nodos, por ejemplo, su propio sistema de ficheros o una base de datos de acceso restringido. Por tanto, si se reproducen los eventos relacionados con estos dispositivos en el resto de nodos se estaría tratando de acceder a unos dispositivos que no son accesibles, lo que provocaría la pérdida del sincronismo.

El planteamiento general para tratar de resolver este problema es que los eventos relacionados con el acceso a los recursos externos sean reproducidos únicamente en el usuario activo. Además si se ha accedido a un dispositivo de entrada, como el sistema de ficheros, será necesario transmitir los datos leídos al resto de los usuarios [Abe93].

El caso más característico de los dispositivos de entrada es el acceso al sistema de ficheros. Los sistemas *Dialogo*, *MMConf* y *VConf* utilizan un mecanismo similar para abordar esta cuestión que consiste en la creación de un *Directorio de Conferencia* en cada participante que contiene los archivos que intervienen en la sesión. Cuando un archivo se deposita en este directorio, se transfiere a los directorios del resto de nodos. Esta tarea se realiza automáticamente en *Dialogo* y *MMConf*, mientras que en *VConf* se

debe realizar manualmente. Este mecanismo presenta también sus limitaciones. Únicamente permite identificar los ficheros con su nombre, pero sin incluir la localización del directorio, ya que el nombre completo sería diferente en cada nodo. Otra limitación es que no contempla la posibilidad de que un archivo pueda incluir referencias a otros archivos [Lauwers90].

Otro ejemplo aparece en el proyecto *MERMAID*, en el cual se ha diseñado una aplicación cooperativa que accede a una base de datos multimedia. Los eventos de acceso a esta base de datos sólo son procesados en el nodo activo y la información leída es distribuida al resto de usuarios [Abe93].

Por otra parte, en el tratamiento de los dispositivos de salida hay que abordar el problema del acceso redundante. Por ejemplo, las acciones de imprimir un documento, mandar un mensaje a un usuario o modificar un registro de una base de datos son acciones que es necesario que se realicen sólo una vez, y no deben ser reproducidas en cada uno de los nodos de la sesión.

Por el momento, no se ha definido ningún mecanismo general que permita resolver este problema del acceso redundante a dispositivos externos. Así Abe afirma, tras la experiencia del desarrollo del sistema *MERMAID*, que es necesario utilizar mecanismos o funciones dependientes de las aplicaciones o dispositivos a los que se intenta acceder [Abe93]. De la misma forma, *MMConf* incluye una aplicación propia de correo para evitar el envío redundante de mensajes [Lauwers90].

d) Interacción espontánea

Los sistemas replicados requieren un proceso más complejo de inicio de la cooperación que los sistemas centralizados [Lauwers90b]. En los primeros es necesario indicar previamente las aplicaciones y los archivos sobre los que se quiere trabajar. Además es necesario transferir estos archivos al resto de usuarios de la conferencia. En los sistemas centralizados esta operación se puede realizar seleccionando directamente

con el ratón las ventanas que contienen las aplicaciones y documentos sobre las que se quiere cooperar.

Pero sin duda, el problema más grave para soportar una interacción espontánea y flexible, similar a la que se da en los procesos reales de cooperación, es la posibilidad de permitir a nuevos usuarios incorporarse a una sesión ya iniciada.

En los sistemas centralizados, esta operación se reduce a transferir el estado actual de la sesión mediante la transferencia de las ventanas cooperativas. En los sistemas replicados es necesario transferir toda la historia de la sesión desde su momento inicial para que puedan ser reproducidos en el nuevo nodo y llegue a alcanzar el mismo estado que el resto de participantes. Este mecanismo, no obstante, presenta dos serios inconvenientes. El primero es el gran consumo de recursos de almacenamiento y de procesamiento que implica mantener la historia de una sesión. El segundo se debe a que el efecto de algunos eventos depende del estado general del sistema, que va variando con el tiempo. Por tanto, los resultados de la reproducción de los eventos almacenados puede no coincidir con los resultados que produjeron éstos originalmente [Lauwers90b]. Este último caso sería una situación indeterminista, ya que idénticas entradas producirían resultados diferentes.

e) Alteración del orden de reproducción de los eventos

En los sistemas replicados no sólo es necesario transferir los eventos de entrada a todos los nodos participantes, sino que se debe asegurar que en cada uno de ellos sean reproducidos exactamente en el mismo orden. Una alteración del orden de reproducción de un único evento puede ocasionar la pérdida del sincronismo y que el estado de las aplicaciones cooperativas sea diferente a partir de ese momento. La alteración en el orden de los eventos puede ser provocada por problemas en la comunicación o por una asignación errónea del orden de reproducción por parte del sistema de conferencia [Crowley90].

En algunos entornos como *X-Window* [Scheifler90] una aplicación puede estar recibiendo eventos originados por el usuario, pero también puede recibir eventos provenientes del servidor de ventanas. Estos últimos eventos, en una conferencia de documentos, son generados en cada nodo y por tanto, hay que garantizar que se entrelazan con los eventos cooperativos creados por el usuario activo exactamente en el mismo orden en todos los nodos [Lauwers90]. Para ello, en algunos sistemas como *Dialogo* se envían desde el usuario activo todos los eventos, tanto los generados por el usuario como los originados por el servidor de ventanas. Esta cadena de eventos sirve para determinar el orden de reproducción en cada nodo participante. En ellos se irán reproduciendo los eventos provenientes del usuario y se tomarán los eventos del servidor de ventanas como referencia para reproducir en ese orden el citado evento cuando sea creado en cada nodo [Lauwers90]. En el sistema *MMConf* también se toma como referencia el orden de los eventos generados en el usuario activo para determinar el orden de reproducción en el resto de nodos [Crowley90].

f) Mismo entorno de ejecución

En alguna ocasión los sistemas replicados ofrecen una cierta capacidad de adaptación a sistemas heterogéneos [Crowley90], pero en general es necesario contar en cada nodo de una conferencia con el mismo entorno software y hardware [Abe93] [Ahuja90]. Las aplicaciones que se lanzan deben ser idénticas y por tanto deben ser ejecutadas sobre el mismo entorno de ventanas.

g) Licencias en cada nodo

En la arquitectura replicada se debe contar con las aplicaciones correctamente instaladas, incluyendo su licencia, en cada uno de los nodos que van a participar en la conferencia. En los sistemas centralizados únicamente es necesario contar con una copia válida de la aplicación en el nodo que inicia la cooperación.

h) Coste de procesamiento

Otro inconveniente de los sistemas replicados consiste en la aparición de incongruencias debidas a las diferentes capacidades de procesamiento de los nodos. Cada entrada es procesada localmente por lo que pueden existir diferentes tiempos de respuesta en la obtención de las salidas. No obstante, este hecho no suele provocar la pérdida del sincronismo, ni afecta de forma determinante en la cooperación [Crowley90].

Por otra parte, en los sistemas replicados, hay que considerar que el coste de procesamiento se reproduce en cada uno de los nodos. En cambio, en los sistemas centralizados el procesamiento se realiza sólo una vez.

4.6 Consciencia de la cooperación

El concepto de *consciencia* está empezando a ser estudiado con mayor profundidad en el campo del Trabajo Cooperativo. El primer objetivo, como en otros conceptos relacionados con CSCW, es llegar a una única definición del mismo. Algunos autores definen la *consciencia* como las situaciones en las que un usuario *advierde* las acciones que están llevando a cabo otros usuarios en ese momento. Se introduce por tanto el concepto de tiempo real y de sincronismo en la comunicación. La *consciencia* de la presencia y las acciones de los usuarios se puede implementar mediante una comunicación permanente de vídeo [Fish92] o mediante la construcción de una sala de reuniones virtual [Abel90].

Otros autores amplían esta definición a la consciencia asíncrona, en la que incluyen no solo los eventos que suceden en ese momento, sino también los que han sucedido en el pasado [Fuchs95].

En los sistemas replicados, el concepto de *consciencia* aparece relacionado con los usuarios y con las aplicaciones que forman parte de una sesión cooperativa.

Un usuario es *consciente de la cooperación* por medio de la información que proporciona el propio sistema replicado, que hace que el usuario sepa que está participando en una sesión cooperativa. Un ejemplo de este tipo de información es una ventana que contenga la lista de participantes de la sesión. Asimismo, la denominada cooperación visual proporciona esta consciencia, especialmente el tipo *WYSIWIS estricto* (apartado 4.3.2) por el que todos los usuarios ven exactamente la misma información. Un usuario observará cómo se va modificando el contenido de una aplicación por acciones que él no realiza, por lo que sabrá que está participando en una conferencia.

Una aplicación es *consciente de la cooperación* cuando responde de una forma específica a una determinada acción al detectar que está formando parte de una sesión cooperativa [Crowley90]. En principio, uno de los objetivos de los sistemas replicados es hacer uso de aplicaciones existentes, sin necesidad de modificar su funcionalidad. Pero estos sistemas deben afrontar el problema del mantenimiento de la consistencia y el sincronismo durante la cooperación. Para tratar de afrontar esta cuestión resulta imprescindible introducir unos mecanismos que modifican mínimamente la funcionalidad de las aplicaciones mono-usuario y, por tanto, las hacen conscientes de la cooperación [Lauwers90].

Los siguientes mecanismos conscientes de la cooperación intentan paliar algunos de los inconvenientes descritos anteriormente [Crowley90]:

a) Mecanismos de mantenimiento del sincronismo

Se introducen este tipo de mecanismos para eliminar o modificar acciones que pueden provocar la pérdida del sincronismo. Este es el principal inconveniente de los sistemas de arquitectura replicada y se debe analizar independientemente en cada entorno de ejecución. Estos mecanismos deben abordar las situaciones indeterministas desactivando, por ejemplo, la presión continua de las barras de desplazamiento. Otro caso que también deben tratar es la acción de minimizar las ventanas. Si esta operación

se realiza libremente cada icono se podría situar en una posición diferente de la pantalla con lo que se perdería la consistencia al intentar abrir nuevamente la ventana.

b) Mecanismos de gestión de los recursos externos

Estos mecanismos deben abordar el acceso a los dispositivos de entrada y de salida. El ejemplo más característico corresponde al sistema de ficheros. Es necesario contar con un mecanismo de transferencia de archivos que distribuya entre los usuarios los archivos que inicialmente van a formar parte en la sesión o los que se incluyan durante el desarrollo de la misma. Este mecanismo es una excepción que rompe la norma general de la transmisión de los eventos de entrada, ya que en este caso se transmite el contenido de los archivos.

Sobre los dispositivos de salida, como ya se ha indicado, hay que abordar la repetición no deseada de una acción como la impresión de un documento o el envío de un mensaje. Para evitar estas duplicidades es necesario introducir unos mecanismos que impidan la realización de dichas acciones en los nodos pasivos que no poseen el testigo en ese momento. Estos mecanismos dependerán de las aplicaciones y dispositivos que intervengan.

c) Mecanismos para la actividad paralela

La arquitectura replicada puede permitir la actividad paralela de los nodos de una conferencia suprimiendo la gestión del testigo y permitiendo que todos los usuarios puedan trabajar simultáneamente sobre las aplicaciones cooperativas. Este hecho sólo es posible si el propio sistema es consciente y posee mecanismos para gestionar que una aplicación pueda recibir varias entradas de diferentes usuarios, los cuales pueden estar trabajando en diferentes puntos del documento.

4.7 Conclusiones

Muchos autores optan por los sistemas centralizados por su robustez frente a los importantes problemas de consistencia que presentan los sistemas replicados [Ahuja90] [Patterson90] [Hyland92]. No obstante, estos últimos con la inclusión de módulos conscientes de la cooperación pueden mejorar de forma muy significativa el mantenimiento de la consistencia.

Algunos autores, como Greenberg, incluso defienden que no se puede afirmar todavía cuál de los dos modelos de arquitectura, centralizada o replicada, es el más adecuado para el desarrollo de sistemas de trabajo cooperativo [Greenberg94].

El punto fuerte de los sistemas replicados respecto a los centralizados es su rendimiento [Sarin84] [Sarin88]. Esta ventaja es más significativa cuando participan aplicaciones que generan un gran volumen de información, como son las aplicaciones gráficas [Crowley90].

En la misma línea, Lauwers afirma que las ventajas que ofrecen los sistemas replicados son de la suficiente magnitud como para seguir justificando su investigación, a pesar de que se vean limitados por la tecnología actual [Lauwers90].

5. Descripción del Sistema *DocuLAN*

5.1 Introducción

En esta tesis se ha realizado un estudio de los sistemas de trabajo cooperativo basados en una arquitectura replicada y se ha llevado a cabo una comparación con aquellos que se basan en una arquitectura centralizada.

En el capítulo anterior se han analizado de forma detallada las ventajas y desventajas de cada sistema. Se observa cómo el modelo centralizado presenta una mayor robustez, lo que permite la existencia de productos comerciales que siguen este método.

En cambio, los sistemas replicados presentan grandes dificultades en el mantenimiento del sincronismo entre los nodos participantes. Pero, por otra parte, cuentan con ventajas significativas entre las que destacan el bajo tráfico de red que generan y el reducido tiempo de respuesta con el que cuentan.

Estas ventajas potenciales suponen la base y la justificación de este trabajo, en el cual se va a presentar un sistema de conferencia en tiempo real con una arquitectura replicada, denominado *DocuLAN*. A través del desarrollo de este sistema se van a

abordar las dificultades que presenta la arquitectura replicada y se va comprobar el grado de robustez que se puede alcanzar en un entorno concreto, el entorno *Microsoft Windows*.

5.2 Relación con el proyecto europeo MMTCA

El trabajo presentado en esta tesis tuvo su origen en el proyecto europeo *MMTCA: Multimedia Toolbox for Cooperative Applications*, dentro del programa *ESPRIT (European Specific Programme for Research and Development in Information Technologies)*. El consorcio que llevó a cabo el proyecto estuvo formado por las empresas griegas *Intrasoft* e *Intracom*, la Universidad de Paisley en Glasgow, *ISL* de Bremen, *Novosoft* y el Banco del Comercio de Madrid y el centro tecnológico vasco *ROBOTIKER* donde fue realizado el trabajo relacionado con esta tesis. Los resultados finales del proyecto se recogieron en el libro "*Cooperative Work with Multimedia*" [Crowe94].

El objetivo global del proyecto consistía en integrar el trabajo cooperativo síncrono y asíncrono realizado con aplicaciones multimedia. El resultado se plasmaba en una caja de herramientas con la que el usuario podía crear un flujo de procesos (*workflow*) compuesto por diferentes tareas. Estas tareas podían ser asignadas a diferentes personas y realizadas de forma asíncrona. Pero asimismo, en cualquier punto del proceso, se podía definir una tarea cooperativa síncrona que exigía la participación simultánea de varias personas.

En este punto, en la cooperación síncrona, se desarrolló un sistema basado en una arquitectura replicada. El resultado general del sistema y en especial este módulo de cooperación síncrona fue evaluado satisfactoriamente por los revisores oficiales de la Comisión Europea. Una vez finalizado el proyecto se aisló este módulo de cooperación y se siguió profundizando en el estudio de estos sistemas incluyendo nuevas mejoras que dieron lugar al sistema *DocuLAN*.

El sistema *MMTCA*, desarrollado en el proyecto europeo, seguía un modelo cliente-servidor. En el puesto del cliente se optó por el entorno *MS-Windows 3.11*, lo que permitía utilizar como aplicaciones cooperativas todas aquellas que se ejecutaran bajo este entorno. El servidor era una máquina *UNIX* desde la que se accedía a una base de datos *ORACLE*, *INFORMIX* o *INGRES* (se utilizaron estas tres bases de datos en diferentes instalaciones de los participantes en el consorcio). En esta base de datos se mantenía la información de los usuarios y la definición de los flujos de tareas definidos. El usuario activo enviaba los eventos al servidor, y este los distribuía al resto de participantes. Este modo centralizado de transmisión permitía tener un mecanismo de control de errores que posibilitaba el reenvío de los eventos que no hubieran llegado a su destino.

5.3 Entorno de desarrollo y ejecución

El sistema *DocuLAN* es, por tanto, un sistema de conferencia de documentos en tiempo real bajo el entorno *MS-Windows 3.x* que sigue un modelo de arquitectura replicada.

Con el nombre *MS-Windows 3.x* se hace referencia a los entornos *MS-Windows 3.1*, *MS-Windows 3.11* y *MS-Windows 3.11 para Trabajo en Grupo* [Microsoft93]. Respecto a este último entorno, hay que reseñar que no es necesario tener instalada la red de *Microsoft para Trabajo en Grupo* para poder ejecutar *DocuLAN*. Únicamente se necesita el kernel *TCP/IP* [Stevens94] de comunicaciones.

El sistema *DocuLAN* se ha desarrollado sobre ordenadores personales *PCs* compatibles. Partiendo de una configuración estándar, los únicos elementos adicionales necesarios para ejecutar el sistema son aquellos que proporcionan el soporte de comunicaciones. *DocuLAN* trabaja sobre el protocolo *TCP/IP* y sobre la Red Digital de Servicios Integrados (*RDSI*) [Griffiths92]. En este trabajo se ha evaluado la versión de

DocuLAN que utiliza el protocolo TCP/IP. A continuación se presenta la configuración del entorno de ejecución para esta versión.

- PC 486, 50 Mhz
- 4 MB RAM (recomendados 8 MB RAM)
- 500 MB disco duro
- MS-DOS 6.22 con *MS-Windows 3.x*
- Tarjeta de comunicaciones Ethernet [Stallings87]
- Kernel de comunicaciones TCP/IP compatible *Windows Sockets 1.1* [Hall93].

En la mayoría de los casos, los requisitos del sistema van a depender del tipo de aplicaciones con las que se quiera trabajar y no tanto de los requisitos propios de *DocuLAN*. Por ejemplo, este sistema no necesita una gran cantidad de memoria, pero puede que este requisito sea necesario para poder lanzar aplicaciones sobre las que se desea cooperar.

Durante el desarrollo y comprobación del sistema se han utilizado las siguientes las tarjetas de comunicaciones:

- Racal Interlan NI6510 [Racal90]
- SMC EtherCard PLUS [SMC93]
- Ethernet 3Com Etherlink III Bus-Master PCI [3Com95].

El sistema ha sido, asimismo, comprobado con los siguientes kernels de comunicaciones TCP/IP:

- *Trumpet Winsock* para *Windows 3.x* [Tattam94]
- *PC/TCP* de *FTP Software* [Ftp92]
- *Microsoft TCP/IP-32 para Windows para Trabajo en Grupo* [Microsoft94].

En el desarrollo del sistema se ha utilizado el compilador de *Microsoft Visual C/C++* en su versión 1.5. El lenguaje utilizado ha sido principalmente C. Algunos módulos han sido desarrollados con C++ y las librerías *Microsoft Foundation Classes (MFC)* [Kruglinski94].

Las razones que han motivado la elección de esta plataforma son las siguientes:

1. Resulta conveniente evaluar los sistemas de trabajo cooperativo en los ordenadores personales PCs al ser las plataformas más utilizadas.
2. El entorno de ventanas *MS-Windows* domina el mercado de sistemas operativos. Tomando como fuente a la consultora *Dataquest*, el 55% de los sistemas operativos vendidos en 1995 era *Windows 3.x*. A continuación aparecía *Windows 95* con un 26% [Peworld96]. El sistema se ha desarrollado con la versión 3.11, última versión disponible en el momento de iniciar el estudio. Fuera del ámbito de esta tesis se ha realizado la adaptación del sistema *DocuLAN* a *Windows 95* [Herrero96].
3. Hasta el momento, todos los sistemas de conferencia de arquitectura replicada han sido desarrollados sobre el entorno *UNIX*: *MMConf* [Crowley90], *VConf* [Lantz86], *Dialogo* [Lauwers90], *Rapport* [Ahuja90] y *MERMAID* [Watabe90]. Por tanto, es interesante explorar el comportamiento de estos sistemas sobre el entorno *MS-Windows* para detectar las ventajas y los inconvenientes que proporciona a los sistemas replicados.
4. Los protocolos de comunicaciones, TCP/IP y RDSI, se consideran estándares y son ampliamente utilizados.

5.4 Descripción general

DocuLAN se enmarca dentro de los sistemas de conferencia de documentos en tiempo real y cuenta con las siguientes características principales:

1. *DocuLAN* está construido siguiendo un modelo de arquitectura replicada por el que en cada nodo se va a lanzar la aplicación o aplicaciones sobre las que se va a cooperar.
2. El sistema va a permitir que las aplicaciones que se utilizan de forma individual en el entorno *MS-Windows 3.x* puedan ser utilizadas por un grupo de personas simultáneamente.

De esta forma, *DocuLAN*, según la clasificación descrita en el apartado 3.6, se define como un sistema de arquitectura replicada y de cooperación transparente.

Gracias a la primera característica, arquitectura replicada, se van a explorar las ventajas potenciales que este modelo conlleva, y se va a evaluar el grado de consistencia que se puede alcanzar.

Con la segunda característica, cooperación transparente, se posibilita que los usuarios puedan trabajar de forma cooperativa con las mismas aplicaciones que habitualmente utilizan de forma individual con lo que se reduce considerablemente el tiempo de aprendizaje.

El sistema consta de varios módulos independientes. Incluso es posible modificar alguno de estos módulos para adaptarlo a un entorno concreto sin alterar el resto del sistema. Por ejemplo, de esta forma se pueden utilizar diversos medios de comunicación mediante la adaptación del módulo de comunicaciones al protocolo seleccionado. A continuación se enumeran los módulos del sistema *DocuLAN* cuya relación se puede observar en la Figura 5.1.

1. Módulo de Gestión de Aplicaciones. Se encarga de identificar y controlar las aplicaciones durante la sesión cooperativa.
2. Módulo de Gestión del Testigo. Es responsable de otorgar a los usuarios el permiso correspondiente para poder trabajar sobre las aplicaciones cooperativas.

3. Módulo de Comunicaciones. Se encarga de la transmisión de los datos entre los nodos participantes en una sesión.
4. Módulo de Gestión de Eventos. Se encarga de la captación de los eventos en el nodo activo y de su reproducción en los nodos pasivos.

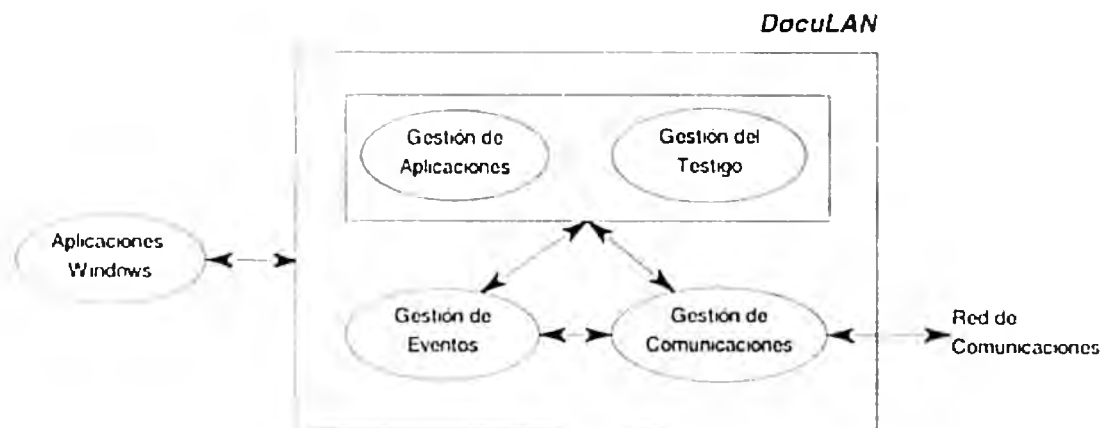


Figura 5.1. Módulos del sistema *DocuLAN*.

En la Figura 5.2 se incluye la pantalla de un usuario que está participando en una sesión cooperativa haciendo uso del sistema *DocuLAN*. En este ejemplo, las aplicaciones cooperativas son *MS-Word* y *MS-Paintbrush*. La aplicación *Bloc de notas* es una aplicación local que no forma parte de la sesión cooperativa. La aplicación *Trumpet Winsock* es el kernel de comunicaciones TCP/IP. En la parte inferior derecha aparece la ventana de control de la sesión cooperativa que permanece siempre visible durante el transcurso de la misma.

La ventana de control contiene tres botones que permiten realizar las siguientes operaciones:

1. Solicitar, ceder o tomar el testigo. Se podrá realizar una de estas operaciones dependiendo de cual sea el estado del testigo. Un usuario Participante que no posee el testigo pulsando este botón lo solicitará. Si el usuario posee el testigo

al pulsar el botón lo cederá a otro usuario. Por último, el Moderador podrá tomar directamente el testigo pulsando el botón.

2. Añadir nuevas aplicaciones y documentos a una sesión ya iniciada.
3. Abandonar la sesión cooperativa.

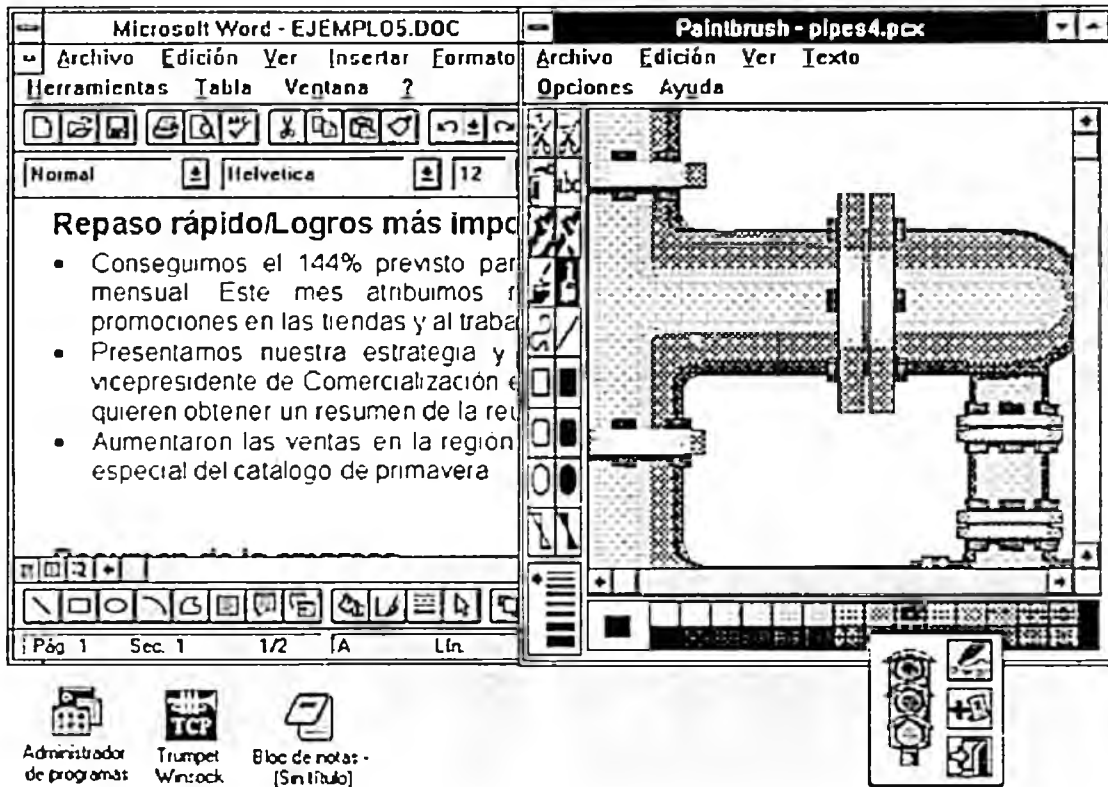


Figura 5.2. Pantalla de un usuario ejecutando *DocuLAN*.

La ventana de control contiene también un semáforo que refleja el estado del testigo. Este elemento se seleccionó por su claridad para poder reflejar la posibilidad o imposibilidad de poder actuar sobre las aplicaciones cooperativas. El semáforo puede presentar las siguientes combinaciones en sus luces:

1. Luz verde: el usuario posee el testigo y por tanto puede actuar sobre las aplicaciones cooperativas.

2. Luces verde y ámbar: el usuario posee el testigo, pero otro usuario ha realizado una solicitud del mismo.
3. Luz roja: el usuario no posee el testigo y por tanto no puede actuar sobre las aplicaciones cooperativas.
4. Luz roja y ámbar: el usuario no posee el testigo, pero ha realizado una petición del mismo.

5.5 Eventos y filtros

El sistema *DocuLAN* va a hacer uso de dos conceptos, los eventos y los filtros, que necesitan ser desarrollados de forma más amplia antes de comenzar una descripción detallada del sistema.

5.5.1 Eventos

El modelo de arquitectura replicada lleva a cabo la cooperación por medio de la transmisión de los eventos de entrada. En el entorno *MS-Windows* estos son los mensajes provenientes del teclado y del ratón. En la Tabla 5.1 se detallan los posibles eventos y su significado.

Estos eventos se almacenan en la denominada cola del sistema. Esta cola se crea cuando se inicia una sesión en el entorno *Windows*. En ella se almacenan los eventos del teclado y del ratón en forma de mensajes. Cada evento se representa por medio de una estructura que contiene la información de la Tabla 5.2. Esta estructura está compuesta por el identificador del mensaje, su tiempo de generación y dos parámetros adicionales que contienen información específica del mensaje.

Eventos:	Descripción:
WM_MOUSEMOVE	Se originan con el movimiento del ratón.
WM_LBUTTONDOWN WM_MBUTTONDOWN WM_RBUTTONDOWN	Se originan con la pulsación de los botones del ratón izquierdo, central y derecho respectivamente.
WM_LBUTTONUP WM_MBUTTONUP WM_RBUTTONUP	Se originan con la liberación de los botones del ratón izquierdo, central y derecho respectivamente.
WM_KEYDOWN WM_SYSKEYDOWN	Se originan con la pulsación de las teclas normales y las teclas del sistema (Control, Alt, Shift...) respectivamente.
WM_KEYUP WM_SYSKEYUP	Se originan con la liberación de las teclas normales y las teclas del sistema (Control, Alt, Shift...) respectivamente.

Tabla 5.1. Eventos del entorno *Windows 3.x*.

A su vez cada aplicación en ejecución cuenta con una propia cola de mensajes. El funcionamiento de una aplicación *Windows* se basa en procesar o responder los mensajes que llegan a la citada aplicación. El entorno *Windows* revisa en primer lugar las colas de las aplicaciones y si no existen mensajes para procesar o los existentes son de baja prioridad, revisa entonces la cola del sistema. Esta forma de funcionamiento se denomina multitarea preferente [Richter92].

Identificador:	Descripción:
<i>message</i>	Identificativo del mensaje. Contiene uno de los valores descritos en la Tabla 5.1.
<i>paramL</i>	Parámetro <i>Low</i> específico del mensaje.
<i>ParamH</i>	Parámetro <i>High</i> específico del mensaje.
<i>Time</i>	Tiempo en que ocurrió el evento medido en milisegundos desde el inicio de la sesión <i>Windows</i> .

Tabla 5.2. Información del evento en la cola del sistema.

En el sistema *DocuLAN*, la cooperación se lleva a cabo mediante la captación de los eventos del teclado y del ratón en el nodo que posee el testigo, su transmisión por la red y su reproducción en los nodos participantes. Esta reproducción consiste en insertar los eventos recibidos en la cola del sistema del citado nodo. Una vez en esa cola, ese mensaje será procesado de igual forma que un evento generado en el propio nodo.

Los eventos pueden originar a su vez más mensajes. En un sistema que siga una arquitectura replicada, estos eventos se generan localmente en cada uno de los nodos. Este hecho es uno de los principales inconvenientes de este modelo, ya que hay que conseguir que esos nuevos mensajes generados en cada nodo sean idénticos en todos ellos y mantengan el mismo orden. Si esto no sucede, aparecen las situaciones indeterministas. En estas situaciones, una secuencia de entradas idéntica (los eventos) provoca diferentes estados finales (los mensajes generados).

5.5.2 Filtros

La manipulación de los eventos y de los mensajes se va a realizar por medio de los filtros. Los filtros permiten a una aplicación capturar determinadas acciones que van a ser procesadas en el ámbito de la propia aplicación o en el ámbito de todo el entorno *Windows*. Esta captura consiste en poder acceder a los mensajes de la cola del sistema y de la cola de las aplicaciones inmediatamente antes de que sean procesados. Por tanto, este mecanismo permite a una aplicación ser consciente de cuáles van a ser las siguientes acciones antes de que se lleven a cabo. En algunos casos es posible modificar los parámetros de los mensajes de la cola o incluso eliminar los mensajes, lo que implicaría la no realización de la acción prevista.

Este mecanismo afecta al funcionamiento ordinario del entorno *Windows* y repercute en todas las aplicaciones abiertas en ese momento. Es una herramienta de programación especialmente poderosa que ofrece grandes posibilidades. Por lo general,

es poco utilizada debido a la escasa documentación con la que se cuenta y a su dificultad de programación [Richter92].

Dos fuentes de información recomendadas sobre los filtros son el artículo técnico *Microsoft Windows Hooks* [Marsh92] y el capítulo 7 del libro *Windows 3.1: A Developer's Guide* [Richter92].

Windows ofrece para su versión 3.1 doce filtros diferentes cada uno de los cuales se activa cuando se cumplen unas condiciones determinadas. Por ejemplo, si en la cola del sistema se encuentra el evento WM_KEYDOWN (provocado por la pulsación de una tecla) y va a ser procesado, se activará el filtro WH_KEYBOARD.

Para que una aplicación pueda ser consciente del proceso anterior es necesario instalar previamente el citado filtro. Esta instalación se realiza mediante una función que toma como parámetros, entre otros, el nombre del filtro (WH_KEYBOARD en el caso anterior) y la función que va a ser llamada cuando se cumplan las condiciones de activación de ese filtro. Esta función, que se denomina *función de filtro*, va a ser definida por el programador y contendrá el proceso que se quiera realizar, que puede incluir la eliminación o modificación del evento. El programador también será responsable de desinstalar los filtros.

Las *funciones de filtro* van a ser llamadas por el propio entorno *Windows* y no desde ninguna otra función de la aplicación. Únicamente el entorno *Windows* es consciente del estado de las colas del sistema y de las aplicaciones y, por tanto, consciente de cuando se cumplen las condiciones de activación de un filtro.

Para un mismo filtro es posible tener activadas varias *funciones de filtro*, cada una de las cuales puede realizar procesos diferentes. Si el citado filtro se activa se irán llamando sucesivamente las *funciones de filtro* empezando por la última función instalada, tal y como se observa en la Figura 5.3.

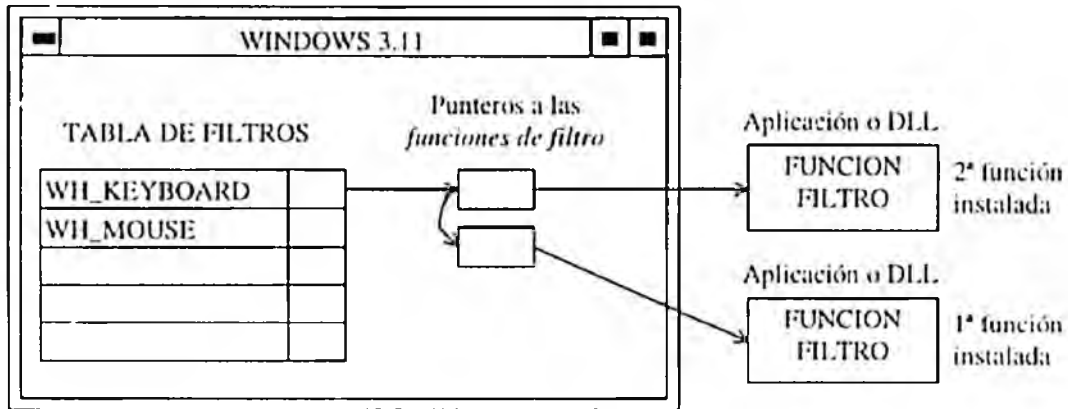


Figura 5.3. Llamada y encadenamiento de las *funciones de filtro*.

En el sistema *DocuLAN* se han utilizado seis filtros cuyo significado se incluye en la Tabla 5.3. Estos filtros han sido instalados para poder capturar los mensajes de toda la sesión *Windows*, ya que el objetivo de *DocuLAN* es gestionar los eventos de las aplicaciones cooperativas y no los que genera la propia aplicación *DocuLAN*.

Identificador	Condiciones de activación del filtro
WH_CBT	Se activa antes de crear, activar, destruir, mover o cambiar el tamaño de una ventana; antes de recuperar un evento del ratón y del teclado de la cola del sistema; o antes de completar un comando del sistema.
WH_JOURNALRECORD	Se activa cada vez que un evento es recuperado de la cola del sistema.
WH_JOURNALPLAYBACK	Se activa cada vez que se realiza una petición de eventos a la cola del sistema.
WH_KEYBOARD	Se activa antes de recuperar un evento del teclado de la cola del sistema.
WH_MOUSE	Se activa antes de recuperar un evento del ratón de la cola del sistema.
WH_CALLWNDPROC	Se activa cuando se llama la función <i>SendMessage</i> .

Tabla 5.3. Descripción de los filtros.

Por cada uno de estos filtros se han definido en la aplicación *DocuLAN* una o varias *funciones de filtro* cuyo comportamiento se describe en la Tabla 5.4.

Ante unas determinadas condiciones se pueden activar varios filtros simultáneamente, por lo que también hay que considerar el orden de llamada de cada uno. Las funciones asociadas a los filtros van a proporcionar el soporte a los módulos del sistema que se describen a continuación.

Identificador	Nombre y descripción de las <i>funciones de filtro</i>
WH_CBT	<i>InitCbtFunc, CSCbtFunc</i> . Se encargan de la gestión de las ventanas cooperativas en el inicio y en el desarrollo de una sesión.
WH_JOURNALRECORD	<i>RecordFunc</i> . Se encarga de detectar los eventos de las aplicaciones cooperativas, los captura y los envía al módulo de comunicaciones.
WH_JOURNALPLAYBACK	<i>RecKeyboardFunc</i> . Se encarga de reproducir los eventos cooperativos que le llegan a través del módulo de comunicaciones.
WH_KEYBOARD	<i>RecKeyboardFunc, PbKeyboardFunc</i> . Se encargan de detectar la entrada de ciertas teclas para las que hay que realizar un tratamiento específico. También posibilitan que se puedan descargar los eventos dirigidos a las aplicaciones cooperativas cuando no se está en posesión del testigo.
WH_MOUSE	<i>RecMouseFunc, PbMouseFunc</i> . Se encarga de desactivar los eventos del ratón dirigidos a las aplicaciones cooperativas cuando ese nodo no posee el testigo.
WH_CALLWNDPROC	<i>RecCallWndFunc</i> . Realiza un tratamiento específico con los mensajes relacionados con las barras de desplazamiento con el objeto de mantener el sincronismo en todos los nodos (situaciones indeterministas).

Tabla 5.4. Descripción de las *funciones filtro*.

5.6 Módulo de Gestión de las Aplicaciones

Se encarga de identificar y controlar las aplicaciones durante la sesión cooperativa. A través de este módulo se puede distinguir si una determinada ventana pertenece a una aplicación cooperativa o pertenece a una aplicación privada del usuario. También es responsable de garantizar en todo momento que las ventanas cooperativas ocupen el mismo espacio y tamaño en todos los nodos.

Esta gestión de las aplicaciones se divide en cuatro procesos que se llevan a cabo de forma consecutiva y que se describen en la Figura 5.4.

1. Proceso de preparación de la sesión.
2. Proceso de arranque de las aplicaciones cooperativas.
3. Proceso de cooperación.
4. Proceso de finalización de la sesión.

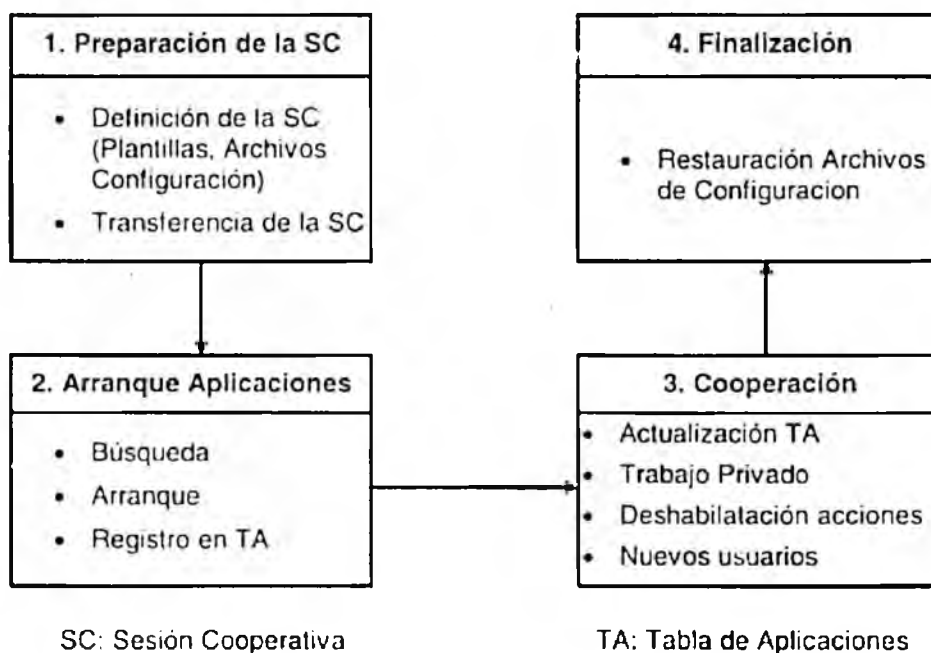


Figura 5.4. Procesos del Módulo de Gestión de Aplicaciones.

5.6.1 Proceso de preparación de la sesión

Esta etapa se inicia cuando el Moderador ha lanzado una petición de establecimiento de una sesión cooperativa y ha sido respondido al menos por un usuario. En ella se realizan las acciones de definición y transferencia de la sesión.

En la definición de la sesión, el Moderador selecciona las aplicaciones y los archivos de datos con los que se quiere cooperar. El sistema *DocuLAN* puede utilizar cualquier aplicación que trabaje bajo el entorno *MS-Windows 3.x*. La única condición necesaria es que la aplicación haya sido definida previamente en una *Plantilla* dentro del ámbito de este sistema. Una *Plantilla* consta de la localización completa de la aplicación y sus archivos de configuración. Los archivos de configuración están asociados a las aplicaciones y contienen información relacionada con diferentes opciones de su apariencia externa. En el entorno *MS-Windows* suelen tener la extensión *ini*. Resulta necesario incluir estos archivos para garantizar que una aplicación es abierta con la misma apariencia externa en todos los nodos participantes.

Para definir una sesión, es suficiente con que la *Plantilla* se encuentre en el nodo Moderador. No obstante, si la citada *Plantilla* está definida en todos los nodos se va a simplificar el proceso posterior de búsqueda de las aplicaciones

En la definición de la sesión, el Moderador selecciona las aplicaciones a través de sus *Plantillas* y selecciona también los archivos de datos que desea abrir con cada aplicación. El número de estos archivos está en función de las propias aplicaciones. Algunas, como *MS-Word*, permiten abrir simultáneamente varios archivos y otras, como *MS-Paintbrush*, únicamente permiten abrir un solo archivo en cada instante.

El sistema también ofrece la posibilidad de almacenar toda la información de definición de una sesión, que consta fundamentalmente de las *Plantillas* y los archivos de datos, en un archivo propio. De esta forma se puede relanzar una sesión anterior mediante la apertura de este archivo.

Una vez que el Moderador ha establecido la conexión con al menos otro usuario y ha seleccionado las aplicaciones y archivos cooperativos comienza el proceso de transferencia de dicha sesión a los participantes. En este proceso se transfieren los siguientes elementos:

1. El nombre de las aplicaciones que van a tomar parte.
2. Los archivos de configuración de las aplicaciones.
3. Los archivos de datos de las aplicaciones.
4. Opciones generales de la sesión.

Los elementos 1 y 2 corresponden a la *Plantilla*. El nombre de las aplicaciones incluye el directorio completo en que se encuentra. Esta información se va a utilizar para localizar dicha aplicación en cada uno de los nodos participantes.

Los archivos de configuración se depositan en el directorio *Windows* o en el directorio donde se encuentra la aplicación. Con esta transferencia se está sustituyendo un archivo existente por otro nuevo. Para no perder los valores del fichero antiguo se cambia de nombre y una vez que la sesión ha finalizado es restaurado con su nombre original. De esta forma, el usuario mantiene sus opciones personalizadas cuando trabaja de forma individual y únicamente debe adaptarse a las opciones elegidas por el Moderador durante el desarrollo de una sesión cooperativa.

Con este mecanismo se consigue que todas las aplicaciones aparezcan en todos los nodos con la misma apariencia. Esta es la primera condición necesaria para el mantenimiento de la consistencia durante una sesión cooperativa.

Asimismo, los archivos de datos son depositados en cada nodo en el *directorio cooperativo*. Este directorio se crea en el momento de instalar el sistema *DocuLAN* y cada usuario determina su nombre y su localización.

Entre las opciones generales de la sesión se encuentran los valores de la *resolución cooperativa*. Este elemento permite que se pueda cooperar de forma

consistente entre nodos que poseen diferentes resoluciones de la pantalla. Para ello es necesario determinar una configuración a la que todos los nodos deben adaptarse. Para realizar este proceso, los participantes de la sesión envían al Moderador la resolución de su pantalla junto con la contestación afirmativa de participación. El Moderador recoge las resoluciones de todos los participantes y toma la menor de ellas como la *resolución cooperativa*, a la cual se van a adaptar todos los nodos. Aquellos que posean unas resoluciones mayores van a ocupar únicamente la parte correspondiente a la *resolución cooperativa*. Por ejemplo, si dos usuarios establecen una sesión y cuentan con resoluciones de 800x600 y 640x480 pixels, la *resolución cooperativa* va a ser 640x480. Una ventana maximizada en el nodo que posee la resolución mayor ocupará únicamente la parte de la pantalla comprendida desde el vértice superior izquierdo hasta la coordenada (640, 480). El proceso de decisión de la *resolución cooperativa* es transparente a los usuarios que van a participar en la sesión.

Otro valor que se transmite es el tiempo máximo que determina que se ha producido una pulsación doble del ratón. Este valor debe ser idéntico en todos los nodos para impedir que esta acción sea considerada en un nodo como dos mensajes independientes de pulsación y en otros como un mensaje de pulsación doble. Este hecho provocaría resultados diferentes en cada nodo y por tanto se originaría la pérdida de la consistencia. Durante la sesión cooperativa, el valor de este tiempo en el Moderador se va a aplicar en todos los nodos. Una vez finalizada la misma se restaura el valor antiguo en cada nodo.

5.6.2 Proceso de arranque de las aplicaciones cooperativas

En este punto, las aplicaciones cooperativas son lanzadas en cada nodo, como corresponde a una arquitectura replicada. Para llevar a cabo esta acción se deben localizar en cada nodo dichas aplicaciones por medio de la información que ha sido transferida desde el Moderador. Esta búsqueda se realiza de forma ordenada en los siguientes lugares:

1. En la variable *path* del sistema.
2. En el archivo de *Plantillas*, donde se almacena la localización de cada aplicación.
3. En el directorio proporcionado por el moderador. Este caso es útil cuando se lanza una aplicación de red que tiene la misma localización para todos los usuarios.

Esta búsqueda no es exhaustiva y puede que no se encuentre la aplicación después de pasar por los tres puntos señalados. Este problema se soluciona dentro del ámbito de *DocuLAN* definiendo las *Plantillas* de las posibles aplicaciones cooperativas en todos los nodos que pueden participar en la sesión.

El sistema *DocuLAN* no incluye un mecanismo de comparación de versiones como el que cuenta el sistema *MMConf* [Crowley90] que garantiza que se lanza la misma versión en todos los nodos.

Después de su localización, estas aplicaciones son abiertas y registradas en una *Tabla de Aplicaciones* de cada nodo, en la que se almacena el identificativo de la ventana principal y el identificativo de la tarea. Todo programa que se ejecuta en el entorno *MS-Windows* crea una tarea en esa sesión y, al menos, una ventana principal. Esta *Tabla de Aplicaciones* es creada y actualizada por medio de los filtros. A través del filtro *WH_CBT* se detecta cuando va a ser creada la aplicación y permite conocer los valores que son almacenados en la *Tabla de Aplicaciones*.

Los filtros también permiten que las aplicaciones cooperativas sean abiertas en la misma posición y con el mismo tamaño en la pantalla de todos los nodos. A través del mismo filtro *WH_CBT* se detecta cuando va a ser activada por primera vez la aplicación y se sustituyen los valores de la posición original de la ventana por los valores que proporciona *DocuLAN*. Estos valores van a distribuir las ventanas cooperativas en forma de mosaico. Durante la cooperación, el usuario activo podrá cambiar esta disposición inicial.

5.6.3 Proceso de cooperación

Una vez que las aplicaciones han sido arrancadas en todos los nodos se puede iniciar la cooperación. Durante la misma es necesario realizar diversas acciones de gestión de las aplicaciones. Una de ellas consiste en la actualización dinámica de la *Tabla de Aplicaciones*. Se añadirán a la tabla las nuevas aplicaciones que se incorporen a la sesión cooperativa y se irán eliminando aquellas que se vayan cerrando. Si la *Tabla de Aplicaciones* no contiene ningún elemento indica que la sesión ha finalizado.

Este proceso permite, asimismo, la realización del trabajo privado. Un usuario pasivo puede trabajar durante el desarrollo de una sesión en sus aplicaciones locales de forma individual e independiente al resto de usuarios. Este trabajo privado se puede realizar mientras no lleguen eventos cooperativos a ese nodo. En el momento que llegue un evento cooperativo debe ser procesado inmediatamente ya que en el entorno *MS-Windows* los eventos del ratón y del teclado son procesados por la ventana activa. Por tanto, para que ese evento se reproduzca correctamente debe ser activada la ventana cooperativa correspondiente. Este proceso se realiza también a través del filtro *WH_CBT*. Cuando se inicia un trabajo privado, se almacena la ventana cooperativa que en ese momento estaba activa, la cual se restaura cuando llega un nuevo evento cooperativo. No sólo es necesario activar la última ventana cooperativa, sino también hay que mantener el resto de ventanas cooperativas en el mismo orden de activación. Este orden, que se denomina *Z-order*, determina qué partes de las ventanas están visibles y cuáles están tapadas por otras ventanas cooperativas. En *DocuLAN* se debe mantener siempre la misma apariencia en las ventanas cooperativas y por tanto cualquier manipulación de la activación de una ventana no debe implicar una alteración del *Z-order*, que se debe mantener idéntico en todos los nodos.

Por esta razón, para asegurar que no se varía el *Z-order* de los participantes, se han deshabilitado algunas funciones relacionadas con la activación de aplicaciones durante la sesión cooperativa. De esta forma, las operaciones de *ALT+TAB* y *ALT+ESC*, que permiten variar la ventana activa, sólo van a tener efecto entre las aplicaciones privadas y por tanto no afectan al orden de activación de las aplicaciones cooperativas.

Por el mismo motivo, se ha deshabilitado la ventana de la lista de tareas. Todas estas acciones son también realizadas a través del filtro WH_CBT. En el primer caso, resulta más complejo porque se sustituye el comportamiento habitual de las operaciones de *ALT+TAB* y *ALT+ESC* por otro nuevo. Es decir, se detecta cuándo se van a realizar estas operaciones, se cancelan y se incluye la nueva funcionalidad implementada, que consiste en buscar a través del *Z-order* únicamente las aplicaciones no cooperativas.

Asimismo, para prevenir la pérdida de la consistencia, la acción de minimizar ha sido desactivada de las ventanas cooperativas. Esta restricción es necesaria ya que el sistema no controla la posición en que aparece el icono. Si se colocaran en posiciones diferentes, la acción de restauración no se realizaría en todos los nodos o podría provocar que se abrieran ventanas diferentes. Ambos resultados provocarían la ruptura de la cooperación.

Una de las principales dificultades de los sistemas de arquitectura replicada es la posibilidad de permitir a nuevos usuarios sumarse a una sesión cooperativa ya iniciada. Esta dificultad radica en el hecho de que habría que reconstruir toda la historia de los eventos generados a partir del estado inicial de la sesión, lo que implica grandes consumos de recursos.

No obstante, en *DocuLAN* se ha implementado una solución que permite incorporarse a un nuevo usuario. Este nuevo participante realiza una petición de entrada a una sesión en curso. El Moderador decide sobre su incorporación y en caso afirmativo el mismo Moderador guarda los cambios realizados en los archivos, cierra la sesión y la reinicia inmediatamente incorporando al nuevo usuario. El resto de usuarios observarán una interrupción de la cooperación y su inmediata reanudación incluyendo al nuevo usuario. Esta solución presenta el inconveniente de la paralización momentánea de la cooperación, pero por otra parte resuelve uno de los principales y más clásicos problemas de la arquitectura replicada.

5.6.4 Proceso de finalización de la sesión

Como ya se ha indicado, los usuarios participantes reciben los archivos de configuración del Moderador para garantizar que las aplicaciones cooperativas se abren con el mismo estado en todos los nodos. Una vez finalizada la sesión se deben restaurar los archivos de configuración originales de cada nodo. Asimismo se deben restaurar algunas opciones recibidas como el valor del tiempo máximo de pulsación doble.

Para que la restauración de los archivos de configuración sea válida, la aplicación correspondiente a la que afecta esa configuración debe estar cerrada. Si no fuera así, en el momento de cerrar la aplicación se actualizarían los ficheros de configuración y por tanto se mantendrían los valores de la sesión cooperativa y no los del propio usuario. Por este motivo, *DocuLAN* comprueba que cuando se realiza una solicitud de finalización de la sesión no se mantienen abiertas las aplicaciones que tienen asociados ficheros de configuración. Si permanecen abiertas estas aplicaciones aparece un mensaje solicitando al usuario que las cierre antes de finalizar la sesión.

5.7 Módulo de Gestión del Testigo

Este módulo se encarga de otorgar a los usuarios el permiso correspondiente para poder trabajar sobre las aplicaciones cooperativas. La posesión del testigo determina el usuario que posee el privilegio de actuar sobre las citadas aplicaciones.

En *DocuLAN* existe un único testigo en la sesión y por tanto sólo un usuario podrá en cada instante escribir sobre las aplicaciones cooperativas. El resto de usuarios podrá solicitarlo u obtenerlo inmediatamente dependiendo de la política de gestión del testigo que se utilice.

El sistema ha sido comprobado con cuatro políticas de intercambio que se describen a continuación:

1. Dirigida por el Moderador. El Moderador decide quien es el usuario que posee el testigo en cada instante.
2. Dirigida por el Poseedor del Testigo. El usuario que posee el testigo decide quien es el siguiente usuario activo.
3. Por peticiones. El usuario que desea obtener el testigo realiza una petición, la cual es almacenada en una cola del tipo FCFS (*First Come First Served*). Cuando el poseedor actual libera el testigo, éste pasa al primer usuario de la cola.
4. Por prioridades. Es similar a la política anterior con la particularidad de que las peticiones son ordenadas en la cola dependiendo de la prioridad asignada previamente a cada usuario. El testigo pasará al primer usuario de la cola.

Asimismo, en todas las políticas anteriores se permite que el Moderador tome inmediatamente el testigo en cualquier momento, lo que garantiza que pueda existir un cierto control sobre el desarrollo de la sesión e impedir que un usuario paralice la cooperación al no querer ceder el testigo.

No se puede afirmar cuál es la política óptima. En cada caso, dependiendo de las características de los participantes y de los objetivos de la reunión habrá que seleccionar el método apropiado [Lauwers90b]. En este trabajo no se han realizado estudios exhaustivos en este sentido.

Las políticas de gestión del testigo se implementan por medio de mecanismos. En *DocuLAN* estos mecanismos de gestión del testigo están basados en las tres primitivas siguientes:

1. Petición del testigo. El usuario realiza una petición para tomar el testigo. Esta petición se almacena en una cola.
2. Captura del testigo. El usuario toma el testigo y pasa a ser el usuario activo.
3. Cesión del testigo. El usuario activo se desprende del testigo y pasa a ser un usuario pasivo.

La combinación de estas primitivas permite implementar fácilmente cualquier política de intercambio del testigo.

5.8 Módulo de Comunicaciones

Este módulo es responsable de la transmisión de la información entre los participantes en una sesión cooperativa. Esta información comprende los archivos de datos, los archivos de configuración, los eventos cooperativos e información de gestión de la sesión.

Hay que destacar que el diseño modular del sistema permite utilizar diferentes protocolos de comunicaciones sin tener que modificar el resto del sistema. Este módulo se implementa a través de una Librería de Enlazado Dinámico (DLL) que se enlaza con el programa principal en tiempo de ejecución y no en el tiempo de compilación. De esta forma la librería puede ser modificada sin afectar al resto del programa. El sistema *DocuLAN* ha sido evaluado con tres módulos diferentes de comunicaciones, que se detallan a continuación.

1. Módulo punto a punto sobre TCP/IP. Permite la cooperación entre dos usuarios. Por su sencillez ha sido el módulo más utilizado para estudiar los temas relacionados con la consistencia, el sincronismo, el tráfico de red y el tiempo de respuesta. Se ha utilizado el servicio denominado *stream* sobre el protocolo TCP, el cual implementa un *circuito virtual* entre el emisor y el receptor que garantiza que los paquetes de datos llegan de forma fiable y ordenada a su destinatario [Stevens94].
2. Módulo multipunto con servidor TCP/IP. Este módulo se utilizó en el proyecto ESPRIT MMTCA [Crowe94]. Incluía un servidor *UNIX* que se encargaba de gestionar la sesión y recoger los eventos cooperativos del usuario activo y distribuirlos al resto de participantes. Se utilizó el servicio *datagrama* sobre el

protocolo UDP. Este servicio no garantiza que los datos lleguen correctamente a su destino [Stevens94] por lo que se implementó un mecanismo de seguridad que en caso de error en la transmisión reenviaba de nuevo la información desde el servidor al usuario correspondiente. Este módulo permitía establecer una conferencia entre más de dos usuarios.

3. Módulo punto a punto sobre RDSI. Permite la cooperación entre dos usuarios conectados a través de la Red Digital de Servicios Integrados (RDSI). Esta versión se ha integrado en una aplicación de videoconferencia de sobremesa sobre RDSI. Esta integración se ha realizado en el proyecto VIDOC [Herrero95]. Este sistema se basa en la utilización de las tarjetas de codificación de la empresa *Bitfield* [Bitfield98] y el driver CAPI 1.1 [ISDN90] para las comunicaciones RDSI. En la actualidad están apareciendo bastantes productos de este tipo que integran la comunicación de vídeo, audio y documentos. Esta última opción, la conferencia de documentos, se implementa siguiendo el modelo centralizado [Labriola95] [Taylor95].

5.9 Módulo de Gestión de Eventos

Este módulo constituye la base de la cooperación ya que se encarga de la gestión de los eventos cooperativos, es decir, de su captación en el nodo donde se originan y de su reproducción en el resto de nodos participantes. Estas tareas se llevan a cabo fundamentalmente a través de los filtros WH_JOURNALRECORD y WH_JOURNALPLAYBACK.

El filtro WH_JOURNALRECORD se encarga de capturar los eventos cooperativos en el nodo activo. La función asociada a este filtro se llama siempre que se genere un evento del ratón o del teclado. Esta función almacenará los eventos únicamente cuando la aplicación activa sea cooperativa. Estos eventos son enviados al resto de participantes a través del módulo de comunicaciones. En los nodos pasivos los

eventos cooperativos son almacenados temporalmente en una cola, al mismo tiempo que se activa el filtro de reproducción WH_JOURNALPLAYBACK. Este filtro insertará los eventos en la cola del sistema y los tratará a partir de ese momento de igual forma que los eventos generados por el propio nodo. El resultado será su reproducción sobre la aplicación cooperativa activa.

Este módulo por su complejidad necesita asimismo unos mecanismos específicos para tratar determinadas situaciones.

5.9.1 Mecanismo de eliminación de eventos no cooperativos

Los nodos pasivos, al no poseer el testigo, no deben reproducir los eventos que se generen localmente y que tengan como destinatario una aplicación cooperativa. Un usuario pasivo puede tener como ventana activa una ventana cooperativa que sería la destinataria de los eventos que generase con el teclado y el ratón. Pero al estar participando en una conferencia, esos eventos deben ser anulados ya que el citado usuario no posee en ese momento el privilegio de escribir sobre las aplicaciones cooperativas.

En una política libre de gestión del testigo no sería necesario controlar esta situación ya que todos los participantes tendrían la posibilidad de generar entradas sobre las aplicaciones cooperativas.

En *DocuLAN* se implementa este mecanismo a través de los filtros WH_JOURNALPLAYBACK, WH_KEYBOARD y WH_MOUSE. Con el primero, se pueden identificar los eventos cooperativos que llegan al nodo, los cuales son marcados con un *flag* cuando se insertan en la cola del sistema. A través de los filtros WH_KEYBOARD y WH_MOUSE se pueden acceder a todos los eventos de una aplicación, tanto los cooperativos marcados con el *flag*, como los no cooperativos generados por el usuario. En sus *funciones de filtro* se eliminarán los eventos generados localmente que no están marcados con el *flag*.

5.9.2 Mecanismo de pulsar y arrastrar

Una de las principales dificultades para el mantenimiento del sincronismo ha sido ocasionada por la secuencia de acciones de pulsar, arrastrar y liberar un botón del ratón (*click & drag*). En la mayoría de los casos, estas acciones no ocasionan ningún problema, como por ejemplo, al mover una ventana mediante la acción de pulsar en la barra de título y trasladar la ventana hasta el lugar deseado. Pero hay determinadas situaciones en la que esta secuencia de acciones es problemática. Un ejemplo ocurre en las aplicaciones gráficas (ej. *MS-Paintbrush*) cuando se utiliza el botón para dibujar una línea, un rectángulo u otra figura. Esta acción se lleva a cabo pulsando inicialmente el botón, arrastrándolo hasta completar la figura y liberándolo en ese momento. Durante el tiempo que dura esta operación la ventana sobre la que se dibuja posee el control total del entorno *Windows* y ninguna otra ventana puede procesar mensajes que tenga en espera hasta que se libere el botón del ratón.

Este comportamiento se debe a que el entorno *Windows 3.x* no es un entorno de multitarea real. Únicamente proporciona la denominada multitarea preferente por la que, en determinados momentos, una aplicación puede ceder el control a otra aplicación para que procese sus mensajes en espera [Petzold90]. Pero esta posibilidad es limitada y no se produce cuando se realizan determinadas operaciones como la de dibujar una figura en una aplicación gráfica. Esta limitación queda solventada en el entorno *Windows 95*, el cual soporta una multitarea real [King94].

Este comportamiento en las aplicaciones gráficas provocaba la pérdida del sincronismo en los nodos pasivos. En estos se recibía y se reproducía el primer evento de la secuencia, la pulsación del ratón. La aplicación entraba en ese momento en el periodo en el que poseía el control total del entorno. Pero, al mismo tiempo, esa condición impedía a la propia aplicación *DocuLAN* recibir y procesar el resto de eventos de la secuencia. El sistema se quedaba bloqueado, a la espera de más mensajes, pero su propio estado impedía que se recibieran esos mensajes y se finalizara la acción. El resultado final era que la acción se había realizado sin problemas en el nodo activo, pero

en los nodos pasivos, únicamente se había reproducido un evento de pulsación del ratón y a continuación el sistema quedaba bloqueado.

Para poder solucionar este problema, los eventos comprendidos entre la pulsación y la liberación del botón del ratón únicamente podían empezar su procesamiento cuando todos ellos se encontraran ya en el propio nodo pasivo. Para implementar esta solución, se creó una segunda cola en la que se almacenaba la secuencia de eventos entre la pulsación y la liberación. Cuando llegaba este último evento, toda la secuencia completa de eventos era traspasada a la cola general implementada en *DocuLAN*. Desde ese punto los eventos podían ser ya procesados a través del filtro de reproducción *WH_JOURNALPLAYBACK*. Cuando se reproducía el primer evento de la secuencia, el siguiente podía ser recuperado inmediatamente y así hasta llegar al evento de liberación del botón.

La aplicación *DocuLAN* trata todos los eventos de pulsación y liberación del ratón a través de este mecanismo aunque, como ya se ha dicho, no sería necesario realizarlo en todas las situaciones. Pero en el momento de recibir el primer evento de la secuencia no es posible determinar si esa acción va a resultar problemática o no, por lo que hay que iniciar este mecanismo en todos los casos.

Como resultado de la introducción de este mecanismo, la acción de pulsación, arrastre (si lo hay) y liberación del ratón se va a iniciar en los nodos pasivos con un retraso que dependerá de la propia duración de la acción. Este retraso no resulta problemático, ya que los usuarios pasivos no son conscientes de que el usuario activo ha iniciado una acción de ese tipo. Por otra parte, en el momento en que se inicie la reproducción de la secuencia de eventos en los nodos pasivos, ésta se va a realizar de forma rápida ya que se van a reproducir todos los eventos consecutivamente sin ningún tiempo de espera entre un evento y el siguiente, aunque hubiera habido un retraso en su generación.

5.9.3 Mecanismo de pulsación doble

La introducción del mecanismo anterior resuelve un problema, pero crea otro nuevo en la acción de doble pulsación del botón del ratón (*doble click*). Esta pulsación doble provoca una acción determinada si el intervalo entre ambas pulsaciones es menor que un valor configurable en cada entorno *MS-Windows*.

De la forma en que inicialmente se ideó el mecanismo anterior, cuando en el nodo pasivo se recibía el evento de liberación del ratón se transmitía toda la secuencia a la cola general de eventos. No se tenía en cuenta la posibilidad de que tras esa primera pulsación se produjera inmediatamente otra similar que provocaría en el nodo activo una pulsación doble, mientras que en los nodos pasivos se traduciría como dos pulsaciones individuales consecutivas. En ambos nodos, se generarían diferentes mensajes por lo que al perderse el sincronismo, se perdería la consistencia y la cooperación.

La solución ha consistido en introducir un control adicional en la cola en la que se almacena la secuencia de eventos de las pulsaciones del ratón. Cuando se recibe el evento de liberación se espera un tiempo para comprobar que no llega otro evento de pulsación dentro del periodo máximo establecido. Si se sobrepasa ese tiempo se transmite la secuencia de eventos a la cola general. Si llega un evento de pulsación dentro del margen, se espera al próximo evento de liberación y se transmiten conjuntamente los eventos de la acción de pulsación doble para su correcta reproducción.

Para que este mecanismo sea válido y en general para evitar la pérdida del sincronismo, durante una sesión cooperativa todos los nodos deben poseer el mismo valor del tiempo máximo de pulsación doble. Este ajuste es realizado por el sistema al inicio de la sesión pasando el valor del Moderador al resto de usuarios. Una vez finalizada la sesión, el valor antiguo es restaurado en cada nodo.

5.9.4 Mecanismo de control de situaciones indeterministas

Una situación indeterminista ocurre cuando partiendo del mismo estado inicial y la misma secuencia de entradas se producen resultados diferentes [Lauwers90]. Esta situación aparece en la sesión cooperativa cuando los eventos son transferidos correctamente, pero se obtienen resultados diferentes en cada nodo.

El ejemplo más característico y que se ha abordado en este sistema es la utilización de las barras de desplazamiento (*scrolling*). Por ejemplo, si se toma la barra de desplazamiento vertical de un procesador de textos, se pueden realizar tres tipos de operaciones:

1. Mover el documento una o varias líneas hacia arriba o hacia abajo.
2. Mover el documento una o varias páginas hacia arriba o hacia abajo.
3. Moverse a una posición determinada del documento.

En el primer caso aparecen las situaciones indeterministas. La acción de mover el documento varias líneas se realiza mediante la presión continua de las flechas superior o inferior de la barra. En esta operación hay que distinguir tres conceptos:

1. Los eventos de pulsación y liberación del botón del ratón.
2. Los mensajes de desplazamiento que se generan (WM_VSCROLL), cuyo número dependerá del tiempo de pulsación del botón.
3. El número de líneas que se mueve el documento que dependerá del número de mensajes generados y de la propia aplicación.

Al presionar de forma continua la flecha de desplazamiento el documento se moverá un número de líneas que dependerá del número de mensajes de desplazamiento generados. Este número de mensajes dependerá, a su vez, del tiempo que se mantenga presionada la flecha de la barra de desplazamiento.

Al introducir el factor tiempo se produce una situación indeterminista en la sesión cooperativa, ya que el número de mensajes que se pueden generar dependerá de la capacidad de procesamiento del ordenador de cada nodo.

Pero, asimismo, el número de mensajes de desplazamiento generados va a depender del momento en que se transmitan los eventos de pulsación y liberación del ratón. En esta circunstancia interviene el mecanismo introducido anteriormente que provoca que la cadena de eventos comprendidos entre la pulsación de la flecha de desplazamiento y su liberación sean reproducidos de forma consecutiva en los nodos pasivos (sin tener en cuenta el tiempo que transcurre entre ellos en su generación). Este hecho origina en los nodos pasivos un único mensaje de desplazamiento y, por tanto, el movimiento de una sola línea.

De esta forma, la presión continua de la flecha de la barra de desplazamiento provoca en el nodo activo el movimiento del documento un número determinado de líneas, mientras que en los nodos pasivos, únicamente, el documento será desplazado una sola línea. Este comportamiento provoca la pérdida del sincronismo.

La solución adoptada ha sido un tanto restrictiva y basada en el comportamiento de los nodos pasivos. Ese comportamiento se ha simulado en el nodo activo. Por tanto, en el sistema *DocuLAN*, la opción de pulsación continua de la flecha de desplazamiento se ha transformado en la pulsación y liberación inmediata de la flecha, independientemente del tiempo que se mantenga pulsado el botón, y va a tener como resultado el movimiento de una sola línea.

Este mecanismo se ha implementado a través del filtro `WH_CALLWNDPROC`. Este filtro se activa cuando se procesan la mayoría de los mensajes del entorno *Windows*, por lo que se recomienda utilizarlo de forma muy limitada porque debido a su frecuente activación se ralentiza significativamente todo el entorno [Marsh92]. En *DocuLAN* se van a tratar con este filtro los mensajes de desplazamiento vertical y horizontal (`WM_VSCROLL` y `WM_HSCROLL`), pero el filtro únicamente va a estar activo durante los periodos comprendidos entre la pulsación

y la liberación del ratón por lo que no va a afectar al rendimiento general del entorno. La *función de filtro* asociada se va a encargar de transformar en el nodo activo la pulsación continua de la flecha de desplazamiento en el movimiento de una sola línea. Esta función permitirá el procesamiento de un único mensaje de desplazamiento (WM_VSCROLL o WM_HSCROLL) y eliminará el resto de mensajes.

5.9.5 Mecanismo de gestión de los recursos externos

Durante el desarrollo de una sesión un usuario puede tener la necesidad de introducir un nuevo archivo para cooperar. Esta operación se puede realizar en *DocuLAN* pulsando un botón de la ventana de control. A través de una caja de diálogo se seleccionará el archivo y posteriormente se transmitirá al resto de usuarios de la sesión. Si en lugar de realizar esta operación, el usuario activo se moviera por el árbol de directorios de la caja de diálogo de *Abrir archivo* de las aplicaciones, cada usuario estaría visualizando su propio estructura de archivos. En consecuencia, si se seleccionase un archivo, sería diferente en cada nodo con lo que se perdería la consistencia.

Este mecanismo aborda una operación concreta de la gestión de los recursos externos, una cuestión importante en la que *DocuLAN* presenta limitaciones, como se explica posteriormente.

5.9.6 Mecanismo de mejora del rendimiento

Para aprovechar al máximo las ventajas de este modelo se ha incluido también un mecanismo por el cual no se transmiten aquellos eventos que no proporcionan una información significativa. Por ejemplo, no se envían los eventos de movimiento de ratón cuando el puntero está situado sobre una aplicación no cooperativa. Además esta hecho podría resultar molesto para los usuarios pasivos al ver como el puntero se mueve por una zona de la pantalla privada que no forma parte de la sesión cooperativa.

El filtro WH_JOURNALRECORD se encarga de detectar este tipo de eventos y de no enviarlos al módulo de comunicaciones para que no lleguen al resto de usuarios.

5.10 Limitaciones de *DocuLAN*

El sistema *DocuLAN*, como cualquier sistema que sigue un modelo replicado, presenta una serie de limitaciones debidas a las dificultades en el mantenimiento de la consistencia. Algunas pueden provocar la pérdida de la cooperación, otras obligan a los usuarios a realizar determinadas operaciones de forma diferente a como las realizan habitualmente. A continuación se detallan estas limitaciones.

5.10.1 Recursos externos

DocuLAN no incluye un mecanismo general para trabajar con los recursos privados de cada usuario. Como ya se ha indicado la cooperación se rompería, por ejemplo, si se tratara de incorporar un archivo a la sesión a través de la caja de diálogo de *Abrir archivo* de las aplicaciones porque provocaría que cada usuario abriría un archivo diferente. *DocuLAN* permite incorporar nuevos archivos a una sesión ya iniciada a través de un botón de la ventana de control de la sesión.

Una de las cuestiones no resuelta en *DocuLAN* es el traspaso de información a una aplicación a través del portapapeles. El portapapeles es un espacio de almacenamiento en el que cualquier aplicación puede depositar información o tomar la que allí se encuentre. En cada instante únicamente se almacena un bloque de información que es el último que ha sido depositado. Si una aplicación cooperativa toma información del portapapeles es probable que sea diferente en cada nodo, ya que puede que se haya introducido información de una aplicación local que no participa en la sesión. La solución sería similar a la apertura de archivos. Se debería implementar un portapapeles cooperativo, accesible a través de la ventana de control de la sesión, que

simulara un espacio común a nivel de la sesión en el que cualquier usuario pudiera depositar o recuperar información.

Por otra parte, el acceso a los dispositivos de salida, como la impresión de un documento o el envío de un mensaje de correo, normalmente no va a provocar la pérdida de la cooperación, sino la repetición de la operación en cada nodo.

En general, el usuario debe ser consciente del comportamiento de *DocuLAN* ante los dispositivos de entrada y de salida para no realizar acciones que originen la pérdida de la cooperación.

5.10.2 Problemas concretos

Asimismo *DocuLAN* presenta otros problemas con operaciones específicas que se detallan a continuación.

1. El mecanismo introducido de pulsar y arrastrar presenta problemas en el tratamiento de los menús expandidos. A estos menús se accede a través de una flecha situada en la parte derecha de una opción de menú anterior. El problema aparece porque los eventos comprendidos entre la pulsación y la liberación del ratón son reproducidos consecutivamente sin tener en cuenta su tiempo relativo de generación. Si el acceso a una opción del menú expandido se realiza en una única secuencia de pulsación y liberación del ratón, este menú expandido puede que no se despliegue en los usuarios pasivos, ya que el evento de selección de la opción final del menú se procesará antes de que se haya desplegado el propio menú. En cambio, no habrá problemas si primero se pulsa la opción de menú anterior (la que tiene la flecha a su derecha) y se libera el botón del ratón. Esta acción provocará que se despliegue el menú correctamente en todos los nodos.

2. En las situaciones indeterministas no se ha abordado el desplazamiento automático de un documento. Esta situación se produce cuando se selecciona parte de un texto en un documento y con el ratón pulsado se continúa la selección hacia los bordes de la ventana. La parte visible del documento se irá desplazando dependiendo del tiempo de selección. En una sesión cooperativa, el documento se desplazará a una posición diferente en cada nodo. No se ha introducido en *DocuLAN* ningún mecanismo para solucionar este problema por la imposibilidad de crear un mecanismo general ya que se generan mensajes diferentes en cada aplicación. Así *MS-Notepad* genera mensajes WM_VSCROLL, y *MS-Word* únicamente genera mensajes WM_MOUSEMOVE. La única posibilidad se basaría en lo introducción de mecanismos específicos para cada aplicación.

3. Otra situación indeterminista no tratada se produce con los controles denominados *spin-boxes*. Este control consiste en un campo de edición que cuenta en su parte derecha con una flecha hacia arriba y otra hacia abajo. Con su pulsación se modifica el valor del campo de edición. Es habitual su uso para actualizar los campos de fecha y hora del sistema. La situación indeterminista se produce si se mantiene pulsada durante un periodo de tiempo una de estas flechas. El resultado del campo de edición será diferente en cada usuario ya que dependerá de la capacidad de procesamiento del ordenador. Para solucionar este problema habría que introducir un mecanismo similar al de la pulsación continua de la flecha de desplazamiento.

4. Otra limitación ya mencionada afecta a la activación de las ventanas cooperativas. Para evitar la pérdida de la consistencia, las operaciones *ALT+TAB* y *ALT+ESC*, que permiten variar la aplicación activa, sólo afectan a las ventanas no cooperativas. Por el mismo motivo, se ha deshabilitado la ventana que contiene la lista de tareas y la operación de minimizar las ventanas cooperativas.

6. Evaluación del Sistema *DocuLAN*

6.1 Comparación con sistemas centralizados

Para evaluar el sistema *DocuLAN* se va a realizar un estudio comparativo con dos aplicaciones comerciales que siguen el modelo tradicional de una arquitectura centralizada. Estas aplicaciones son *FarSite Document Conferencing* de DataBeam [Davis95] [World96] y *ProShare Personal Conferencing* de Intel [Gold94] [Taylor95].

Los productos seleccionados son un ejemplo muy representativo del modelo centralizado y pertenecen a dos empresas que lideran el desarrollo de sistemas de trabajo cooperativo.

6.1.1 *Farsite Document Conferencing*

Esta aplicación ha sido desarrollada por la compañía *DataBeam*, empresa que ha mantenido desde su creación en 1983 una destacada posición dentro del mercado de sistemas de conferencia de datos multipunto. Ha participado activamente en la definición del estándar T.120 y ha sido pionera en el lanzamiento de productos que siguen este estándar [DataBeam98].

La primera versión de la aplicación *FarSite* apareció en 1993. Permite trabajar simultáneamente a personas situadas en diferentes lugares físicos a través de una pizarra electrónica, la compartición de aplicaciones o la transferencia de ficheros. En este estudio se ha utilizado la versión *FarSite Corporate Edition 3.0a* [Farsite96]. Sus principales características son las siguientes:

- Soporta la conexión mediante modems y sobre redes con protocolo TCP/IP y Novell IPX.
- Permite establecer una conferencia entre más de dos usuarios.
- Cumple el estándar T.120.
- Tiene la capacidad de incluir nuevos usuarios en una conferencia ya iniciada.
- Permite compartir cualquier aplicación bajo el entorno *MS-Windows*, lo que posibilita el trabajo simultáneo sobre un procesador de textos o una hoja de cálculo. Estas aplicaciones se lanzan en un único ordenador.
- Permite compartir Libros de Trabajo, que consisten en una o varias ventanas, denominadas Páginas que pueden importar documentos, dibujos y gráficos en diferentes formatos.
- Puede capturar cualquier aplicación abierta y transferir su ventana o ventanas a un Libro de Trabajo.
- Permite realizar anotaciones en forma de líneas, rectángulos, elipses o texto sobre las Páginas compartidas.
- Transfiere documentos a los usuarios remotos sin interrumpir para ello la conferencia.
- Ha sido incluida en algunos productos de videoconferencia de sobremesa sobre RDSI, como el *MediaMaster* de la empresa *Bitfield* [Mmaster96].

6.1.2 Intel ProShare Personal Conferencing Software

Esta aplicación permite interactuar a dos usuarios mediante la visualización simultánea de la misma información en ambas pantallas. Pertenece a la familia de productos de trabajo cooperativo *Intel ProShare* [Intel98]. Se distribuye de forma independiente y junto con los productos de videoconferencia de la misma empresa. En estas pruebas se ha utilizado la versión *ProShare Personal Conferencing Premier Edition Software 1.9a* [Proshare94]. Sus principales características son las siguientes:

- Soporta la conexión por medio de modems y con redes con protocolo NetWare, NetBIOS y TCP/IP.
- Permite compartir cualquier aplicación sobre el entorno *MS-Windows*. La aplicación se lanza en un ordenador y puede ser utilizada simultáneamente por dos usuarios.
- Soporta la realización de anotaciones (líneas, rectángulos, elipses, texto) sobre una aplicación de pizarra electrónica de múltiples ventanas.
- Permite transferir ficheros al usuario remoto sin interrumpir la conferencia.

Se observa que la aplicación *FarSite* ofrece mayores prestaciones que *ProShare*, como la posibilidad de establecer conferencias entre más de dos usuarios. Pero en general la funcionalidad de ambas aplicaciones es muy similar y coincide con lo que ofrecen otras aplicaciones del mismo tipo.

6.1.3 Descripción del entorno de pruebas

En este estudio se va a hacer uso de la funcionalidad de compartición de aplicaciones de *FarSite* y *ProShare*. Su comportamiento va a ser comparado con la aplicación *DocuLAN*. En las pruebas se van a establecer conferencias entre dos usuarios a través del protocolo TCP/IP.

Las pruebas se han realizado con aplicaciones habituales del entorno *MS-Windows*, como *MS-Word 6.0*, *CorelDraw 5.0*, *MS-Paintbrush* y *MS-Notepad*. A priori, se prevé que las diferencias más significativas se produzcan en las aplicaciones gráficas. Por este motivo se han escogido dos aplicaciones de este tipo, *CorelDraw 5.0* y *MS-Paintbrush*, de diferentes características. La primera de ellas consume más recursos que la segunda. Por otra parte, la aplicación *MS-Word* es uno de los procesadores de texto más utilizados. También se ha escogido *MS-Notepad* por ser un editor básico muy utilizado que consume muy pocos recursos.

Los ordenadores sobre los que se van a realizar las pruebas son dos equipos similares con las siguientes características:

- PC compatible con procesador Pentium 100 Mhz
- 16 MB RAM
- 1 GB de disco duro
- 1 MB de VGA
- Adaptador Ethernet *3Com Etherlink III Bus-Master PCI [3Com95]*
- Sistema Operativo *Microsoft Windows para Trabajo en Grupo 3.11*
- Kernel TCP/IP compatible Winsock 1.1 *Trumpet Winsoek Version 2.1 Rev F [Tattam94]*.

Un elemento que influye en el rendimiento final de los sistemas centralizados *FarSite* y *ProShare* es el ordenador en el que se lanzan las aplicaciones. Se recomienda siempre que esta operación se realice en el ordenador más potente. En estas pruebas, al utilizar dos equipos similares, no es necesario considerar esta circunstancia.

Se van a realizar ocho pruebas con las aplicaciones seleccionadas (*MS-Word 6.0*, *CorelDraw 5.0*, *MS-Paintbrush* y *MS-Notepad*). Cada una de ellas consta de una serie de acciones básicas como: escribir un documento, utilizar la barra de desplazamiento, dibujar un gráfico o mover objetos seleccionados. Todas las pruebas son ejecutadas tanto por el Moderador (el usuario que inicia la sesión) como por el

Participante (quien acepta la invitación) ya que existen diferentes comportamientos en ambos casos.

Las mediciones que se van a llevar a cabo son dos:

- Tráfico de red: la cantidad de bytes que se transmiten de un equipo a otro durante la ejecución de cada una de las pruebas.
- Tiempo de respuesta: el tiempo que transcurre desde la finalización de una acción realizada mediante el teclado o el ratón hasta la visualización de su resultado en la pantalla.

Para medir el tráfico de red que se genera en cada uno de los sistemas se ha utilizado la aplicación *NetXRay 2.0.5* de la compañía *Cinco Networks Inc.* [Cinco98]. Este producto permite, entre otras funciones, medir el número de bytes que se transmite entre dos nodos conectados en red bajo un determinado protocolo. En las tablas donde se reflejan los resultados se indica el número de bytes y de paquetes que transmiten el Moderador y el Participante en cada una de las pruebas.

El tiempo de respuesta se ha medido en segundos. No se considera necesario realizar unas mediciones más exactas ya que sólo se van a comentar aquellos resultados en los que se produzcan grandes diferencias. En este sentido, los valores de 0 y 1 segundo van a corresponder con una respuesta casi inmediata para el usuario.

Las pruebas se han realizado sobre una red Ethernet de 10 Mb en la que están conectados habitualmente 125 ordenadores PCs bajo una red Novell y 25 estaciones de trabajo *UNIX*. No obstante, todas las pruebas se han llevado a cabo con una muy baja actividad de la red lo que ha evitado cualquier incidencia negativa del tráfico general sobre el flujo concreto de datos que se deseaba medir.

Todas las operaciones han sido medidas como mínimo en tres ocasiones. Estos tres resultados ofrecían en todos los casos valores similares y coherentes. Las mínimas diferencias entre ellos se deben a que las pruebas dependen del tiempo de ejecución, es decir, del tiempo que se tarda en escribir un texto o en dibujar un gráfico, el cual puede

variar en cada una de las mediciones. En este estudio se ha reflejado en cada prueba el valor intermedio, entre los tres resultados coherentes obtenidos.

A continuación, en las Tablas 6.1 - 6.8 se recogen los resultados de las pruebas. En cada una de ellas, se diferencian los datos obtenidos cuando la acción es llevada a cabo por el Moderador y cuando es realizada por el Participante. En la primera columna se indica las aplicaciones utilizadas. Las siguientes dos columnas reflejan la cantidad de información (en bytes) y el número de paquetes generados por el Moderador. En la siguiente columna se incluye el tiempo de respuesta del Moderador. Las tres últimas columnas recogen los mismos datos descritos anteriormente generados por el Participante.

6.2 Resultados de las pruebas

Prueba 1

Aplicación: MS-Word 6.0

Descripción: Escribir en un documento nuevo el siguiente texto:

GROUPWARE: Sistemas que dan soporte a grupos de personas involucradas en una tarea común (o en un objetivo común) y que proporcionan una interfaz a un entorno de trabajo compartido.

Acción realizada por el Moderador:

	Moderador			Participante		
	Bytes	Paquetes	T.r.	Bytes	Paquetes	T.r.
FarSite	337.058	1.038	0	33.540	559	0
ProShare	232.033	976	0	62.437	976	0
DocuLAN	25.908	381	0	22.860	381	0

Acción realizada por el Participante:

	Moderador			Participante		
	Bytes	Paquetes	T.r.	Bytes	Paquetes	T.r.
FarSite	226.972	823	0	34.797	481	1
ProShare	248.639	1.354	0	93.041	1.350	1
DocuLAN	22.320	372	0	25.296	372	0

T.r. (Tiempo de respuesta en segundos): Tiempo que transcurre desde que se pulsa la última tecla del texto hasta que se visualiza en la pantalla.

Tabla 6.1. Resultados de la Prueba 1.

Prueba 2

Aplicación: *MS-Word 6.0*

Descripción: Realizar un movimiento continuo de la barra de desplazamiento del documento *ejemplo5.doc* incluido en el subdirectorio *wordcbt* de la aplicación *MS-Word 6.0*. Este documento ocupa 2 páginas y 29.696 bytes. El cursor se encuentra en el inicio del documento. La prueba finaliza cuando el cursor alcanza el final después de pulsar de forma continua la tecla *Av Pág.* (Algunas configuraciones significativas son: *Ver - Diseño de Página* y *Zoom - Ancho de Página.*)

Acción realizada por el Moderador:

	Moderador			Participante		
	Bytes	Paquetes	T.r.	Bytes	Paquetes	T.r.
<i>FarSite</i>	144.212	406	4	12.720	212	6
<i>ProShare</i>	52.695	181	3	11.565	181	3
<i>DocuLAN</i>	1.402	20	2	1.200	20	2

Acción realizada por el Participante:

	Moderador			Participante		
	Bytes	Paquetes	T.r.	Bytes	Paquetes	T.r.
<i>FarSite</i>	168.178	472	8	21.466	305	10
<i>ProShare</i>	70.107	242	3	16.705	242	4
<i>DocuLAN</i>	1.140	19	2	1.306	19	2

T.r. (Tiempo de respuesta en segundos): Tiempo que transcurre desde que se inicia la operación mediante la tecla de *Av Pág* hasta que se llega al final del documento.

Tabla 6.2. Resultados de la Prueba 2.

Prueba 3

Aplicación: Corel 5.0

Descripción: Dibujar un gráfico en un documento vacío. Las operaciones que se realizan son las siguientes: 1) Dibujar dos elipses. 2) Rellenar ambas con diferentes colores. 3) Alinear ambos dibujos con *Organizar-Alinear*. 4) Dibujar un rectángulo que contenga ambas elipses. 5) Ajustar el zoom al tamaño del gráfico. 6) Escribir en la parte inferior del gráfico el siguiente texto: *Prueba de tráfico de red*. 7) Cambiar el formato del texto con *Texto-Character*.

Acción realizada por el Moderador:

	Moderador			Participante		
	Bytes	Paquetes	T.r.	Bytes	Paquetes	T.r.
<i>FarSite</i>	368.727	1.881	3	78.120	1.302	6
<i>ProShare</i>	255.104	1.139	1	72.528	1.134	4
<i>DocuLAN</i>	62.474	914	1	54.480	908	1

Acción realizada por el Participante:

	Moderador			Participante		
	Bytes	Paquetes	T.r.	Bytes	Paquetes	T.r.
<i>FarSite</i>	261.568	1.529	4	75.675	983	6
<i>ProShare</i>	264.220	1.626	1	118.353	1.622	5
<i>DocuLAN</i>	52.740	879	1	60.254	883	1

T.r. (Tiempo de respuesta en segundos): Tiempo que transcurre desde que se pulsa el botón *Aceptar* del último diálogo *Texto-Character* hasta que se visualiza completamente la figura en la pantalla.

Tabla 6.3. Resultados de la Prueba 3.

Prueba 4

Aplicación: *Corel 5.0*

Descripción: Mover todos los objetos del archivo *eye.cdr* incluido en el subdirectorio *draw/muestras* de la aplicación *Corel 5.0*. Este gráfico ocupa 69.242 bytes. En el estado inicial, están seleccionados todos los objetos del gráfico y la prueba consiste en mover toda la figura hacia un extremo de la ventana mediante la pulsación continua del ratón. El zoom está ajustado al tamaño del dibujo.

Acción realizada por el Moderador:

	Moderador			Participante		
	Bytes	Paquetes	T.r.	Bytes	Paquetes	T.r.
<i>FarSite</i>	62.428	258	2	10.860	181	7
<i>ProShare</i>	118.415	290	2	18.090	283	3
<i>DocuLAN</i>	2.584	38	1	2.280	38	1

Acción realizada por el Participante:

	Moderador			Participante		
	Bytes	Paquetes	T.r.	Bytes	Paquetes	T.r.
<i>FarSite</i>	57.504	249	3	8.159	118	8
<i>ProShare</i>	76.025	224	2	14.828	218	3
<i>DocuLAN</i>	2.040	34	1	2.380	35	1

T.r. (Tiempo de respuesta en segundos): Tiempo que transcurre desde que se libera el botón del ratón hasta que la imagen se visualiza en la pantalla en su nueva ubicación.

Tabla 6.4. Resultados de la Prueba 4.

Prueba 5

Aplicación: *Corel 5.0*

Descripción: Mover todos los objetos del archivo *5calogo.cdr* incluido en el subdirectorio *venturaltypeset* de la aplicación *Corel 5.0*. Este gráfico ocupa 578.670 bytes. En el estado inicial, están seleccionados todos los objetos del gráfico y la prueba consiste en mover toda la figura hacia un extremo de la ventana mediante la pulsación continua del ratón. El zoom está ajustado al tamaño del dibujo.

Acción realizada por el Moderador:

	Moderador			Participante		
	Bytes	Paquetes	T.r.	Bytes	Paquetes	T.r.
<i>FarSite</i>	161.593	618	6	20.280	338	25
<i>ProShare</i>	156.669	400	4	24.960	390	5
<i>DocuLAN</i>	2.992	44	3	2.640	44	3

Acción realizada por el Participante:

	Moderador			Participante		
	Bytes	Paquetes	T.r.	Bytes	Paquetes	T.r.
<i>FarSite</i>	118.114	417	7	12.066	194	25
<i>ProShare</i>	136.860	379	5	24.302	370	6
<i>DocuLAN</i>	1.560	26	3	1.768	26	3

T.r. (Tiempo de respuesta en segundos): Tiempo que transcurre desde que se libera el botón del ratón hasta que la imagen se visualiza en la pantalla en su nueva ubicación.

Tabla 6.5. Resultados de la Prueba 5.

Prueba 6**Aplicación:** *MS-Paintbrush***Descripción:** Mover el gráfico *house1.bmp*. El archivo ocupa 116.598 bytes. En el estado inicial, está seleccionado todo el dibujo y la prueba consiste en moverlo hacia un extremo de la ventana mediante la pulsación continua del ratón.**Acción realizada por el Moderador:**

	Moderador			Participante		
	Bytes	Paquetes	T.r.	Bytes	Paquetes	T.r.
<i>FarSite</i>	335.270	667	15	30.300	505	45
<i>ProShare</i>	262.203	631	2	40.577	636	2
<i>DocuLAN</i>	1.904	28	0	1.680	28	1

Acción realizada por el Participante:

	Moderador			Participante		
	Bytes	Paquetes	T.r.	Bytes	Paquetes	T.r.
<i>FarSite</i>	284.828	559	12	26.216	433	45
<i>ProShare</i>	164.644	425	2	28.101	428	2
<i>DocuLAN</i>	1.380	23	1	1.728	25	0

T.r. (Tiempo de respuesta en segundos): Tiempo que transcurre desde que se libera el botón del ratón hasta que la imagen se visualiza en la pantalla en su nueva ubicación.

Tabla 6.6. Resultados de la Prueba 6.

Prueba 7

Aplicación: *MS-Paintbrush*

Descripción: Dibujar un gráfico en un documento vacío. Las operaciones que se realizan son las siguientes: 1) Dibujar dos elipses. 2) Rellenar ambas con diferentes colores. 3) Dibujar un rectángulo que contenga ambas elipses. 4) Cambiar el estilo de la fuente en *Texto-Fuentes*. 5) Escribir en la parte inferior del gráfico el siguiente texto: *Prueba de tráfico de red*.

Acción realizada por el Moderador:

	Moderador			Participante		
	Bytes	Paquetes	T.r.	Bytes	Paquetes	T.r.
<i>FarSite</i>	50.155	244	1	12.420	207	2
<i>ProShare</i>	93.320	628	0	40.582	652	0
<i>DocuLAN</i>	48.960	720	0	43.200	720	1

Acción realizada por el Participante:

	Moderador			Participante		
	Bytes	Paquetes	T.r.	Bytes	Paquetes	T.r.
<i>FarSite</i>	45.551	315	1	30.496	274	2
<i>ProShare</i>	109.609	1.066	0	80.751	1.072	1
<i>DocuLAN</i>	37.140	619	1	42.092	619	0

T.r. (Tiempo de respuesta en segundos): Tiempo que transcurre desde que se pulsa el botón *Aceptar* del último diálogo *Texto-Fuentes* hasta que se visualiza completamente la figura en la pantalla.

Tabla 6.7. Resultados de la Prueba 7.

Prueba 8

Aplicación: MS-Notepad

Descripción: Escribir en un archivo nuevo el siguiente texto:

GROUPWARE: Sistemas que dan soporte a grupos de personas involucradas en una tarea común (o en un objetivo común) y que proporcionan una interfaz a un entorno de trabajo compartido.

Acción realizada por el Moderador:

	Moderador			Participante		
	Bytes	Paquetes	T.r.	Bytes	Paquetes	T.r.
<i>FarSite</i>	42.692	241	0	9.600	160	0
<i>ProShare</i>	52.452	491	0	31.876	492	0
<i>DocuLAN</i>	25.160	370	0	22.200	370	0

Acción realizada por el Participante:

	Moderador			Participante		
	Bytes	Paquetes	T.r.	Bytes	Paquetes	T.r.
<i>FarSite</i>	51.353	381	0	23.702	295	1
<i>ProShare</i>	76.279	870	0	62.602	870	0
<i>DocuLAN</i>	22.200	370	0	25.160	370	0

T.r. (Tiempo de respuesta en segundos): Tiempo que transcurre desde que se pulsa la última tecla del texto hasta que se visualiza en la pantalla.

Tabla 6.8. Resultados de la Prueba 8.

6.3 Valoración de los resultados

En primer lugar, se observa que el comportamiento de *FarSite* y de *ProShare* presenta algunas diferencias. En general, la aplicación *FarSite* ofrece un mejor rendimiento. *ProShare* cuenta con un serio inconveniente ya que en algunos casos no visualiza de forma completa toda la información en la pantalla del Participante. Al seguir un modelo centralizado, las aplicaciones sobre las que se coopera no se lanzan en el nodo del Participante. Este usuario es consciente de estas aplicaciones al recibir las ventanas que la componen. Al parecer *ProShare* utiliza algún algoritmo para reducir este volumen de datos de la transmisión que no resulta muy eficaz. Por ejemplo, no se visualiza el fondo de los diálogos, ni de los menús o no se reproducen todos los colores. En las Pruebas 4 y 5, realizadas con el *Corel*, resulta casi imposible reconocer la figura original en el nodo del Participante. Además esta circunstancia no se acompaña con una reducción del tráfico de red ya que en términos globales el volumen de datos que circulan en *ProShare* y en *FarSite* es muy similar. Sumando los datos de las 8 pruebas se transmiten 3.156.620 bytes en *FarSite* y 3.110.572 bytes en *ProShare* como se observa en la Tabla 6.9. La única ventaja significativa de este último sistema es que presenta un menor tiempo de respuesta, pero en una valoración global no es suficiente para mejorar las prestaciones que ofrece *FarSite*.

	Bytes
<i>FarSite</i>	3.156.620
<i>ProShare</i>	3 110.572
<i>DocuLAN</i>	622.428

Tabla 6.9. Tráfico total en las 8 pruebas.

6.3.1 Tráfico de red

Se va a considerar, en primer lugar, el número de paquetes y a continuación el número total de bytes. En esta última medición es donde se aprecian de forma más nítida las diferencias entre las aplicaciones.

a) Tráfico en paquetes

El comportamiento de *DocuLAN* es similar al de *ProShare* respecto al número de paquetes que circulan. En ambos sistemas, el Moderador y el Participante envían, en líneas generales, el mismo número de paquetes. En cambio, en *FarSite*, el número de paquetes que genera el Participante es considerablemente menor que los generados por el Moderador. En la Tabla 6.10 se recogen nuevamente los datos de la Prueba 1.

	Moderador			Participante		
	Bytes	Paquetes	Bytes/Paq	Bytes	Paquetes	Bytes/Paq
Acción realizada por el Moderador:						
<i>FarSite</i>	337.058	1.038	325	33.540	559	60
<i>ProShare</i>	232.033	976	238	62.437	976	64
<i>DocuLAN</i>	25.908	381	68	22.860	381	60
Acción realizada por el Participante:						
<i>FarSite</i>	226.972	823	276	34.797	481	72
<i>ProShare</i>	248.639	1.354	183	93.041	1.350	69
<i>DocuLAN</i>	22.320	372	60	25.296	372	68

Tabla 6.10. Resultados de la Prueba 1 con el tamaño medio de los paquetes.

La explicación de estas diferencias se debe al propio funcionamiento interno del protocolo TCP/IP. Las tres aplicaciones hacen uso del servicio *stream* dentro de este protocolo, a través del cual se implementa un circuito virtual que garantiza que los

paquetes llegan correctamente a su destino. El nodo receptor genera un paquete de reconocimiento notificando que ha recibido correctamente los datos. El protocolo TCP/IP permite que un paquete de reconocimiento pueda indicar la recepción correcta de varios paquetes. El número de paquetes de este tipo que se generan se establece a través de un algoritmo del propio protocolo que depende del tráfico existente [Stevens90].

Cuando la acción es realizada por el Moderador, el Participante únicamente genera paquetes de reconocimiento. En la Tabla 6.10 se observa que, en ese caso, en *FarSite* se genera un paquete de reconocimiento aproximadamente por cada dos paquetes de datos recibidos. En cambio, en *ProShare* y *DocuLAN* se genera un paquete de reconocimiento por cada paquete de datos recibidos.

En la Tabla 6.10 también se ha incluido el tamaño medio de cada paquete en bytes. En el resto de las pruebas se obtienen valores similares. Asimismo, en la Tabla 6.11 se indica de manera ordenada la información que transmite cada usuario en cada una de las pruebas, diferenciándose los pasos que se llevan a cabo en *FarSite* y *ProShare* por un lado y en *DocuLAN* por otro.

Cuando la acción es realizada por el Moderador, el Participante únicamente transmite paquetes de reconocimiento, cuyo tamaño es muy similar en las 3 aplicaciones (60, 64 y 60 bytes en la Tabla 6.10). En esta misma acción el tamaño de los paquetes generados por el Moderador en *FarSite* (325 bytes) y en *ProShare* (238 bytes) es considerablemente mayor que en *DocuLAN* (68 bytes). Esta diferencia se debe a que *DocuLAN* transfiere en cada paquete un único evento. *DocuLAN* ofrecería un comportamiento aún mejor si se introdujera un mecanismo mediante el cual se enviaran varios eventos en cada paquete. Este mecanismo originaría un menor número de paquetes de eventos y de reconocimiento.

FarSite y ProShare

	MODERADOR	PARTICIPANTE
Acción realizada por Moderador	1. Transfiere estado de salida de las ventanas.	
		2. Transfiere mensajes de reconocimiento.
Acción realizada por Participante		1. Transfiere eventos de entrada.
	2. Transfiere mensajes de reconocimiento.	
	3. Transfiere estado de salida de ventanas.	
		4. Transfiere mensajes de reconocimiento.

DocuLAN

	MODERADOR	PARTICIPANTE
Acción realizada por Moderador	1. Transfiere eventos de entrada.	
		2. Transfiere mensajes de reconocimiento.
Acción realizada por Participante		1. Transfiere eventos de entrada.
	2. Transfiere mensajes de reconocimiento	

Tabla 6.11. Acciones realizadas por cada Aplicación.

Cuando la acción es realizada por el Participante el tamaño medio de los paquetes que genera este usuario se incrementa en las tres aplicaciones respecto a la acción anterior (72, 69 y 68 bytes en la Tabla 6.10). Este hecho se debe a que el Participante genera además los eventos de entrada cuyo tamaño es mayor que los de reconocimiento. En *FarSite y ProShare*, el Participante recibe a continuación el estado de las ventanas de salida y por tanto debe enviar los paquetes de reconocimiento. En esta acción, el Moderador en *DocuLAN* genera únicamente mensajes de reconocimiento

y su tamaño (60 bytes) coincide exactamente con el tamaño de los paquetes del Participante en la acción anterior. En *FarSite* y *ProShare*, el Moderador en primer lugar genera mensajes de reconocimiento al ir recibiendo los eventos de entrada y a continuación transfiere los paquetes con el estado de las ventanas de salida. Al ser el tamaño de los paquetes de reconocimiento bastante reducido hace que el tamaño medio (275 y 183 bytes) sea inferior respecto a la acción anterior (325 y 238 bytes).

b) Tráfico en bytes

En los resultados de la Prueba 1 (Tabla 6.10) se aprecia que, independientemente de quién realice la acción, las mayores diferencias respecto a *DocuLAN* se producen en los datos transmitidos por el Moderador. En *FarSite* y en *ProShare* el Moderador siempre debe enviar el estado de las ventanas, mientras que en *DocuLAN* envía los eventos de entrada o únicamente mensajes de reconocimiento. Las diferencias entre estos dos volúmenes de información, el estado de las ventanas y los eventos de entrada, van a marcar las ventajas que ofrece *DocuLAN* respecto a los otros dos productos.

En la Tabla 6.12 se recogen de forma global los resultados de los ocho pruebas realizadas. También se incluye el porcentaje de reducción del tráfico de red entre *DocuLAN* y cada uno de los productos *FarSite* y *ProShare*.

En la primera columna se indican las aplicaciones, el número de la prueba (P1,P2, etc.) y quién realiza la acción, el Moderador (M) o el Participante (P). Las dos columnas siguientes reflejan la cantidad de información en bytes que generan el Moderador y el Participante en cada una de las operaciones. En las columnas cuarta y quinta se señala el porcentaje de mejora de *DocuLAN* respecto a *FarSite* y *ProShare*. Por ejemplo el primer valor 92,3% significa que se produce una reducción del 92,3% en el tráfico de red al realizar la misma operación con *DocuLAN* en vez de con *FarSite*. En la siguiente columna se incluye el total de bytes transmitidos conjuntamente por el Moderador y por el Participante en las dos operaciones. Es decir, corresponde al sumatorio de los cuatro valores que se obtienen por cada aplicación en cada prueba. La

siguiente columna refleja el porcentaje de mejora de *DocuLAN* tomando como referencia estos valores totales. En las dos últimas columnas se indica el tiempo de respuesta de cada acción.

	Moderador Bytes	Particip. Bytes	Moder. Mejora	Particl. Mejora		Total Bytes	Total Mejora	Mod. T. Resp. (sg)	Part.
P1-M	<i>MS-Word: texto</i>				P1				
FarSite	337.058	33.540	92,3%	31,8%		632.367	84,8%	0	0
ProShare	232.033	62.437	88,8%	63,4%		636.150	84,8%	0	0
DocuLAN	25.908	22.860				96.384		0	0
P1-P									
FarSite	226.972	34.797	90,2%	27,3%				0	1
ProShare	248.639	93.041	91,0%	72,8%				0	1
DocuLAN	22.320	25.296						0	0
P2-M	<i>MS-Word: scroll</i>				P2				
FarSite	144.212	12.720	99,0%	90,6%		346.576	98,5%	4	6
ProShare	52.695	11.565	97,3%	89,6%		151.072	96,7%	3	3
DocuLAN	1.402	1.200				5.048		2	2
P2-P									
FarSite	168.178	21.466	99,3%	93,9%				8	10
ProShare	70.107	16.705	98,4%	92,2%				3	4
DocuLAN	1.140	1.306						2	2
P3-M	<i>Corel: dibujar</i>				P3				
FarSite	368.727	78.120	83,1%	30,3%		784.090	70,7%	3	6
ProShare	255.104	72.528	75,5%	24,9%		710.205	67,6%	1	4
DocuLAN	62.474	54.480				229.948		1	1
P3-P									
FarSite	261.568	75.675	79,8%	20,4%				4	6
ProShare	264.220	118.353	80,0%	49,1%				1	5
DocuLAN	52.740	60.254						1	1
P4-M	<i>Corel: eye.cdr</i>				P4				
FarSite	62.428	10.860	95,9%	79,0%		138.951	93,3%	2	7
ProShare	118.415	18.090	97,8%	87,4%		227.358	95,9%	2	3
DocuLAN	2.584	2.280				9.284		1	1
P4-P									
FarSite	57.504	8.159	96,5%	70,8%				3	8
ProShare	76.025	14.828	97,3%	83,9%				2	3
DocuLAN	2.040	2.380						1	1

	Moderador Bytes	Particip. Bytes	Moder. Mejora	Particip. Mejora		Total Bytes	Total Mejora	Mod. T. Resp. (sg)	Part.
P5-M	<i>Corel: Scalogo.cdr</i>				P5				
FarSite	161.593	20.260	98,1%	87,0%		312.053	97,1%	6	25
ProShare	156.669	24.960	98,1%	89,4%		342.791	97,4%	4	5
DocuLAN	2.992	2.640				8.960		3	3
P5-P									
FarSite	118.114	12.066	98,7%	85,3%				7	25
ProShare	136.860	24.302	98,9%	92,7%				5	6
DocuLAN	1.560	1.768						3	3
P6-M	<i>Pbrush: housa1.bmp</i>				P6				
FarSite	335.270	30.300	99,4%	94,5%		676.614	99,0%	15	45
ProShare	262.203	40.577	99,3%	95,9%		495.525	98,6%	2	2
DocuLAN	1.904	1.680				6.692		0	1
P6-P									
FarSite	284.828	26.216	99,5%	93,4%				12	45
ProShare	164.644	28.101	99,2%	93,9%				2	2
DocuLAN	1.380	1.728						1	0
P7-M	<i>Pbrush: dibujar</i>				P7				
FarSite	50.155	12.420	2,4%	-247,8%		138.622	-23,6%	1	2
ProShare	93.320	40.582	47,5%	-6,5%		324.262	47,1%	0	0
DocuLAN	48.960	43.200				171.392		0	1
P7-P									
FarSite	45.551	30.496	18,5%	-38%				1	2
ProShare	109.609	80.751	66,1%	47,9%				0	1
DocuLAN	37.140	42.092						1	0
P8-M	<i>MS-Notepad: texto</i>				P8				
FarSite	42.692	9.600	41,1%	-131,3%		127.347	25,6%	0	0
ProShare	52.452	31.876	52,0%	30,4%		223.209	57,6%	0	0
DocuLAN	25.160	22.200				94.720		0	0
P8-P									
FarSite	51.353	23.702	56,8%	-6,2%				0	1
ProShare	76.279	62.602	70,9%	59,8%				0	0
DocuLAN	22.200	25.160						0	0

Tabla 6.12. Resultados globales de las pruebas.

Los resultados de las 6 primeras pruebas son bastante homogéneos respecto al tráfico de red. Considerando la columna de totales por cada prueba, se observa como la reducción en la aplicación *FarSite* va desde el 70,7% en la Prueba 3 hasta el 99% en la Prueba 6. En la aplicación *ProShare*, el margen abarca desde el 67,6% hasta el 98,6% en las mismas pruebas que en la anterior. Dentro de este conjunto de datos bastante homogéneo los resultados más bajos, por debajo de un 90% de reducción, se producen en las Pruebas 1 y 3. En la Prueba 1 se escribe un texto y en la 3 se dibuja un sencillo gráfico, lo que provoca que la relación entre el volumen de datos de salida (las ventanas) y el volumen de datos de entrada (los eventos) sea más pequeña que en las otras 4 pruebas.

También se observa como en las 6 primeras pruebas la reducción del tráfico de red es más importante en los bytes transmitidos por el Moderador que en los transmitidos por el Participante. Un ejemplo significativo se da en la Prueba 1. En la primera fila, en el Moderador hay una reducción de 92,3% y en el Participante del 31,8%. Esta diferencia se debe a que en *FarSite* y *ProShare* siempre es el Moderador, independientemente de quien realice la acción, el que transmite el estado de salida de las ventanas lo que provoca un mayor volumen de información. En las 3 aplicaciones, el Participante se dedica a transmitir eventos de entrada y mensajes de reconocimiento (Tabla 6.11), lo que provoca un volumen de información mucho más reducido, con lo que resulta más difícil establecer diferencias.

Las pruebas 4 y 5 son similares. En ambas se mueve un gráfico desde la aplicación *CorelDraw*. La única diferencia es el tamaño del fichero que se carga (69.242 y 578.670 bytes respectivamente). Se han escogido estas pruebas para comprobar si las ventajas en la reducción de la información dependen de forma determinante del tamaño del fichero inicial. En los resultados se observa que en la Prueba 5 se genera mayor volumen de datos que en la Prueba 4, pero proporcionalmente esta diferencia es bastante menor que la diferencia inicial del tamaño de los ficheros. Por tanto, el tamaño del fichero inicial influye sólo de forma relativa. El hecho determinante es la información de salida que está compuesta por las ventanas en las que se ha cargado ese fichero.

Esta afirmación también se completa observando la Prueba 6 en la que se mueve una imagen desde la aplicación *MS-Paintbrush*. El tamaño inicial del archivo es de 116.598 bytes, un valor intermedio entre las imágenes de las Pruebas 4 y 5. En cambio, el volumen de información que se transmite es bastante superior al de la Prueba 5. Por tanto, el volumen de información de salida que se genera depende de forma directa de la aplicación que se está utilizando.

Las Pruebas 7 y 8 presentan algunos resultados en los que no existe una reducción del tráfico de red en *DocuLAN* respecto a los otros dos productos, *FarSite* y *ProShare*.

Inicialmente se observa que estos resultados se producen en acciones que generan un volumen de tráfico muy reducido y en aplicaciones sencillas, como *MS-Paintbrush* y *MS-Notepad*, en las cuales se consumen escasos recursos. Por ejemplo, las Pruebas 3 y 7 son similares en cuanto a la acción que se realiza. Se diferencian en la aplicación que se utiliza, *CorelDraw* en la Prueba 3 y *MS-Paintbrush* en la Prueba 7. En cambio, la diferencia de volumen de información que se genera en cada una de ellas es considerable (784.090 frente a 138.622 bytes en *FarSite*). Lo mismo ocurre entre las Pruebas 1 y 8 realizadas con *MS-Word* y *MS-Notepad* respectivamente (632.367 y 127.347 bytes).

También se aprecia cómo los resultados negativos únicamente se producen en la columna que recoge los datos transmitidos por el Participante. En la columna del Moderador se producen valores positivos para *DocuLAN*, pero significativamente menores que en el resto de pruebas. La causa de estos resultados es la menor diferencia existente entre el volumen de datos de salida y el volumen de los eventos de entrada que los generan. El volumen de salida es más pequeño porque se realizan acciones que no implican un gran volumen de información y se hace uso de aplicaciones con una sencilla interfaz de usuario.

A pesar de la sencillez de las Pruebas 7 y 8 en ningún caso el volumen de los eventos de entrada supera al volumen de salida con lo que la aplicación *DocuLAN*

ofrecerá siempre para el usuario Moderador un mejor rendimiento y un menor tráfico de red. La relación entre los dos volúmenes de información se puede observar directamente en las acciones realizadas por el Participante (P7-P y P8-P en la Tabla 6.12). El Participante genera los eventos de entrada (30.496 bytes en *FarSite* en P7-P) y el Moderador genera la información de salida (45.551 bytes en la misma acción). En ambas cifras se incluyen paquetes de reconocimiento, pero no influyen de forma determinante en la relación. Por tanto, el volumen de información de salida es siempre mayor que el volumen de información de entrada. La diferencia es de 15.055 bytes, más pequeña que en el resto de las pruebas, pero siempre es un valor positivo.

En ambas pruebas, 7 y 8, las diferencias que se producen en los bytes transmitidos por el Participante en la aplicación *FarSite* respecto a *DocuLAN* son en algunos casos muy importantes en porcentaje (247,8% y 131,3%) y favorables a la aplicación *FarSite*. Estos resultados aunque significativos tienen una relevancia pequeña y no modifican las conclusiones obtenidas tras el estudio de las pruebas anteriores por las siguientes razones:

1. Se produce en volúmenes pequeños de información ya que el Participante únicamente transmite los eventos de entrada y mensajes de reconocimiento.
2. Los bytes transmitidos por *DocuLAN* desde el Participante se podrían reducir de forma significativa si se utilizara un mecanismo de empaquetamiento de eventos. Este mecanismo reduciría de forma significativa tanto el número de paquetes de eventos como de reconocimiento.

En general, se observa que la reducción del tráfico de red que proporciona *DocuLAN* es muy importante respecto al comportamiento de *FarSite* y *ProShare*. La dimensión de esta reducción depende de forma directa de la diferencia entre el volumen de información de salida (formado por las ventanas) y el volumen de información de entrada (generado por los eventos). Cuanto mayor sea esta diferencia, mejor será el comportamiento de *DocuLAN* respecto a las otras dos aplicaciones. En principio, hay más probabilidades que estas diferencias sean mayores en las aplicaciones gráficas, pero

depende de forma más determinante de las operaciones que se realicen con las aplicaciones.

6.3.2 Tiempo de respuesta

El tiempo de respuesta va a depender de la infraestructura de red que se utilice y del volumen de datos que se genere en cada operación. En general, en un sistema replicado el usuario no percibirá un tiempo de respuesta mayor al que se produciría si estuviera trabajando de forma individual con las mismas aplicaciones. En estos sistemas la salida se genera localmente procesando los eventos que se reciben. El único tiempo de espera se debe a la transmisión de los eventos de entrada por la red a cada nodo que participa en una sesión cooperativa.

En cambio, la arquitectura centralizada implica un mayor tiempo de respuesta ya que, para ver reflejado el resultado de una acción, se debe transmitir el estado de las ventanas de salida lo que implica un mayor volumen de información.

Los valores del tiempo de respuesta de cada acción también se incluyen en la Tabla 6.12. Inicialmente se observan unas diferencias importantes entre *FarSite* y *ProShare*. Los tiempos de respuesta de *ProShare*, aunque algo superiores, son en general similares a los que ofrece *DocuLAN* y aceptables para cualquier usuario que adopte este sistema. Pero como ya se ha mencionado cualquier ventaja de *ProShare* queda contrarrestada por el importante inconveniente de no reproducir de forma exacta el estado de las ventanas en el Participante, lo que provoca que no se pueda identificar la información en este nodo.

FarSite tiene unos tiempos de respuesta más altos. Las diferencias mayores se producen en las Pruebas 5 y 6 (25 y 45 sg. en la Tabla 6.12) en las que hay que transmitir gran cantidad de información en un espacio corto de tiempo. La operación que se realiza en esta prueba dura aproximadamente 1 segundo y a continuación se debe transmitir la información de salida al Participante. Este comportamiento de *FarSite*

resulta muy poco aceptable para el usuario final que no comprenderá un retardo tan grande para ver los resultados de la aplicación.

En las Pruebas 2, 3 y 4 los tiempos de respuesta son también más altos que en *DocuLAN*, pero se mantienen en un rango aceptable para el usuario final. En el resto de pruebas no hay diferencias significativas ya que el volumen de información a transmitir es más reducido o se transmite en un periodo de tiempo más prolongado.

En los sistemas centralizados la situación más problemática se produce cuando el Participante posee el testigo y realiza alguna operación. Al ser plenamente consciente de la acción realizada espera que su resultado pueda ser visualizado inmediatamente. Pero hasta llegar a ese resultado final se deben realizar tres acciones:

1. Los eventos deben ser transmitidos al Moderador.
2. Se deben reproducir dichos eventos en el Moderador.
3. Se debe transmitir la salida de la aplicación a los Participantes.

Todo este proceso provocará un retardo que dependerá fundamentalmente del volumen de información de salida. En esta situación, el Participante es consciente de que ha generado unas entradas y esperará ver los resultados de forma inmediata. Si dicho usuario no es un experto en el sistema y debido al retraso puede llegar a pensar que se ha producido algún error e intentará repetir de nuevo la operación con lo que se incrementará el retardo final y se pueden producir resultados inesperados.

En estos momentos existen en el mercado numerosos sistemas de videoconferencia de sobremesa, fundamentalmente sobre RDSI, que integran la comunicación de vídeo y audio con sistemas de conferencia de documentos sobre una arquitectura centralizada [Labriola95] [Taylor95]. En ellos al existir una limitación sobre la capacidad del canal de comunicaciones que se dedica a la conferencia de documentos se obtienen comportamientos poco eficaces en el tiempo de respuesta que dificultan en gran medida la cooperación.

En general, los sistemas centralizados pueden producir situaciones en las que el tiempo de respuesta resulta poco aceptable, especialmente cuando el Participante es el usuario activo.

7. Discusión

En el capítulo anterior se ha realizado una evaluación cuantitativa de dos aspectos importantes, el tráfico de red y el tiempo de respuesta, en los que el sistema *DocuLAN* de arquitectura replicada ha mostrado un comportamiento superior al de los sistemas centralizados. En este capítulo se va a comparar *DocuLAN* con otros sistemas, especialmente con los que siguen un modelo replicado. Inicialmente se constatan dos cuestiones importantes:

- Todas las referencias de sistemas replicados corresponden a sistemas desarrollados en el entorno *UNIX*.
- Existen muy pocas fuentes sobre análisis cuantitativos del comportamiento de los sistemas replicados existentes. Esta misma afirmación la realiza también Ahuja al exponer los resultados del estudio del sistema *Rapport* [Ahuja90].

7.1 Comparación con otros sistemas de arquitectura replicada

En primer lugar, se va a presentar la evaluación realizada sobre el sistema *Rapport*. Este estudio es el que posee mayores similitudes con las pruebas realizadas en

DocuLAN. A continuación se comentarán otras evaluaciones más generales de otros sistemas replicados.

7.1.1 Sistema *Rapport*

En un trabajo realizado en 1990 en los laboratorios AT&T Bell, Ahuja presenta el sistema *Rapport* y realiza una comparación entre una versión que sigue el modelo centralizado y otra versión que sigue el modelo replicado desarrolladas ambas sobre el entorno *X-Window* [Ahuja90]. El autor también menciona la falta de otras fuentes que contengan estudios similares.

El sistema *Rapport* gestiona el desarrollo de una reunión a través de dos mecanismos:

- Un mecanismo de control que permite iniciar, desarrollar y finalizar una reunión.
- Un mecanismo que permite visualizar las aplicaciones con la misma información en cada uno de los equipos que participan en la reunión.

Rapport es un sistema especialmente interesante ya que existe una versión bajo un modelo de arquitectura replicada y otra sobre un modelo de arquitectura centralizada. Ambas se ejecutan sobre un entorno *UNIX* con *X-Window* y las pruebas se han realizado sobre una red Ethernet de 10 Mb. El sistema *Rapport* consta de un Gestor de Conferencia que se encarga de controlar las comunicaciones entre el servidor *X-Window* y las aplicaciones cooperativas.

En primer lugar se desarrolló la versión que sigue un modelo de arquitectura replicada. Se denominó *MULTI* y se muestra en la Figura 7.1. En esta versión se ejecutan en cada nodo el servidor *X-Window* (*X-Server*), el Gestor de Conferencia (*C-Mgr*) y las aplicaciones con las que se coopera (*P1* y *P2*). Estas aplicaciones cooperativas se comunican con el servidor *X-Window*, el servidor *X-Window* con el Gestor de Conferencia y éste con los Gestores de Conferencia del resto de los nodos. El

servidor *X-Window* ha sido modificado para intercambiar mensajes con el Gestor de Conferencia y permitir así filtrar y enviar las entradas de las aplicaciones. En la Figura 7.1 se muestra la secuencia de operaciones producida por una entrada para la aplicación *P1* en el nodo *B*. El Gestor de Conferencia detecta que ese nodo posee el testigo de la sesión y dirige la entrada al servidor *X-Window*, el cual la envía a la aplicación *P1*. Asimismo, el Gestor de Conferencia envía el evento de entrada a los Gestores de Conferencia de los nodos *A* y *C*. En cada nodo el evento pasa al servidor *X-Window* y de ahí a la aplicación *P1*. Las salidas se gestionan a través de cada servidor *X-Window* local.

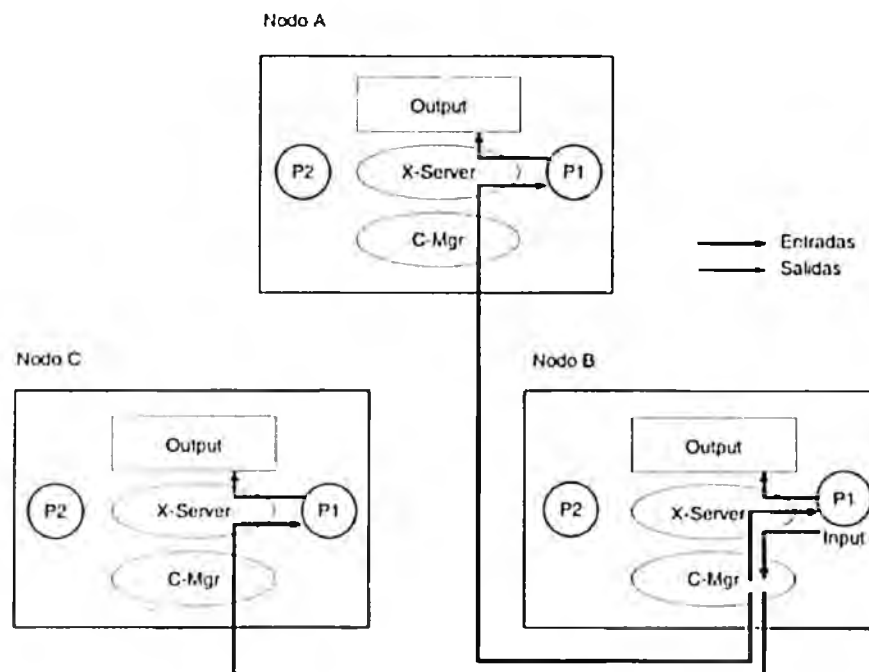


Figura 7.1. Versión *MULTI* del sistema *Rapport*.

Posteriormente se desarrolló una versión de *Rapport* bajo un modelo centralizado. Se denominó *SINGLE* y se muestra en la Figura 7.2. En cada nodo se ejecuta un servidor *X-Window* y un Gestor de Conferencia, pero las aplicaciones con las que se coopera únicamente se lanzan en un nodo. Estas aplicaciones no se comunican con el servidor *X-Window*, sino con el Gestor de Conferencia que asume las funciones

del primero. En esta versión el servidor *X-Window* no se ha modificado. En la Figura 7.2 se muestra la secuencia de una entrada producida en el nodo *B* para la aplicación *P1* que se ejecuta en el nodo *A*. El Gestor de Conferencia del nodo *B*, que posee el testigo, envía el evento de entrada al gestor del nodo *A* para que pueda ser procesado por la aplicación *P1*. La salida que se produce se envía al gestor, el cual la dirige al servidor *X-Window* local y a los gestores de los nodos *B* y *C*. En estos nodos se realiza una adaptación a los recursos propios y se envía la salida al servidor *X-Window* local.

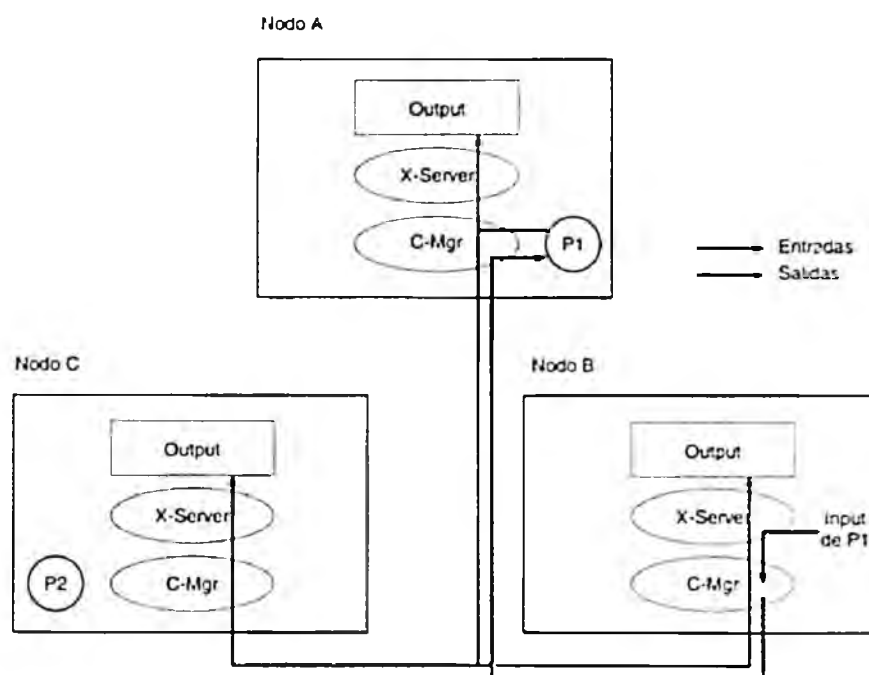


Figura 7.2. Versión *SINGLE* del sistema *Rapport*.

Para evaluar el comportamiento de ambas versiones, *MULTI* y *SINGLE*, Ahuja consideró los parámetros del tráfico de mensajes y el tiempo de respuesta. Para ello utiliza dos conceptos: los *mensajes entre nodos* y la *trayectoria de Entrada/Salida*. En *DocuLAN* también se han utilizado los parámetros del tráfico de mensajes y el tiempo de respuesta.

Los *mensajes entre nodos* corresponden al número total de mensajes que fluyen entre los equipos por cada evento de entrada o de salida en el entorno *X-Window*. Es un indicador del ancho de banda necesario y del tráfico de red que se genera.

La *trayectoria de Entrada/Salida* es el máximo número de mensajes que pueden circular desde la generación de una entrada hasta la visualización de su salida. Incluye mensajes internos que no salen del propio nodo y los mensajes externos que circulan entre los nodos. Este indicador es directamente proporcional al tiempo de respuesta.

Ahuja también apunta que es necesario considerar varios aspectos para evaluar el comportamiento de ambas versiones como son la relación entre el volumen de los datos de entrada y los de salida de las aplicaciones o la relación entre el lugar donde se genera la entrada y el lugar donde se procesa. Para ello ideó un escenario común para las pruebas y una relación de acciones que incluyera los aspectos anteriores. Esta secuencia de acciones es la siguiente:

El usuario *A* establece una conferencia con los usuarios *B* y *C*. El usuario *A* inicia una aplicación de edición de textos (*vi*) y realiza unas operaciones. El testigo pasa a *B* que realiza también sus operaciones. *C* inicia una aplicación de edición de gráficos y dibuja una figura. El testigo pasa a *A* quien modifica la figura anterior.

En la Figura 7.3, se describe el recorrido que realiza un evento de entrada y una salida en la versión *MULTI*.

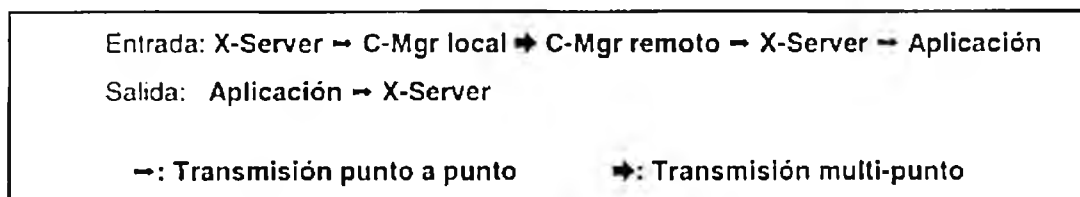


Figura 7.3. Secuencia de un evento en la versión *MULTI*.

El número de *mensajes entre nodos* es N , siendo N el número de nodos que participan en la conferencia. Estos mensajes se originan en la transmisión multi-punto de la entrada. La salida es procesada localmente y por tanto no genera mensajes de este tipo. La *trayectoria de Entrada/Salida* es de 5 mensajes, que corresponden con las 4 flechas que aparecen en la secuencia de la entrada y la única de la secuencia de la salida.

Por su parte, el camino que describen las entradas y las salidas en la versión *SINGLE* se muestra en la Figura 7.4. Se considera el camino más largo posible que se produce cuando la entrada se genera en un nodo remoto respecto al que ha lanzado la aplicación.

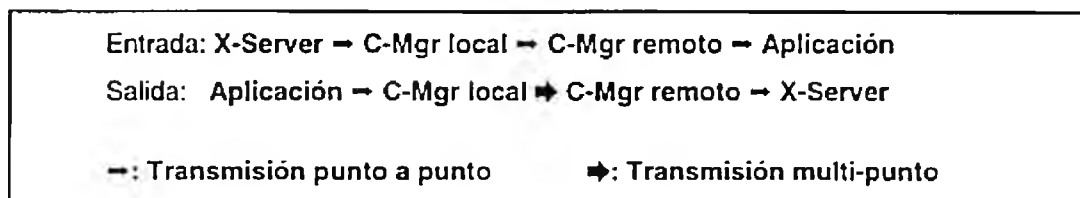


Figura 7.4. Secuencia de un evento en la versión *SINGLE*.

La secuencia de entrada de la versión *SINGLE* se ve afectada por la variación en la arquitectura respecto a la versión *MULTI*, ya que el Gestor de Conferencia (*C-Mgr*) en la versión *SINGLE* envía el evento directamente a la aplicación en lugar de a través del *X-Server*.

Se genera un *mensaje entre nodos* para la secuencia de entrada, el que se produce desde el nodo que genera la entrada al que contiene la aplicación. En la salida se generan N *mensajes entre nodos*, uno por cada participante de la conferencia. La *trayectoria de Entrada/Salida* es de 6 mensajes, 3 para la entrada y 3 para la salida (uno por cada flecha de la figura).

Ahuja destaca que se produce una diferencia importante entre ambas versiones en el *trayectoria de Entrada/Salida*, aunque el valor que se obtiene es bastante similar, 5 en *MULTI* y 6 en *SINGLE*. Pero en este último caso existen 2 mensajes externos que se transmiten entre los nodos, mientras que en *MULTI* sólo existe 1 mensaje externo.

Asimismo Ahuja presenta unas mediciones cuantitativas del tráfico de mensajes y del tiempo de respuesta. Los mensajes son clasificados en tres tipos:

- Mensajes de administración: aquellos relacionados con el establecimiento de la conferencia y la gestión de los participantes.
- Mensajes de entrada: aquellos que transmiten un evento de entrada hasta la aplicación correspondiente.
- Mensajes de salida: transmiten una solicitud de salida desde la aplicación hasta el servidor *X-Window*.

También Ahuja considera el empaquetamiento de los eventos y presenta los resultados obtenidos con ambas posibilidades, sin mecanismo y con mecanismo de empaquetamiento.

En la Tabla 7.1 se reflejan los mensajes que se generan con las versiones *MULTI* y *SINGLE* al realizar la secuencia de acciones descrita anteriormente sin introducir ningún mecanismo de empaquetamiento. Por tanto, en cada mensaje se envía un único evento.

Versión	Entrada	Salida	Admin.	Total	Ratio
<i>MULTI</i>	538 (85%)	0 (0%)	98 (15%)	636	1
<i>SINGLE</i>	465 (11%)	3.402 (87%)	98 (2%)	3.875	6,09

Tabla 7.1. Mensajes generados en *Rapport* sin mecanismo de empaquetamiento.

Los mensajes administrativos suman el mismo número en ambas versiones y en ningún caso inciden de forma importante en la cantidad de mensajes que circulan (el 15% en *MULTI*). El dato más relevante corresponde a los mensajes de salida de *SINGLE* (3.402, el 87%). El ratio entre el volumen de salida y el de entrada en esta versión alcanza 7,3 a 1, es decir, por cada mensaje de entrada se generan 7,3 mensajes de salida. En cambio en *MULTI* no se generan mensajes de salida. Los resultados también indican que el ratio entre el volumen total de *SINGLE* y de *MULTI* es de 6 a 1. Ahuja indica que en aplicaciones de edición de textos que no utilicen gráficos este ratio puede descender a 4 a 1.

En la Tabla 7.2 se presentan los resultados obtenidos realizando un empaquetamiento de los mensajes. Este mecanismo reduce especialmente los mensajes de salida ya que su frecuencia de generación es alta. El número de mensajes de entrada no se reduce de forma considerable ya que se generan de forma más aislada.

Versión	Entrada	Salida	Admin.	Total	Ratio
<i>MULTI</i>	455 (82%)	0 (0%)	98 (18%)	553	1
<i>SINGLE</i>	435 (22%)	1.478 (73%)	98 (5%)	2.011	3,6

Tabla 7.2. Mensajes generados en *Report* con mecanismo de empaquetamiento.

En esta implementación, en la versión *SINGLE*, el ratio entre el volumen de salida y el de entrada es de 3,4 a 1. El ratio entre el volumen total de mensajes entre *SINGLE* y *MULTI* es de 3,6 a 1. A partir de estos datos, Ahuja concluye que es necesario contar con una red que posea una capacidad de transmisión 3,5 veces mayor para la versión *SINGLE* que para la versión *MULTI*.

El tiempo de respuesta es inmediato en ambas versiones. Existe un ligero aumento en la versión *SINGLE*, pero no resulta significativo, tal y como se observa en la Tabla 7.3. Se presentan por separado los datos obtenidos con aplicaciones de edición de

textos y aplicaciones gráficas. También se indican los datos medios obtenidos, que en todos los casos es de 0,1 segundos.

Versión	Máximo (texto)	Máximo (gráficos)	Medio (texto)	Medio (gráficos)
<i>MULTI</i>	0,5	1,1	0,1	0,1
<i>SINGLE</i>	0,8	1,3	0,1	0,1

Tabla 7.3. Tiempos de respuesta en segundos en *Rapport*.

Ahuja realiza la comparación entre la versión *MULTI* y *SINGLE* por medio de los ratios entre el número de mensajes total que se generan en cada versión. En el capítulo anterior se detallaban los resultados obtenidos en *DocuLAN* y se indicaba el porcentaje de reducción del tráfico de red que se obtenía al utilizar *DocuLAN* en lugar de las aplicaciones *FarSite* y *ProShare*. Para poder realizar una comparación con el estudio realizado por Ahuja, en la Tabla 7.4 se presentan los ratios entre el volumen total de bytes que se transmiten en cada una de las pruebas. Por ejemplo, el primer valor de 6,56 indica que por cada byte que se transmite en *DocuLAN*, se transmiten 6,56 bytes en *FarSite*. No se incluye una comparación entre el volumen de mensajes o paquetes ya que en *DocuLAN*, a diferencia de *FarSite* y *ProShare*, no utiliza un mecanismo de empaquetamiento de eventos.

	<i>FarSite</i>	<i>ProShare</i>	<i>DocuLAN</i>
P1	6,56	6,60	1
P2	68,66	29,93	1
P3	3,41	3,09	1
P4	14,97	24,49	1
P5	34,83	38,26	1
P6	101,11	74,05	1
P7	0,81	1,89	1
P8	1,34	2,36	1
Total	5,07	5,00	1

Tabla 7.4. Ratios de volumen de información en *FarSite*, *ProShare* y *DocuLAN*.

En cada prueba se realizaban diferentes tipos de operaciones lo que explica las variaciones de los ratios de unas pruebas a otras. Si se realiza una comparación entre el volumen total de información que se transmiten en las 8 pruebas, en la que de alguna forma se compensan las operaciones con textos y gráficos, se obtienen unos ratios (5.07 y 5) similares a los que indica Ahuja.

En general y a pesar de los resultados obtenidos Ahuja opta por la versión *SINGLE* debido a los problemas de inconsistencia que pueden ocurrir en la versión *MULTI* [Ahuja90]. Pero asimismo admite que la preferencia por la versión *SINGLE* tiene dos importantes condicionantes:

- Se realiza en entornos en los que no se utilizan aplicaciones que generen un gran volumen de datos de salida.
- Los problemas de inconsistencia no han sido abordados en profundidad en la versión *MULTI*.

En resumen, Ahuja realiza una comparación entre el modelo centralizado y el replicado a través de una evaluación cuantitativa, similar a la realizada en *DocuLAN*. Los resultados de ambos estudios mantienen una coherencia entre sí. A pesar de ello, Ahuja muestra su preferencia por la versión centralizada. No obstante, su estudio es utilizado por otros autores, como Lauwers [Lauwers90] y como se hace en este trabajo, para demostrar el mejor rendimiento de los sistemas replicados sobre la base de su mejor comportamiento en los parámetros del tráfico de red y el tiempo de respuesta.

7.1.2 Sistema *MERMAID*

El sistema *MERMAID* es un completo sistema de trabajo en grupo, desarrollado por NEC en Japón en 1989. Permite establecer la comunicación a través del vídeo, el audio y documentos multimedia. Se ejecuta bajo el entorno *UNIX* y sobre diferentes tipos de redes como LAN y RDSI. [Watabe90]. Las fuentes consultadas no presentan

una evaluación cuantitativa del mismo, pero resulta especialmente interesante ya que ha sido utilizado en un entorno real de trabajo durante varios años.

MERMAID implementa la compartición de aplicaciones mediante el modelo de arquitectura replicada. Permite utilizar aplicaciones existentes y aplicaciones desarrolladas específicamente para este sistema. Estas últimas cuentan con la ventaja de haber sido diseñadas para ser utilizadas por varios usuarios simultáneamente [Abe93].

Los autores del sistema afirman que la arquitectura replicada utilizada en *MERMAID* proporciona un mejor tiempo de respuesta y un menor tráfico de red. También indican que la principal dificultad radica en el mantenimiento de la consistencia. Este problema se aborda mediante el esquema de control de las aplicaciones cooperativas que se describe en la Figura 7.5 [Abe93].

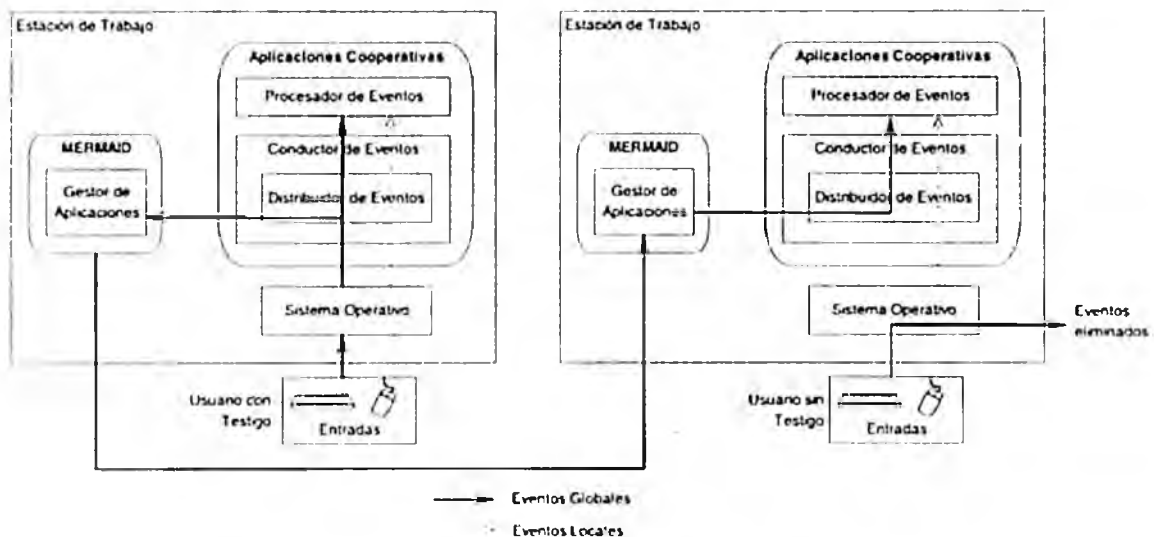


Figura 7.5. Esquema de control de las aplicaciones cooperativas en *MERMAID*.

En este proyecto se ha realizado un análisis sobre cuál debe ser el elemento que distribuya los eventos cooperativos al resto de nodos. Así se apunta que esta operación se podría llevar a cabo por tres entidades:

- Sistema Operativo. Se encarga de depositar los eventos de entrada generados por el usuario en la cola de eventos de cada aplicación.
- Conductor de Eventos. Toma los eventos de la cola de la aplicación y los conduce al Procesador de Eventos.
- Procesador de Eventos. Se encarga de ejecutar los eventos que recibe.

Si el Sistema Operativo realizara la distribución de eventos implicaría una modificación del mismo y ofrecería la ventaja de que las aplicaciones cooperativas no tendrían que ser modificadas.

Si el responsable de la distribución de eventos fuese el Procesador de Eventos implicaría una modificación de cada una de las aplicaciones cooperativas que participasen en una conferencia.

Ambas modificaciones, en el Sistema Operativo o en las aplicaciones, suponen un alto coste y resultan inaceptables. Por tanto, la solución consiste en distribuir los eventos en el Conductor de Eventos. Esto supone realizar una modificación en esta entidad, pero es común a todas las aplicaciones. Este nuevo módulo que incluye las modificaciones necesarias se denomina Distribuidor de Eventos. No obstante, esta modificación implica una recompilación de las aplicaciones para incluir la nueva entidad.

En *DocuLAN* se usa el mismo análisis y se consigue similar resultado sin la necesidad de retocar las aplicaciones mono-usuario, ni el sistema operativo. Se creaba una capa intermedia entre el entorno *MS-Windows 3.x* y las aplicaciones que participaban en la sesión. La capa se basaba en el mecanismo de los filtros que permite capturar los eventos antes de que sean procesados.

Además del Distribuidor de Eventos, *MERMAID* cuenta con otro módulo específico, el Gestor de Aplicaciones, que define la interfaz entre el propio *MERMAID* y las aplicaciones cooperativas. Ambos módulos se muestran en la Figura 7.5.

En el esquema de control de las aplicaciones cooperativas de *MERMAID* se distinguen entre eventos globales y locales. Los primeros corresponden a las entradas generadas por los usuarios y en el caso en que dicho usuario posea el testigo se transmitirán al resto de usuarios. Los eventos locales son generados internamente por el sistema operativo y no se transmiten entre los usuarios.

Los autores del sistema no presentan un estudio cuantitativo sobre su funcionamiento. No obstante, *MERMAID* ha sido evaluado en el trabajo habitual entre 10 centros de investigación, 9 situados en Japón y 1 en Estados Unidos. Se han realizado reuniones técnicas, se ha desarrollado software desde diferentes puntos y se han realizado demostraciones. El resultado, según los autores del estudio, ha sido en todos los casos satisfactorio. También se destaca la gran flexibilidad que proporciona para que un grupo de personas pueda compartir la información según sus necesidades [Abe93].

7.1.3 Otros sistemas

Existen otros sistemas replicados como *MMConf*, *VConf* y *Dialogo* que presentan unas características muy similares entre sí.

El sistema *MMConf* se desarrolló en la empresa Bolt Beranek and Newman de Massachusetts siguiendo un modelo replicado [Crowley90]. Como los anteriores se ejecuta sobre el entorno *UNIX*. Crowley no presenta un análisis cuantitativo del mismo, pero entre sus conclusiones destaca que los sistemas replicados ofrecen un mejor rendimiento que los sistemas centralizados, especialmente cuando las aplicaciones utilizadas generan un alto volumen de datos de salida. Asimismo afirma que la introducción de mecanismos conscientes de la cooperación permite un funcionamiento robusto del sistema evitando el inconveniente de la pérdida de la consistencia [Crowley90].

Lauwers y Lantz desarrollaron un sistema replicado, denominado *VConf*, en la Universidad de Stanford y posteriormente crearon el sistema *Dialogo* en Olivetti de California. Ambos se ejecutan sobre el entorno *UNIX*. Los autores afirman que la principal ventaja de los sistemas replicados es su rendimiento, motivado por su menor tiempo de respuesta y el bajo tráfico de red que generan respecto a los sistemas centralizados. De igual forma que en el proyecto *MMConf*, los problemas de consistencia son abordados con la introducción de mecanismos conscientes de la cooperación. Lauwers y Lantz concluyen que las ventajas de los sistemas replicados son de la suficiente magnitud como para continuar profundizando en los mismos [Lauwers90].

Todos los autores que han analizado los sistemas replicados coinciden en que la ventaja más destacada es su rendimiento, provocado por el menor tráfico de red que se genera y el menor tiempo de respuesta que posee [Ahuja90] [Lauwers90] [Abe93]. Estas conclusiones coinciden con los resultados obtenidos tras la evaluación de *DocuLAN*.

7.2 **Consistencia en la arquitectura replicada**

El problema de la consistencia es el principal inconveniente de los sistemas replicados [Ahuja90] [Hyland92]. Los autores de *MMConf*, *VConf* y *Dialogo* [Crowley90] [Lauwers90] son los que abordan en mayor profundidad este problema, consiguiendo unos resultados aceptables, que se comentan a continuación.

a) Consistencia en el estado inicial de la aplicación

DocuLAN ofrece un mecanismo para garantizar que las aplicaciones cooperativas se abren con la misma apariencia en todos los nodos. Este mecanismo se basa en la transferencia de los archivos de configuración (normalmente con extensión

ini) desde el Moderador al resto de usuarios. Una vez finalizada la sesión se restauran las opciones de configuración originales en cada nodo.

Cualquier solución a esta problemática dependerá del sistema operativo que se utilice, pero, en este caso, los sistemas existentes sobre *UNIX* utilizan una solución similar a la empleada en *DocuLAN*. Así *MMConf*, *Dialogo* y *MERMAID* también establecen, por cada sesión, unas variables de entorno y unos archivos con opciones de configuración de las aplicaciones que se transfieren a cada uno de los usuarios participantes [Crowley90] [Lauwers90] [Abe93].

Estos mecanismos, tanto los empleados en *DocuLAN* como en los sistemas sobre *UNIX*, ofrecen una solución sólida para asegurar que las aplicaciones se inician con la misma apariencia y por tanto permitir que se pueda iniciar la cooperación mediante la transferencia de eventos.

Además en *DocuLAN* también se permite que los usuarios cuenten con diferentes resoluciones de pantalla. Para ello introduce un mecanismo por el que todos los nodos se adaptan de forma transparente a la resolución menor. En los sistemas sobre *UNIX* no se incluyen mecanismos de este tipo. Únicamente Crowley lo menciona como una posible mejora futura del sistema *MMConf* [Crowley90].

Por otra parte, en *DocuLAN* no se ha introducido ningún procedimiento para verificar que se lanza la misma versión de una aplicación en cada uno de los nodos. El sistema *MMConf* cuenta con un dispositivo de este tipo [Crowley90]. La implementación de un mecanismo similar en *DocuLAN* dependería directamente de la información que facilitase la propia aplicación y que puede ser recuperada a través de funciones de programación. En algunas aplicaciones *MS-Windows* no se proporciona la versión en el nombre asociado al fichero ejecutable. En estos casos no sería posible controlar que en todos los nodos se ha cargado la misma versión de una determinada aplicación.

b) Indeterminismo

Por la naturaleza del problema no es posible crear en ningún entorno un mecanismo general que resuelva esta situación. Es necesario tratar de forma individual cada caso. *DocuLAN* ofrece una solución restrictiva al caso más representativo, el producido por la presión continua de la barra de desplazamiento. Si no se aplicara esta restricción, se originaría el desplazamiento del documento a una posición diferente en cada usuario. Por el mecanismo introducido, la operación de la presión continua se sustituye por una única presión que tiene como resultado el movimiento de una línea.

Esta solución limita la funcionalidad de las aplicaciones, pero en otros sistemas se han introducido mecanismos aun más restrictivos. En *MMConf* simplemente se inhabilitan las acciones que pueden ocasionar situaciones indeterministas [Crowley90]. Crowley también apunta que estos problemas son provocados en muchos casos por errores de la propia aplicación. Lauwers afirma que en general las aplicaciones que muestran estos comportamientos indeterministas no pueden ser utilizadas en un entorno replicado y sólo pueden ser controlados algunos casos concretos [Lauwers90].

En la Universidad de Paisley en Escocia se han realizado trabajos sobre las situaciones indeterministas haciendo uso del mismo módulo de cooperación presentado en este trabajo [Tian95]. Allí han obtenido una solución no restrictiva del problema de la presión continua de la barra de desplazamiento.

Esta solución se basa en el envío de los mensajes de desplazamiento generados en el nodo activo y no sólo los eventos. Como se indicaba en el apartado 5.9.4 en la manipulación de la barra de desplazamiento intervienen tres conceptos: los eventos de pulsación y liberación del ratón, los mensajes de desplazamiento que se generan y el número de líneas que se mueve el documento. Pero con los mensajes de desplazamiento también se producen inconsistencias que es necesario tratar ya que no se genera el mismo número de mensajes por cada pulsación. Por ejemplo, en la aplicación *MS-Notepad* se generan dos mensajes de desplazamiento al realizar una pulsación mientras que en la mayoría de las aplicaciones se genera un único mensaje. Asimismo el número

de líneas que se mueve el documento depende normalmente de forma lineal del número de mensajes generados, pero en algunos casos esta relación no es lineal. Por ejemplo, en *MS-Word* la relación es lineal, sólo si el número de mensajes es menor que 10. Así con 9 mensajes se mueve 9 líneas; pero con 10 mensajes se mueve 14 líneas; y con 14 mensajes, 34 líneas.

Los mensajes de desplazamiento son almacenados en una cola específica en los nodos pasivos y son reproducidos teniendo en cuenta el comportamiento de cada aplicación con los mensajes de desplazamiento. Influye, por una parte, la relación entre el número de pulsaciones y el número de mensajes y, por otra, la relación entre el número de mensajes y el número de líneas que se mueve el documento.

Esta solución introduce un nuevo concepto en el entorno replicado que es el de *Mecanismos Conscientes de la Aplicación*. Cuando los mensajes de desplazamiento llegan al nodo receptor es necesario un proceso de adaptación que depende de la aplicación que se está utilizando. Con este mecanismo se consigue un mayor grado de consistencia a costa de eliminar la característica de un sistema de propósito general para todas las aplicaciones *MS-Windows*. En *DocuLAN* se ha mantenido el propósito general lo que permite hacer uso de cualquier aplicación que se ejecute sobre el entorno *MS-Windows*.

c) Gestión de los recursos externos

En *DocuLAN* los usuarios son responsables de que el acceso a los dispositivos externos, ya sean de entrada (la apertura de un fichero) o de salida (la impresión de un documento) no provoquen la pérdida de la consistencia.

Se ha incluido un mecanismo para abordar la situación más frecuente que consiste en abrir un nuevo archivo en una sesión ya iniciada. Esta operación no se debe realizar a través de la tradicional caja de diálogo *Abrir* sino a través de la ventana de control de la sesión. El nuevo documento introducido será transferido inmediatamente al resto de usuarios para que sea abierto en cada nodo. Siguiendo este procedimiento

también se puede incorporar una nueva aplicación a la sesión. Será necesario, por tanto, instruir a los usuarios para que realicen estas acciones de la forma indicada.

En los sistemas sobre *UNIX* la apertura de un nuevo archivo se puede llevar a cabo mediante el denominado *Directorio Cooperativo* de cada nodo. Cuando se deposita un archivo en uno de estos directorios, el archivo se transfiere al *Directorio Cooperativo* del resto de nodos. Los sistemas *Dialogo* y *MMConf* cuentan con herramientas que copian automáticamente los archivos. En *VConf* esta tarea se debe realizar manualmente [Lauwers90]. Esta solución presenta las limitaciones de que los archivos únicamente se pueden identificar por su nombre y no por la localización completa que puede variar en cada nodo y, además, no se contempla la posibilidad de que un archivo pueda incluir referencias a otros archivos o aplicaciones locales.

Otro caso típico es la orden de impresión de un documento. En *DocuLAN*, como en el resto de sistemas replicados, esta acción se llevaría a cabo en cada uno de los nodos. Por tanto, el usuario debe ser consciente de este comportamiento y controlar la correcta instalación de la impresora en cada nodo. Este problema se evitaría si la impresión se realizase una vez finalizada la conferencia. Además al contar con una arquitectura replicada, cada usuario, si lo desea, podría imprimir de forma independiente el documento resultante de la cooperación.

Ninguno de los sistemas replicados estudiados incluye una solución general para la gestión de los recursos externos. La solución teórica consiste en que los eventos de acceso a los recursos externos se reproduzcan únicamente en el usuario activo y no se transmitan al resto de nodos. Además en el caso en que como resultado de la operación se lean datos de entrada, estos datos leídos se deben enviar al resto de nodos [Abe93].

Abe afirma con acierto que cualquier solución debe ser específica teniendo en cuenta la aplicación que se utiliza y los dispositivos a los que se accede [Abe93]. Por tanto, para un control total de los dispositivos externos resultaría necesario introducir también *Mecanismos Conscientes de la Aplicación*. Por ejemplo, *MMConf* cuenta con

una utilidad propia de correo cooperativo que evita la redundancia en el envío de mensajes [Lauwers90].

d) Interacción espontánea

Un aspecto que tradicionalmente se ha considerado un punto débil de los sistemas replicados es su dificultad para introducir nuevos usuarios en una sesión ya iniciada [Crowley90]. Los sistemas centralizados solventan esta circunstancia de forma sencilla mediante la transferencia del estado de las ventanas en el momento que se incorpora el nuevo usuario. En *DocuLAN*, como ya se ha descrito, se ha introducido un mecanismo que soluciona este problema. Para ello es necesario finalizar la sesión y volver a iniciarla inmediatamente con el nuevo usuario y con el último estado de los documentos sobre los que se cooperaba. Este procedimiento es llevado a cabo por el Moderador y se realizará de forma transparente al resto de Participantes. Es una solución un tanto rígida ya que obliga a detener durante unos instantes la sesión, pero resulta viable en todo momento.

Otras soluciones, como las descritas por Lauwers, presentan aun más inconvenientes. La introducción de un nuevo usuario mediante la reproducción de todos los eventos generados en la sesión implica un gran consumo de recursos y de tiempo para el almacenamiento de dichos eventos y su posterior reproducción [Lauwers90b]. Esta solución no es factible en algunas configuraciones [Hyland92].

Asimismo, *DocuLAN* permite la inclusión de nuevas aplicaciones y documentos a una sesión ya iniciada. En general, *DocuLAN* presenta unas mayores facilidades para facilitar la reconfiguración dinámica de la sesión que el resto de sistemas replicados.

e) Alteración del orden de reproducción de los eventos

DocuLAN no incluye ningún mecanismo propio que gestione posibles errores en la transmisión de los eventos. En este trabajo no se han abordado en profundidad los

aspectos de la comunicación. La seguridad aportada reside en la utilización del protocolo denominado *circuito virtual* sobre TCP/IP que garantiza que los paquetes de datos llegan ordenadamente a su destino.

El sistema *Dialogo* introduce un control adicional ya que además de transmitir los eventos de entrada, transmite también los mensajes que se generan en cada nodo. Estos últimos no se reproducen en el resto de usuarios. Únicamente se toman como guía para reproducir dichos mensajes en el orden adecuado [Lauwers90].

El principal problema encontrado en *DocuLAN* en la reproducción de los eventos se debe a determinadas acciones de pulsación del ratón. Estas acciones provocan no sólo la alteración del orden de reproducción, sino su paralización completa quedando bloqueado el sistema. Este es un problema específico del entorno *MS-Windows 3.x* debido a que no proporciona una multitarea real. Aparece especialmente en las aplicaciones gráficas cuando se realizan figuras que implican pulsar el botón del ratón y arrastrarlo hasta completar la figura. Durante el intervalo de tiempo que dura esta operación, la aplicación posee el control total del entorno y no permite a otras aplicaciones procesar mensajes que tengan en su cola. En *DocuLAN*, este hecho implica que tras reproducirse el primer evento de esta operación, el sistema se bloquea porque la aplicación gráfica impide que se recuperen más eventos de la cola donde los almacena *DocuLAN*. Este problema ha sido solucionado satisfactoriamente creando una cola intermedia en la que se almacenan todos los eventos de estas operaciones de pulsar y arrastrar y únicamente son reproducidos cuando la secuencia completa de eventos está ya almacenada en cada nodo. Esta solución se ha implementado en el *Mecanismo de Pulsar y Arrastrar* y en el *Mecanismo de Pulsación Doble*.

f) Resumen

En *DocuLAN* se han utilizado los denominados *Mecanismos Generales Conscientes de la Cooperación* [Lauwers90] [Crowley90] para solventar de forma satisfactoria el problema de la consistencia. En concreto se han abordado estas acciones:

- Garantizar la consistencia en el estado inicial de las aplicaciones mediante la transferencia de los ficheros de configuración y la gestión de las resoluciones de las pantallas de los usuarios participantes.
- Resolver la situación indeterminista de la pulsación continua de la barra de desplazamiento.
- Permitir incorporar nuevos usuarios a una sesión ya iniciada.
- Permitir incorporar nuevas aplicaciones y documentos a una sesión ya iniciada.
- Resolver el problema de reproducción de los eventos de pulsación y arrastre del ratón.
- Desactivar determinadas acciones que provocan la pérdida del sincronismo, como minimizar una ventana o visualizar la lista de tareas.

Algunos de estos mecanismos pueden ser mejorados o completados, pero se ha intentado, al menos, abordar los diferentes tipos de problemas generales e introducir una solución inicial para los casos más habituales.

También resulta necesario en *DocuLAN*, y en general en el resto de sistemas replicados, que el usuario posea una mínima formación del sistema y que sea consciente del tipo de arquitectura que está ejecutando. Así, por ejemplo, para el control de los recursos externos se necesita la colaboración del propio usuario. Por tanto, además de introducir *Mecanismos Generales Conscientes de la Cooperación*, se requiere también una mayor *consciencia del usuario*.

7.3 Comparación con los sistemas de arquitectura centralizada

En el Capítulo 6 se realizó una comparación entre el comportamiento del sistema replicado *DocuLAN* y las aplicaciones *FarSite* y *ProShare* que siguen un

modelo centralizado. En esta comparación se demostró que *DocuLAN* ofrecía un menor tráfico de red y un menor tiempo de respuesta que los sistemas centralizados. Los estudios comparativos realizados hasta la fecha se basaban en sistemas desarrollados para el entorno *UNIX* [Greenberg90], el más detallado corresponde al sistema *Rapport* como ya se ha comentado [Ahuja90].

Al margen de esta evaluación cuantitativa, *DocuLAN* coincide con el resto de sistemas replicados al ser comparados con la arquitectura centralizada, a pesar de haber sido desarrollado en un entorno diferente, *MS-Windows 3.x*.

La principal desventaja de los sistemas replicados, y por tanto de *DocuLAN*, respecto a los que siguen una arquitectura centralizada es la dificultad para mantener la consistencia [Ahuja90] [Patterson90] [Hyland92]. En el apartado anterior se ha evaluado el comportamiento de *DocuLAN* ante este problema y se ha comprobado que se puede alcanzar un grado aceptable de estabilidad mediante la introducción de *Mecanismos Generales Conscientes de la Cooperación*.

No obstante, algunas de las desventajas de los sistemas replicados son inherentes al propio modelo de arquitectura y simplemente deben ser asumidas en cualquier desarrollo. Estas desventajas son:

1. Mismo entorno de ejecución. Es necesario que todos los nodos cuenten con el mismo entorno de ejecución. Esta limitación se ve agravada por la tendencia al uso de sistemas heterogéneos. Un ejemplo es Internet que permite el acceso y el intercambio de información desde plataformas diferentes.
2. Licencias en cada nodo. Las aplicaciones que se van a utilizar deben estar correctamente instaladas en cada nodo.
3. Computación. Si las aplicaciones que se utilizan implican un alto coste de procesamiento, este coste se reproducirá en cada uno de los nodos de la sesión.

Algunas de estas condiciones necesarias de los sistemas replicados, como el mismo entorno de ejecución y las licencias en cada nodo, ya se cumplen en muchas

organizaciones o empresas. Por tanto, si existen delegaciones situadas en diferentes puntos, se convierten en escenarios potenciales para utilizar sistemas replicados como *DocuLAN*. Esta situación cuenta además con la ventaja adicional de que los usuarios mantienen el vínculo de pertenecer a la misma organización, lo que favorece la cooperación.

8. Conclusiones

Los sistemas de conferencia de documentos constituyen un grupo importante dentro del software para el Trabajo Cooperativo Soportado por Ordenador. Dentro de estos sistemas de conferencia de documentos se distinguen dos tipos de arquitecturas:

- Arquitectura Centralizada, en la que las aplicaciones se ejecutan en un solo nodo de los participantes en la sesión cooperativa.
- Arquitectura Replicada, en la que las aplicaciones se ejecutan en todos los nodos que participan en la sesión cooperativa.

En esta tesis se ha desarrollado un sistema de conferencia de documentos dotado de una arquitectura replicada sobre un entorno de bajo coste. El sistema se ha denominado *DocuLAN* y el entorno elegido ha sido *MS-Windows*. *DocuLAN* se ha comparado con los sistemas de conferencia de documentos más representativos de arquitectura centralizada y de arquitectura replicada (de estos últimos únicamente existen sistemas que se ejecutan en *UNIX*). De dicha comparación se han extraído las siguientes conclusiones:

- El sistema *DocuLAN* ofrece una reducción media del tráfico de red del 80% respecto a los sistemas centralizados. Esta reducción implica un ratio de 1 a 5, es decir, el volumen de información que circula en *DocuLAN* se multiplica por 5 al utilizar otras aplicaciones con modelo centralizado.

- En la manipulación de gráficos, se alcanza en *DocuLAN* una reducción del tráfico de red del 99% respecto a los sistemas centralizados, que implica un ratio de 1 a 100 entre *DocuLAN* y dichos sistemas centralizados.
- La mejora en el tiempo de respuesta de *DocuLAN* frente a los sistemas centralizados varía desde 1 segundo a 45 segundos en las pruebas realizadas.
- Las mejoras más importantes en el tiempo de respuesta de *DocuLAN* frente a los sistemas centralizados aparecen cuando el usuario activo (aquel que trabaja con las aplicaciones cooperativas) es un usuario Participante (aquel que se suma a una sesión iniciada por el usuario Moderador). En las circunstancias anteriores, en los sistemas centralizados se alcanzan tiempos de respuesta de 25 y 45 segundos, que resultan inaceptables para el usuario. Sin embargo, en *DocuLAN*, en los mismos casos, se obtienen tiempos de 3 y 1 segundos.
- Las mejoras más significativas de *DocuLAN* respecto a los sistemas centralizados se producen cuando el volumen de información de salida de las aplicaciones es muy superior al volumen de información de entrada generada por los eventos del teclado y del ratón. Esta circunstancia depende directamente de las operaciones que se realicen y de las características de las aplicaciones, pero existen más probabilidades de que estas condiciones aparezcan en las aplicaciones que manipulan gráficos.
- Por tanto, se puede afirmar que el sistema *DocuLAN* ofrece un menor tráfico de red y un menor tiempo de respuesta que los sistemas que siguen un modelo centralizado (*ProShare* y *FarSite*).
- Los sistemas centralizados ofrecen una mayor robustez que los sistemas replicados debido a la dificultad de estos últimos para mantener la consistencia durante la cooperación.
- El problema de la consistencia no se soluciona de forma completa en *DocuLAN*, pero se alcanza un alto grado de robustez mediante la introducción de *Mecanismos Generales Conscientes de la Cooperación*. Estos mecanismos modifican el comportamiento habitual de las aplicaciones, pero resultan

necesarios para garantizar el mantenimiento de la consistencia y el sincronismo durante la cooperación.

- El sistema *DocuLAN* ofrece un comportamiento similar a los sistemas replicados existentes que se ejecutan en un entorno *UNIX*. La evaluación de estos últimos muestra unos resultados similares a *DocuLAN* respecto al tráfico de red y al tiempo de respuesta.
- *DocuLAN* mantiene la importante ventaja de haber sido desarrollado para un entorno de bajo coste y que cuenta con un gran volumen de usuarios y de aplicaciones. En cambio, los sistemas sobre *UNIX* únicamente permiten trabajar de forma cooperativa a un volumen de usuarios y de aplicaciones considerablemente más reducido.
- Los resultados de este trabajo muestran la viabilidad de los sistemas de conferencia de documentos que siguen un modelo replicado. Las importantes ventajas que presentan, como su reducido tráfico de red y el menor tiempo de respuesta, son de la suficiente magnitud como para defender esta viabilidad. Asimismo se debe continuar analizando el problema de la consistencia con el objeto de alcanzar una mayor robustez.

8.1 Trabajo futuro

Algunos problemas como el indeterminismo y el acceso a los recursos externos se podrían mejorar con la introducción de *Mecanismos Específicos Conscientes de la Aplicación*. Estos mecanismos se deben implementar para cada una de las aplicaciones que pueden participar en una conferencia. Esta filosofía ya se ha utilizado en parte en *DocuLAN* con las *Plantillas*. A través de este elemento, que es propio de cada aplicación, se podrían incorporar los módulos que contuvieran los *Mecanismos Específicos*. Se deberá realizar un estudio detallado del comportamiento de las aplicaciones seleccionadas para detectar aquellas circunstancias que provocan la inconsistencia y, de esta forma, poder implementar los mecanismos correctores.

Fundamentalmente se deberán abordar los problemas del indeterminismo y el acceso a los recursos externos. En todo momento, se mantendría un módulo general que trataría las aplicaciones para las cuales no se ha introducido ninguna especificidad. Se mantendría, por tanto, un propósito general del sistema, pero además se incluiría una especificidad para diferentes aplicaciones que aumentaría su nivel de consistencia.

Otra cuestión que ya ha sido iniciada es la adaptación de *DocuLAN* a los entornos de ventanas de 32 bits de *Microsoft, Windows 95* [King94] y *Windows NT* [Custer93]. En una primera versión para *Windows 95* [Herrero96] se hace uso de las ventajas que proporciona este entorno. Se incluye una multitarea real lo que permite lanzar diferentes procesos paralelos para transferir los archivos cooperativos a los Participantes de forma simultánea. Asimismo, todas las opciones de configuración de las aplicaciones se almacenan en una base de datos denominada *Registry* lo que facilita su tratamiento. Sobre el entorno *Windows 95*, los resultados de *DocuLAN* se deberán comparar con el sistema de arquitectura centralizada *NetMeeting* de *Microsoft* [Microsoft98], que se está convirtiendo en el producto más utilizado en este sector [Byte97b].

El siguiente paso sería evaluar el funcionamiento de la versión de *Windows 95* sobre el entorno *Windows NT* para realizar las adaptaciones que fuesen necesarias. Al ser ambos entornos de 32 bits se podría mantener una misma versión válida para ambos sistemas operativos. La situación actual y las previsiones futuras indican que estos entornos de *Microsoft* y sus futuras actualizaciones van a dominar el mercado mundial de sistemas operativos. En un informe de la consultora *Dataquest*, se prevé que para el año 2000 la cuota de mercado de *Windows 9x* sea del 54%, la de *Windows NT* el 38% y *UNIX* ocupará únicamente el 0,3% [Computing97]. Incluso, tomando como referencia únicamente el sector de las estaciones de trabajo, y según un informe de *IDC*, las ventas de equipos basados en *Windows NT* alcanzaron en 1997 1,3 millones de unidades (crecimiento del 80% respecto a 1996), mientras que en el caso de los sistemas *UNIX* esta cifra fue de 660.000 unidades (descenso del 7%) [Peworld98].

El sistema *DocuLAN*, en ningún caso requiere una máquina que realice las funciones de servidor, por tanto la versión de estación de trabajo de *Windows NT* sería suficiente. La elección del sistema operativo, y en particular la elección entre *Windows NT* y *UNIX*, es un debate permanente [Byte97] [Ganousis97] [Computing97] [Computing98]. Pero en este caso, habrá que tener en cuenta fundamentalmente que es necesario proporcionar sistemas que permitan establecer una cooperación haciendo uso de los entornos y aplicaciones que utilizan de forma mayoritaria los usuarios y, en este sentido, los entornos de *Microsoft* ocupan una posición destacada sobre los sistemas *UNIX*.

Referencias

- [3Com95] 3Com. *EtherLink III Parallel Tasking PCI Bus Master Network Adapters User Guide*. 3Com Corporation 1995.
- [Abe93] Abe, T., Maeno, K., Sakata S., Fukuoka, H. "Distributed Cooperative Control for Sharing Applications Based on the MERMAID". *NEC Res. & Develop.* January 1993. Vol. 34, no. 1, pp. 122-131.
- [Abel90] Abel, M. "Experiences in an Exploratory Distributed Organization". *Intellectual teamwork: social and technological foundations of cooperative work*, eds. Galegher, J., Kraut, R., Egido, C. New Jersey, Lawrence Erlbaum, 1990, pp. 489-510.
- [Acampora94] Acampora, A.S. *An Introduction to Broadband Networks LANs, MANs, ATM, B-ISDN and Optical Networks for Integrated Multimedia Telecommunications*. Plenum Press 1994.
- [Agostini97] Agostini, A., De Michelis, G. "Rethinking CSCW systems: the architecture of MILANO". *Proceedings of the European Conference on Computer Supported Cooperative Work (ECSCW'97)*, eds. Hughes, J.A., Prinz, W., Rodden, T., Schmidt, K. Lancaster, UK, September 7-11 1997, Kluwer Academic Press, pp. 33-48.
- [Ahuja88] Ahuja, S.R., Ensor, J.R., Horn, D.N. "The Rapport Multimedia Conferencing system". *Proceedings of the Conference on Office Information Systems*, ed. R.B. Allen. Palo Alto, California, March 23-25 1988, ACM Press, pp.1-8.
- [Ahuja90] Ahuja, S.R., Ensor, J.R., Lucco, S.E. "A Comparison of Application Sharing Mechanism in Real-time Desktop Conferencing Systems". *Proceedings of the Conference on Office Information Systems*. Boston, April 25-27 1990, ACM Press, pp. 238-248.
- [Baecker93] Baecker, R.M. *Readings in Groupware and Computer-Supported Cooperative Work. Assistant Human-Human Collaboration*. Morgan Kaufmann Publishers 1993.
- [Bahrtdt58] Bahrtdt, H.P. *Industriebürokratie. Versuch einer Soziologie des Industrialisierten Bürobetriebs und seiner Angestellten*. Ferdinand Enke Verlag, Stuttgart 1958.
- [Bahrtdt84] Bahrtdt, H.P. *Schlüsselbegriffe der Soziologie*. Verlag C.H. Beck, München 1984.
- [Bannon88] Bannon, L. "CSCW - What does it mean?". Panel Session in *Conference on Computer Supported Cooperative Work (CSCW'88)*. Portland, Oregon, September 26-28 1988.

- [Bannon91] Bannon, L.J., Schmidt, K. "CSCW: Four Characters in Search of a Context". *Studies in Computer Supported Cooperative Work*, eds. Bowers, J.M., Benford, S.D. Elsevier, Netherlands, 1991, pp. 3-16.
- [Bannon97] Bannon, L., Bødker, S. "Constructing Common Information Spaces". *Proceedings of the European Conference on Computer Supported Cooperative Work (ECSCW'97)*, eds. Hughes, J.A., Prinz, W., Rodden, T., Schmidt, K. Lancaster, UK, September 7-11 1997, Kluwer Academic Press, pp. 81-96.
- [Beaudouin92] Beaudouin-Lafon, M. "Groupware: Towards a new computer-based communication medium". *Proceedings of Informatique'92 International Conference Interface to Real and Virtual Worlds*, Montpellier, March 23-27 1992, pp. 49-51.
- [Bentley94] Bentley, R., Rodden, T., Sawyer, P., Sommerville, I. "Architectural Support for Cooperative Multiuser Interfaces". *IEEE Computer*. May 1994. Vol. 27, no. 5, pp. 37-46.
- [Bentley96] Bentley, R., Busbach, U., Sikkel, K. "The Architecture of the BSCW Shared Workspace System". *Proceedings of ERCIM Workshop on CSCW and the Web*. Sankt Augustin, Germany, February 7-9 1996.
- [Bitfiled98] Bitfield Oy. <http://www.bitfield.fi>
- [Bullen90] Bullen, C.V., Bennett, J.L. "Learning from User Experience with Groupware". *Proceedings of the Conference on Computer Supported Cooperative Work (CSCW'90)*. Los Angeles, California, USA, October 7-10 1990, ACM Press, pp. 291-302.
- [Butterfield92] Butterfield, J., Rathman, S., Whinston, A. "Groupware: A survey of Perceptions and Practice". *SIGOIS Bulletin* 1992. Vol. 13, no. 3, pp. 6-7.
- [Byte97] "Unix or Windows NT? Both". *Byte Magazine*. September 1997.
- [Byte97b] "Editors' Choice Awards". *Byte Magazine*. December 1997.
- [Chen94] Chen, H. "Collaborative Systems: Solving the Vocabulary Problem". *Computer*, IEEE Computer Society, May 1994. Vol. 27, no. 5, pp. 58-79.
- [Chung93] Chung, G., Jeffay, K., Abdel-Wahab, H. "Accommodating Late-Comers in Shared Window Systems". *IEEE Computer*. January 1993. Vol. 26, no. 1, pp. 72-74.
- [Cinco98] Cinco Networks Inc. <http://www.cinco.com>
- [Clark90] Clark, H., Brennan, S. *Grounding in Communication*. 1990.
- [Cole92] Cole, P., Nast-Cole, J. "A Primer on Group Dynamics for Groupware Developers". *Groupware: Software for Computer-Supported Cooperative Work*, eds. Marca, D., Bock, G. IEEE Computer Society Press 1992.

- [Computing97] "Sistemas Operativos: UNIX/NT". *Computing España*. 30 Abril - 7 Mayo 1997, nº 91, pp. 16-17.
- [Computing98] "Sistemas Operativos: UNIX/NT". *Computing España*. 4 Marzo - 11 Marzo 1998, nº 125, pp. 16-17.
- [Crowe94] Crowe, M.K. (Ed.). *Cooperative Work with Multimedia*. Research Reports ESPRIT. Project 6310. MMTCA, Volume 1. Springer-Verlag 1994.
- [Crowley90] Crowley, T., Milazzo, P., Baker, E., Forsdick, H., Tomlinson, R. "MMConf: An Infrastructure for Building Shared Multimedia Applications". *Proceedings of the Conference on Computer Supported Cooperative Work (CSCW'90)*. Los Angeles, California, October 7-10 1990. ACM Press, pp. 329-342.
- [Cummings94] Cummings, J. "Defining Groupware and Other Impossible Tasks". *Business Communications Review*. February 1994. Vol. 24, no. 2 pp. 35-39.
- [Custer93] Custer, H. *Inside Windows NT*. Microsoft Press, 1993.
- [DataBeam98] DataBeam Corporation. <http://www.databeam.com>
- [Datapro94] Crooks-Bile, A. "Groupware and Workgroups: Overview". *DATAPRO, Desktop Software & Solutions* 3501, Groupware, 1994, pp. 1-7.
- [Davis95] Davis, A.W. "The Latest Introductions for Still and Video Image Conferencing". *Advanced Imaging*. October 1995, pp.72-79.
- [Edwards96] Edwards, W.K. "Policies and Roles in Collaborative Applications". *Proceedings of the Conference on Computer Supported Cooperative Work (CSCW'96)*. Boston, Massachusetts, USA, November 16-20 1996, ACM Press, pp. 11-20.
- [Ehn88] Ehn, P. "CSCW: What does it mean?". Remarks in panel session in *Conference on Computer Supported Cooperative Work (CSCW'88)*. Portland, Oregon, September 26-28, 1988.
- [Ellis89] Ellis, C.A., Gibbs, S.J. "Concurrency Control in Groupware Systems". *Proceedings of the ACM SIGMOD '89 Conference on the Management of Data*. Seattle Wash., May 2-4 1989, ACM Press, New York, pp. 399-407.
- [Ellis91] Ellis, C.A., Gibbs, S.J., Rein, G.L. "Groupware: Some Issues and Experiences". *Communications of the ACM*. January 1991. Vol.34, no. 1, pp. 38-58.
- [Farsite96] DataBeam. *FarSite Document Conferencing User's Guide*.1996.

- [Fish88] Fish, R., Kraut, R., Leland, M., Cohen, M. "Quilt: A Collaborative Tool for Cooperative Writing". *Proceedings of the Conference on Office Information Systems*, ed. R.B. Allen. Palo Alto, California, March 23-25 1988, ACM Press, pp. 30-37.
- [Fish92] Fish, R., Kraut, R., Root, R. "Evaluating Video as a Technology for Informal Communication". *CHI 92*. 1992, pp. 37-48.
- [Ftp92] FTP. *PC/TCP Network Software for DOS User's Documentation Version 2.1*. FTP Software Inc. 1992.
- [Fuchs93] Fuchs, L., Prinz, W. *Aspects of Organisational Context in CSCW*. COMIC Project Deliverable, 1993. Computing Department, Lancaster University. http://www.comp.lancs.ac.uk/computing/research/soft_eng/comic/
- [Fuchs95] Fuchs, L., Pankoke-Babatz, U., Prinz, W. "Supporting Cooperative Awareness with Local Event Mechanisms: The GroupDesk System". *Proceedings of the European Conference on Computer Supported Cooperative Work (ECSCW'95)*, eds. Marmolin, H., Sundblad, Y., Schmidt, K. Stockholm, Sweden, September 10-14 1995, Kluwer Academic Press, pp. 247-262.
- [Ganousis97] Ganousis D. "UNIX vs. Windows NT. EDA Market Share". *EDA Today Online*, Vol. 3, no. 9, September 1997.
- [Gold94] Gold, E.M. "PCs Rewrite the Rules for Videoconferencing". *Data Communications*, March 1994, pp. 95-104.
- [Greenberg90] Greenberg, S. "Sharing Views and Interactions with Single-User Applications". *Proceedings of the Conference on Office Information Systems*. Cambridge, Massachusetts, April 25-27 1990, ACM Press, pp. 227-237.
- [Greenberg91] Greenberg, S. "An annotated Bibliography of Computer-Supported Cooperative Work". *ACM SIGCHI Bulletin*, July 1991, Vol. 23, no. 3.
- [Greenberg91] Greenberg, S., Roseman, M., Webster, D. "Issues and Experiences Designing and Implementing Two Group Drawing Tools". *Proceedings of the Twenty-fifth Annual Hawaii International Conference on the System Sciences*. IEEE Computer Society Press 1002, Vol. 4, pp. 139-150.
- [Greenberg94] Greenberg, S., Marwood, D. *Real Time Groupware as a Distributed System: Concurrency Control and its Effect on the Interface*. Research Report 94/534/03, Department of Computer Science, University of Calgary, Canada 1994.
- [Greenberg96] Greenberg S., Roseman M. *Groupware Toolkits for Synchronous Work*. Research Report 96/589/09, Department of Computer Science, University of Calgary, Canada, November 1996.

- [Greenberg98] Greenberg, S. "Real Time Distributed Collaboration". *Encyclopedia of Distributed Computing*, eds. Dasgupta, P., Urban, J.E. Kluwer Academic Publishers, 1998.
- [Griffiths92] Griffiths, J.M. *ISDN explained: worldwide network and applications technology*. John Wiley & Sons 1992. 2nd ed.
- [Grudin88] Grudin, J. "Why CSCW Applications Fail: Problems in the Design and Evaluation of Organizational Interfaces". *Proceedings of the Conference on Computer Supported Cooperative Work (CSCW'88)*, Portland, Oregon, September 26-28 1988, ACM Press, pp. 85-93.
- [Grudin91] Grudin, J. "CSCW Introduction". *Communications of the ACM*. December 1991. Vol.34, no. 12, pp. 30-34,.
- [Grudin94] Grudin, J. "CSCW: History and Focus". *IEEE Computer*. May 1994. Vol. 27, no. 5, pp. 19-26.
- [Grudin94b] Grudin, J. *Groupware and Social Dynamics: Eight Challenges for Developers*. Information and Computer Science Department. University of California, Irvine, USA, 1994.
- [Grudin95] Grudin, J., Palen, L. "Why Groupware Succeeds: Discretion or Mandate?" *Proceedings of the European Conference on Computer Supported Cooperative Work (ECSCW'95)*, eds. Marmolin, H., Sundblad, Y., Schmidt, K. Stockholm, Sweden, September 10-14 1995, Kluwer Academic Press, pp. 263-278.
- [Gust88] Gust, P. "SharedX: X in a distributed group work environment". *Presentation at the 2nd Annual X Conference*. MIT, January 1988.
- [Gutwin98] Gutwin, C., Greenberg, S. "Effects of Awareness Support on Groupware Usability". *Proceedings of the CHI'98 Conference on Human Factors in Computing Systems*, ACM Press, 1998.
- [ISDN90] COMMON-ISDN-API. *Standard Interface between Applications Programs and ISDN adapters. Specification. Version 1.1, Profile A*. September 1990. ISDN-PC working group of the German Deutsche Bundespost TELEKOM and the companies AVM, Stollmann and Systec.
- [Hall93] Hall, M. *Windows Sockets: An Open Interface for Network Programming Under Microsoft Windows, Version 1.1, Revision A*. Microsoft Press 1993.
- [Halsall92] Halsall, F. *Data Communications, Computer Networks and Open Systems*. Addison-Wesley Publishing Company, 1992.
- [Herrero95] Herrero, J. *Diseño de Arquitectura VIDOC*. Documento interno del proyecto VIDOC. ROBOTIKER, 1995.

- [Herrero96] Herrero, J. *Diseño de Arquitectura de la Aplicación DocuLAN5*. Documento interno del proyecto VIDOC. SER/14296/ADD/SER/1.0. ROBOTIKER, 1996.
- [Hyland92] Hyland, P. *A Taxonomy and Evaluation of Architectures for Real Time Groupware Systems*. Hewlett-Packard Lab., Bristol, UK, 1992.
- [Intel98] Intel Corporation. <http://www.intel.com>
- [ITU96] ITU-T. *T.120. Series T: Terminal Equipments and Protocols for Telematic Services, Data Protocols for Multimedia Conferencing*. International Telecommunication Union 1996.
- [Jacobs96] Jacobs, S., Gebhardt, M., Kethers, S., Rzasa, W. "Filling HTML Forms Simultaneously: CoWeb - Architecture and Functionality". *Proceedings of the Fifth International World Wide Web Conference*, May 6-10, 1996, Paris, France.
- [Johansen88] Johansen, R. *Groupware: Computer Support for Business Teams*. The Free Press, N.Y. 1988.
- [Johansen91] Johansen, R. *Leading Business Teams*. Addison-Wesley, Reading, Mass 1991.
- [Johnson-Lenz82] Johnson-Lenz, P., Johnson-Lenz, T. "Groupware: The process and impacts of design choices". *Computer-Mediated Communication Systems: Status and Evaluation*. E.B. Kerr, S.R. Hiltz. Academic Press, N. Y. 1982.
- [Kantor75] Kantor, D. *Inside the Family: A Systems Approach*. Jossey-Bass, San Francisco, California, 1975.
- [Kernighan84] Kernighan, B.W., Pike, R. *The UNIX Programming Environment*. Prentice Hall 1984.
- [King94] King, A. *Inside Windows 95*. Microsoft Press. ISBN 1-55615-626-X, 1994.
- [Kling91] Kling, R. "Cooperation, Coordination and Control in Computer-Supported Work". *Communications of the ACM*. December 1991. Vol.34, no. 12, pp. 83-88.
- [Kremer91] Kremer, H. A. O. "Computer Supported Cooperative Work - State of the Art". *Human Aspects in Computing: Design and Use of Interactive Systems and Information Management*, ed. H.-J. Bullinger. Elsevier Science Publishers B.V., 1991, pp. 1.113-1.117.
- [Kruglinski94] Kruglinski, D.J. *Inside Visual C++ Second Edition, Version 1.5*. 1994.
- [Labriola95] Labriola, D. "Desktop Videoconferencing". *PC Magazine*. April 25, 1995, pp. 221-254.

- [Lantz86] Lantz, K.A. "An experiment in integrated multimedia conferencing". *Proceedings of the Conference on Computer Supported Cooperative Work (CSCW'86)*. Austin, Texas, December 1986.
- [Lauwers90] Lauwers, J.C., Joseph, T.A., Lantz, K.A., Romanow, A.L. "Replicated Architecture for Shared Window System: A Critique". *Proceedings of the Conference on Office Information Systems*. Boston, April 25-27 1990, ACM Press, pp.249-260.
- [Lauwers90b] Lauwers, J.C., Lantz, K.A. "Collaboration Awareness in Support of Collaboration Transparency: Requirements for the Next Generation of Shared Window Systems". *Proceedings of the Conference on Human Factors in Computing*. Seattle, Washington, April 1-5 1990, ACM Press, pp. 303-311.
- [Liang94] Liang, T.P., Lai, H., Chen, N.S., Wei, H., Chen, M.C. "When Client/Server Isn't Enough: Coordinating Multiple Distributed Tasks". *IEEE Computer*. May 1994. Vol. 27, no. 5, pp. 73-79.
- [Lotus98] Lotus. <http://www.lotus.com>.
- [Lubich91] Lubich, H.P. "On Resource Reservation Strategies in Synchronous CSCW Applications". *Proceedings of the Support Functionality in the Office Environment Conference*. Canterbury, UK, September 9-12 1991. North-Holland, pp. 103-116.
- [Malone90] Malone, T., Crowston, K. "What is Coordination Theory and How Can It Help Design Cooperative Work Systems?". *Proceedings of the Conference on Computer Supported Cooperative Work (CSCW'90)*. Los Angeles, California, October 7-10 1990, ACM Press, pp. 357-370.
- [Marca92] Marca, D., Bock, G. *Groupware: Software for Computer-Supported Cooperative Work*. IEEE Computer Society Press, 1992.
- [Marsh92] Marsh, K. *Microsoft Windows Hooks*. Microsoft Developer Network Technology Group. July 1996.
- [Marshak94] Marshak, R.T. "Examining Groupware What Is It? Why Use It? What's Going On?". *Workgroup Computing Report*, 1994, vol. 17, no. 2, pp. 3-18.
- [Martin93] Martin, A. *Collaborative Technologies to Support Organizational Change*. Business Intelligence Program. SRI International, 1993.
- [Marx1867] Marx, K. *Das Kapital. Kritik der politischen Ökonomie*. Vol.1, MEGA, vol. II/5, 1867.
- [McGrath84] McGrath, J.E. *Groups: Interaction and Performance*. Prentice Hall, Englewood Cliffs, New Jersey. pp. 60-66, 1984.
- [Michels95] Michels, S. *Computer Supported Cooperative Work*. Graduation thesis. Tilburg University, The Netherlands 1995.

- [Microsoft93] Microsoft. *Microsoft Windows for Workgroups Version 3.1: Architecture Highlights*. July, 1993.
- [Microsoft94] Microsoft. *TCP/IP-32 for Microsoft Windows for Workgroups*. Microsoft Press 1994.
- [Microsoft98] Microsoft. <http://www.microsoft.com>.
- [Miller64] Miller, D.C., Form, W.H. *Industrial Sociology. The Sociology of Work Organizations*. 2nd ed., Harper & Row, New York 1964.
- [Minenko95] Minenko, W. "The Application Sharing Technology". *The X Advisor*, vol. 1, no. 1, June 1995.
- [MMaster96] Bitfield. *Bitfield MediaMaster Guides. MediaMaster Software Installation Guide, BVCS 3.0 Reference manual*. 1996.
- [Nakajima93] Nakajima, A. "Telepointing issues in desktop conferencing systems". *Computer Communications*. Septiembre 1993. Vol. 16, no. 9, pp. 603-610.
- [Ng93] Ng, J.M., Chan, E., Ip, H.S., Kwok, K.Y., Lee, Y.K., Tsang, H.H. "A Multimedia Conferencing System for Co-operative Medical Diagnosis". *Proceedings of Sixth Annual IEEE Symposium on Computer-Based Medical System*. Ann Arbor, MI, June 13-16 1993. IEEE Computer Society Press, pp. 94-99.
- [Norman91] Norman, D.A. "Collaborative Computing: Collaboration First, Computing Second". *Communications of the ACM*. December 1991. Vol.34, no. 12, pp. 88-90.
- [Norris90] Norris, P. "Workgroup Computing". *Personal Computing*, Gartner Group, 1990.
- [Norris93] Norris, R., Pendergast, M. "Critical Success Factors for Groupware Products of the 1990s". *IFIP Transactions*, 1993, Vol. A-31, pp. 135-149.
- [Nunamaker91] Nunamaker, J.F., Dennis, A.R., Valacich, J.S., Vogel, D.R., George, J.F. "Electronic Meeting Systems to Group Work". *Communications of the ACM*, vol.34, no. 7, pp. 40-61, July 1991.
- [Orlikowski95] Orlikowski, W.J. *Evolving with Notes: Organizational Change around Groupware Technology*. Working Paper #186-95, Center for Coordination Science, MIT Sloan Scholl, Cambridge, MA, 1995.
- [Palmer94] Palmer, J.D., Fields, H.A. "Computer-Supported Cooperative Work". *IEEE Computer*. May 1994. Vol. 27, no. 5, pp. 15-17.
- [Palmer94b] Palmer, J.D., Fields, H.A., Brouse, P.L. "Multigroup Decision-Support Systems in CSCW". *IEEE Computer*. May 1994. Vol. 27, no. 5, pp. 67-72.

- [Patterson90] Patterson, J.F., Hill, R.D., Rohall, S.L., Meeks, W.S. "RENDEZVOUS: An Architecture for Synchronous Multi-User Applications". *Proceedings of the Conference on Computer Supported Cooperative Work (CSCW'90)*. Los Angeles, California, October 7-10 1990, ACM Press, pp. 317-328.
- [Patterson96] Patterson, J.F., Day, M., Kucan, J. "Notification Servers for Synchronous Groupware". *Proceedings of the Conference on Computer Supported Cooperative Work (CSCW'96)*. Boston, Massachusetts, USA, November 16-20 1996, ACM Press, pp. 122-129.
- [Peworld96] "Windows NT eclipsa a Windows 95 en las empresas". *PC WORLD*. Octubre 1996, no. 125, pp. 24.
- [Peworld98] "Las estaciones de trabajo con Windows NT superan a los sistemas UNIX, según IDC". *PC WORLD*. Abril 1998, no. 142.
- [Petzold90] Petzold, C. *Programming Windows*. Microsoft Press 1990.
- [Piccardi89] Piccardi, S., Tisato, F. *Conference Desk: An experiment and model for application sharing*. Technical report, Systems Software Laboratory, Direzione Olivetti Ricerca Milan, January 1989.
- [Poltrock95] Poltrock, S., Grudin, J. *Groupware and Workflow: A Survey of Systems and Behavioral Issues*. Tutorial Notes in ECSCW'95, 1995.
- [Popitz57] Popitz, Bahrtdt, H.P., Jüres, E.A., Kesting, H. *Technik und Industriearbeit. Soziologische Untersuchungen in der Hüttenindustrie*. J.C.B. Mohr, Tübingen, 1957.
- [Posner92] Posner, I.R., Baecker, R.M. "How People Write Together". *Proceedings of the Twenty-Fifth Hawaii International Conference on System Sciences*. January 7-10 1992. Vol. 4, pp. 127-138.
- [Proshare94] Intel. *Intel ProShare Software, Getting Started Guide*. 1994.
- [Racal90] Racal. *NI6510 Installation Manual*. Racal InterLan 1990.
- [Reinhard94] Reinhard, W., Schweitzer, J., Völksen, G., Weber, M. "CSCW Tools: Concepts and Architecture". *IEEE Computer*. May 1994. Vol. 27, no. 5, pp. 28-36.
- [Richter92] Richter, J.M. *Windows 3.1: A Developer's Guide*. Prentice Hall. M&T Publishing Inc. 1992.
- [Roberts93] Roberts, M. Groupware. "Innovative New Technology or Mutton Dressed as Lamb". *Information Management & Technology* 1993. Vol. 26, no. 5, pp. 212-214.
- [Rodden91] Rodden, T. *A Survey of CSCW Systems*. Department of Computer Science, Lancaster University, UK, 1991.

- [Rodden91b] Rodden, T. *Technological Support for Cooperation*. Department of Computer Science, Lancaster University, UK, 1991.
- [Rodden96] Rodden, T. "Populating the Application: A Model of Awareness for Cooperative Applications". *Proceedings of the Conference on Computer Supported Cooperative Work (CSCW'96)*. Boston, Massachusetts, USA, November 16-20 1996, ACM Press, pp. 87-96.
- [Rogers83] Rogers, E. *The Diffusion of Innovation*. Free Press, New York, NY, 1983.
- [Romano97] Romano, P. *Application Sharing, ITU-T Recommendation T.128*. 1997.
- [Roseman92] Roseman, M., Greenberg, S. "GROUPKIT A Groupware Toolkit for Building Real-Time Conferencing Applications". *ACM Conference on CSCW*. Toronto, Nov. 1-4 1992, pp. 43-50.
- [Sarin84] Sarin, S.K. *Interactive On-line Conferences*. PhD thesis, MIT. Published as Laboratory for Computer Science Technical Report TR-330, 1984.
- [Sarin88] Sarin, S.K., Greif, I. "Computer-based real-time conferencing systems". *Computer*. October 1985. Vol. 18, no. 10, pp. 33-45. Reprinted in *Computer-Supported Cooperative Work: A Book of Readings*, ed. I. Greif. Morgan Kaufmann Publishers, 1988, pp. 397-420.
- [Sawyer94] Sawyer, T.Y. *Groupware and Business Process Reengineering - The Business Case*. 7900 W. Layton Ave. #903, Littleton, Colorado 80123 tsawyer@usa.net, 1994.
- [Scheifler90] Scheifler, R.W., Gettys, J. "The X Window System". *ACM Transactions on Graphics*. April 1986. Vol. 5, No. 2, pp. 79-108.
- [Shim97] Shim, H.S., Hall, R.W., Prakash, A., Jahanian, F. "Providing Flexible Services for Managing Shared State in Collaborative Systems". *Proceedings of the European Conference on Computer Supported Cooperative Work (ECSCW'97)*, eds. Hughes, J.A., Prinz, W., Rodden, T., Schmidt, K. Lancaster, UK, September 7-11 1997, Kluwer Academic Press, pp. 237-252.
- [Short76] Short, J., Williams, E., Christie, B. *Visual Communication and Social Interaction*, 1976.
- [SMC93] SMC. *EtherCard PLUS Elite16 Series*. SMC 1993.
- [Sorgaard87] Sorgaard, P. *A Cooperative Work Perspective on Use and Development of Computer Artifacts*. Computer Science Department, Aarhus University, Denmark, 1987.
- [Sorgaard88] Sorgaard, P. *A Framework for Computer Supported Cooperative Work*. Computer Science Department, Aarhus University, Denmark 1988.

- [Stallings87] Stallings, W. *Handbook of Computer-Communications Standards. Local Network Standards. Volume 2*. Howard W. Sams & Company 1987.
- [Stefik87] Stefik, M., Bobrow, D.G., Foster, G., Lanning, S., Tartar, D. "WYSIWIS revoiced: Early experiences with multiuser interfaces". *ACM Transactions on Office Information Systems*. April 1987. Vol 5, no. 2, pp. 147-167.
- [Stevens90] Stevens, W.R. *UNIX Network Programming* Prentice Hall 1990.
- [Stevens94] Stevens, W.R. *TCP/IP Illustrated, Volume 1 The Protocols*. Addison-Wesley 1994.
- [Stevens94b] Stevens, W.R. *TCP/IP Illustrated, Volume 2 The Implementation*. Addison-Wesley 1994.
- [Stodolsky95] Stodolsky, D. *Frequently Asked question (FAQ) list for comp.groupware._Comp.groupware_ [Usenet]* 1995. (Available by anonymous FTP from rfm.mit.edu in the directory pub/usenet/news.answers/comp.groupware.)
- [Tattam94] Tattam, P.R. *Trumpet Winsock Version 2.0*. Trumpet Software International Pty Ltd. 1994.
- [Taylor95] Taylor, K. "Desktop Videoconferencing". *Data Communications*. April 1995, pp. 64-80.
- [Teege93] Teege, G., Borghoff, U.W. *Combining Asynchronous and Synchronous Collaborative Systems*. Institut für Informatik, Technische Universität München, Germany 1993.
- [Thimbleby93] Thimbleby, H., Pullinger, D. "Observations on practically perfect CSCW". *IEE Colloquium on CSCW Issues for Mobile and Remote Workers*. London, March 16 1993, Springer-Verlag, pp. 5/1-4.
- [Thompson67] Thompson, J.D. *Organizations in action. Social science bases of administrative theory*. Mc Graw-Hill, New York 1967.
- [Tian95] Tian, S. "Nondeterminate Behaviour: One of the Challenges to Integrating Single User Applications into Groupware Systems". *Computing & Information Systems Department Journal*. University of Paisley, Scotland 1995, pp. 25-31.
- [Tou94] Tou, I., Berson, S., Estrin, G., Eterovic, Y., Wu, E. "Prototyping Synchronous Group Applications". *IEEE Computer*, May 1994, Vol. 27, no. 5, pp. 48-56.
- [Turner97] Turner P., Turner, S. "Supporting Cooperative Working Using Shared Notebooks". *Proceedings of the European Conference on Computer Supported Cooperative Work (ECSCW'97)*, eds. Hughes, J.A., Prinz, W., Rodden, T., Schmidt, K. Lancaster, UK, September 7-11 1997, Kluwer Academic Press, pp. 281-295.

- [Turrell95] Turrell, M. *Learning through Experience: How Companies are using Groupware*, City University Business School, UK, 1995.
- [Ure1835] Ure, A. *The Philosophy of Manufactures*. London, 1835.
- [Wakefield1849] Wakefield, E. *A view of the art of colonization*. London, 1849.
- [Walther96] Walther M. "Supporting Development of Synchronous Collaboration Tools on the Web with GroCo". *Proceedings of ERCIM Workshop on CSCW and the Web*. Sankt Augustin, Germany, February 7-9 1996.
- [Watabe90] Watabe, K., Sakata, S., Maeno, K., Fukuoka, H., Ohmori, T. "Distributed Multiparty Desktop Conferencing System: MERMAID". *Proceedings of the Conference on Computer Supported Cooperative Work (CSCW'90)*, Los Angeles, California, October 7-10 1990, ACM Press, pp. 27-38.
- [Whitehead94] Whitehead, R., Bischoff, D. *Groupware in Action*. Course Notes, 1994.
- [Whitley74] Whitley, R. "Cognitive and social institutionalization of scientific specialities and research areas". *Social Processes of Scientific Development*, ed. Whitley, R. Routledge and Kegan Paul, London, 1974, pp. 69-95.
- [Wilson94] Wilson, D. "Is Groupware the Office panacea for the 90's?". *QA prompt*. September 1994.
- [World95] "Videoconferencia total". *Comunicaciones World*. Octubre 1995, pp. 43.
- [World96] "Colaboración en tiempo real". *Comunicaciones World*. Septiembre 1995, pp. 30.