

Tabla de Contenido

Lista de tablas	III
Lista de figuras	V
1. Introducción	1
1.1. Antecedentes	1
1.2. Definición del problema	2
1.3. Objetivos	2
1.3.1. Objetivo General	2
1.3.2. Objetivos específicos	2
1.4. Justificación	2
1.5. Alcance	3
1.6. Descripción del contenido	3
2. Sistemas Colaborativos	5
2.1. CSCW y Sistemas Colaborativos	5
2.1.1. Groupware	6
2.1.2. Workflow	7
2.2. Awareness	8
2.3. AMENITIES: Metodología para el estudio y desarrollo de sistemas cooperativos . . .	10
2.3.1. Vista de grupo	10
2.3.2. Vista de cognitiva	11
2.3.3. Vista de interacción	11
2.3.4. Vista de información	12
2.4. Refactorización	12
2.4.1. Arquitectura centralizada y replicada	13
3. Interfaces De Usuario	15
3.1. Interfaz De Usuario	15
3.1.1. Usabilidad	16
3.1.2. Naturalidad	16
3.2. Diseño de interfaces gráficas de usuario	17
3.2.1. Sketch Design	17

4. Adaptación	21
4.1. Adaptación transparente y transformación operacional	21
4.2. Adaptación Transparente	21
4.3. Modelo de adaptación	22
Bibliografía	22
Referencias	25

Lista de tablas

Lista de figuras

2.1. Groupware Time Space Matrix. Fuente:	7
2.2. Composition of workspace awareness information. Fuente: Grudin and Greenberg 2002	9
2.3. Arquitectura centralizada. Fuente: Lauwers 90	13
2.4. Arquitectura replicada. Fuente: Lauwers 90	14
3.1. Ejemplo de un prototipo de baja fidelidad.	18
3.2. Ejemplo de un prototipo de fidelidad media.	18
3.3. Ejemplo de un prototipo de fidelidad alta.	19
4.1. Estado inicial del sistema, ejemplo.	23

Capítulo 1

Introducción

El diseño de interfaces en el área del desarrollo de software es una actividad muy importante, la misma es realizada por los diferentes equipos de trabajo y comprende el trabajo conjunto de un grupo de personas, lo que implica colaboración y coordinación al momento de realizarse.

La participación de un grupo de personas puede representar una dificultad a la hora de diseñar una interfaz propiamente dicha, puesto que los miembros del equipo deben generalmente hallarse geográficamente en el mismo lugar y al mismo tiempo para realizar algún aporte inmediato, o esperar a la retroalimentación una vez terminado el diseño. La mayor parte de las herramientas que ayudan en el diseño de interfaces son del tipo mono usuario, esto significa que solo un usuario puede usarlas a la vez.

El proyecto pretende realizar un sistema colaborativo para coadyuvar en la coordinación de dicha actividad a partir de una herramienta ya existente, con ello se pretende facilitar la participación de los integrantes del equipo permitiéndoles diseñar una interfaz de manera conjunta usando algunos principios de CSCW para la adaptación de un sistema mono usuario en un sistema multiusuario.

1.1. Antecedentes

La interfaz es el medio por el cual un usuario puede interactuar con un sistema. Por ello el diseño de la interfaz tiende a ser fundamental dentro del desarrollo de un proyecto.

Al ser esta una actividad importante se involucra una cantidad considerable de personas, como el equipo desarrolló, el usuario para el cual está pensado el proyecto y algunos otros que se consideren importantes para esta actividad.

Los equipos o grupos de trabajo al involucrar cantidades determinadas de personas requieren herramientas y técnicas para hacer sus tareas de la mejor forma posible y con un alto nivel de colaboración.

Se han encontrado trabajos similares como: El Sistema de Diseño de interfaces “pencil” (Evolus, 2010)

“Sometimes you need to see through walls a field study of application programming interface”. “A veces se necesita ver a través de las paredes - un campo de estudio de aplicaciones para programar interfaces” (de Souza, Redmiles, Cheng, Millen, y Patterson, 2004)

1.2. Definición del problema

Gran parte de los equipos de desarrollo se reúnen para tratar el diseño de la interfaz a ser desarrollada, lo cual conlleva a que todos los miembros del equipo se encuentren en un lugar y en un mismo tiempo lo que puede presentar un grave problema al momento de coordinar el trabajo.

Los sistemas de apoyo al diseño de interfaces en su mayoría son del tipo mono-usuario, lo que quiere decir que pueden ser usados por un usuario en un determinado periodo de tiempo, esto representa una limitante al momento de realizar trabajo en equipo.

Al momento de realizar el diseño de la interfaz los miembros del equipo pueden brindar una amplia gama de ideas, y el llevar un registro de las propuestas o los cambios realizados se torna un poco dificultoso.

Por lo expresado anteriormente se define el problema como: Inadecuada coordinación en el equipo para el diseño de interfaces lo que ocasiona dificultad en la participación de los miembros del equipo de trabajo.

1.3. Objetivos

1.3.1. Objetivo General

Apoyar la coordinación del equipo en el diseño de interfaces mediante un Sistema Colaborativo facilitando de este modo la participación de los integrantes del equipo.

1.3.2. Objetivos específicos

1. Brindar la posibilidad de trabajo multiusuario coadyuvando al trabajo en equipo.
2. Facilitar la retroalimentación del trabajo realizado mediante un historial de acciones relevantes realizadas.
3. Brindar una alternativa de trabajar en un mismo proyecto desde diferentes ubicaciones.
4. Facilitar la retroalimentación de las propuestas brindadas por los miembros del equipo de trabajo.

1.4. Justificación

El proyecto se realizara tras advertir la necesidad de los equipos de desarrollo de software por fomentar la participación de la mayor cantidad de personas en el diseño de las interfaces de un proyecto.

Se usara principios de Middleware pues es muy útil para permitir el funcionamiento de aplicaciones distribuidas sobre plataformas heterogéneas y homogéneas.

Se pretende también demostrar los principios de adaptación de un sistema mono usuario a un sistema multiusuario de manera sencilla.

1.5. Alcance

Con el proyecto brindar una herramienta capaz de permitir el trabajo de diseño de interfaces de usuario con la participacion de un equipo de trabajo sin la necesidad de forzar a los mismos a usar una herramienta diferente a la que estan acostumbrados.

1.6. Descripción del contenido

El presente proyecto fue realizado segun el flujo de trabajo presentado en las herramientas de la metodología AMENITIES

Capítulo 2

Sistemas Colaborativos

El trabajo en grupo es una actividad humana fundamental y los entornos colaborativos facilitan este proceso. Un aspecto crucial para su estudio y diseño es el uso de técnicas eficientes que permitan representar los procesos de comunicación, coordinación y acceso a la información. Desde el nacimiento de los sistemas computacionales, el trabajo en muchos ámbitos fue transformado drásticamente, con dicho cambio también surgió la necesidad de compartir información, recursos, memoria y otros.

Con la aparición de las redes y el internet, se logró compartir información y algunos recursos. Pero aún no es suficiente, puesto que también surge la necesidad del trabajo conjunto o en equipo, un claro ejemplo de dichas necesidades es el conocido caso de Linux, un sistema operativo desarrollado por gran cantidad de personas de distintas partes del mundo.

Debido a la naturaleza de algunas actividades dentro el desarrollo de software, a veces es indispensable incorporar a distintos grupos de personas al momento de desarrollar algún trabajo específico. Dentro del diseño de interfaces se recomienda la incorporación de diversos grupos de usuarios como ser: programadores, clientes (algún usuario al que va destinado el sistema), diseñadores gráficos y algún otro tipo de usuario que podrá realizar aportes significativos para el diseño.

Lo descrito anteriormente denota un trabajo de colaboración entre distintos tipos de personas con ideas diferentes, pero con un objetivo en común. Para coadyuvar en la actividad de dicho grupo, se plantea el uso de un sistema colaborativo.

Dado que el foco del proyecto es desarrollar un sistema colaborativo, con el objetivo de identificar la utilidad del mismo dentro del proceso de creación de una interfaz de usuario, es necesario plantear algunos conceptos básicos que servirán como marco conceptual y de referencia.

2.1. CSCW y Sistemas Colaborativos

Según (Garrido, Gea, Gutierrez, y Padilla, 2000) “CSCW es una disciplina emergente que analiza el trabajo en grupo asistido por computadora, con una estrecha aplicación en el control de organizaciones y distribución del trabajo”. Sin embargo, este aspecto cada vez tiene un sentido más amplio, lo que permite recoger entornos en los cuales se produce interacción entre participantes con diversos fines (educativos, visitas turísticas, ocio, etc.), con diferentes habilidades de los usuarios implicados, distintos medios y soportes de acceso (computación ubicua). El término “Computer-supported

Cooperative Work (CSCW)” introducido por Irene Greif y Paul M. Cashman (Brown, Cashman, Malone, y Greif, 1985) se refiere a “Cómo las actividades colaborativas y su coordinación pueden ser apoyadas por medio de sistemas computacionales.”, CSCW suele ser interpretado como un sinónimo de sistema colaborativo, sin embargo algunos autores afirman que, mientras los sistemas colaborativos se refieren a sistemas computacionales reales, CSCW se enfoca en estudiar las herramientas y las técnicas de un sistema colaborativo así como su impacto psicológico, social y el efecto organizacional que causa. Wilson (Wilson, 1991) define los CSCW de la siguiente manera:

“CSCW es un término genérico, que combina la comprensión de la forma en que las personas trabajan en grupo usando tecnologías de redes de computadoras y hardware asociado, software, servicios y técnicas.”

En cambio define los Sistemas colaborativos como:

“Actividades de grupo intencionales con software para su apoyo.”

Dentro del software colaborativo se puede encontrar: los Groupware y Workflow que son las tecnologías más usadas y que serán brevemente descritas a continuación.

2.1.1. Groupware

La relación usuario máquina comenzó a cambiar a medida que el software fue evolucionando, un claro ejemplo se presentó en software orientado al entretenimiento (videojuegos), que a medida que los usuarios comenzaron a necesitar algo más que un mero pasatiempo, se comenzó a desarrollar características importantes como el soporte a multijugador en línea, el cual permite a varios usuarios encontrarse en el mismo videojuego. En muchas labores dentro del ámbito empresarial se necesita interactuar con grupos de personas, en este contexto trasladar la característica de esos videojuegos antes mencionada puede ser muy útil, el interactuar con las personas que necesitas en tiempo real sin necesidad de tenerlos en el mismo lugar geográfico, lo cual representa una gran ventaja al momento de realizar ciertas actividades.

Los groupware (Ellis, Gibbs, y Rein, 1991) son un tipo de sistemas colaborativos que se enfocan en ayudar a los equipos de trabajo a realizar actividades mediante una red. Para describir un Groupware tomaremos la siguiente definición:

“Sistemas basados en computadoras que apoyan a grupos de personas que trabajan en una tarea común y que proveen una interfaz para un ambiente compartido” -Dave Chaffney

Los Groupware hacen énfasis en tres características muy importantes: “Colaboración, Comunicación y Coordinación” (Ellis y cols., 1991). Estos tres elementos son fundamentales a la hora de trabajar en equipo. A su vez buscan promover un ambiente compartido de colaboración, en el que se perciba realmente el trabajo en equipo, un ambiente en el que se pueda interactuar con las demás personas no solo en el trabajo desempeñado, sino también mediante texto, voz o video.

Se puede notar en la definición de Groupware que esta no define que el trabajo deba que ser simultáneo. Estos sistemas pueden proveer diferentes tipos de soporte de acuerdo a las necesidades que se presenten; Si el trabajo dentro del Groupware es efectuado por los usuarios al mismo tiempo, se lo define como síncrono, de lo contrario, si el trabajo es efectuado en distintos lapsos de tiempo por los usuarios, es definido como asíncrono. Si el trabajo se realiza desde diferentes lugares geográficos se denomina distribuido; esto puede verse de mejor manera en la siguiente figura 2.1:

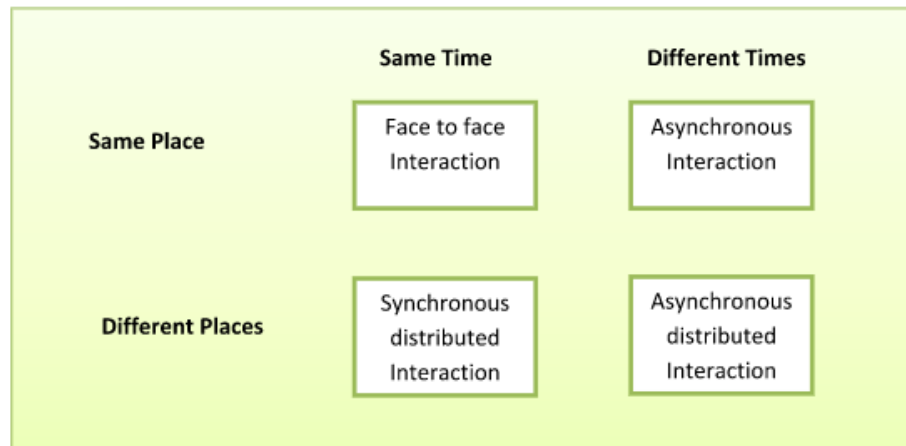


Figura 2.1: Groupware Time Space Matrix. Fuente: (Ellis y cols., 1991)

CSCW se convirtió en un campo de investigación amplio que abarca desde el análisis sociológico sobre la forma de trabajar en grupos a las tecnologías informáticas que apoyan el trabajo en grupo de las personas.

Existe una delgada línea que separa a los sistemas que son considerados groupware y los que no. El criterio más importante (Ellis y cols., 1991) para poder definir un groupware es el soporte que ofrece al trabajo en equipo, por ejemplo un sistema que brinda soporte al trabajo de múltiples usuarios, pero en áreas diferentes y realizando tareas diferentes tiene un bajo nivel de cooperación, a diferencia de un sistema pensado para dar soporte a múltiples usuarios realizando una tarea conjunta en un espacio compartido, por ejemplo el caso de google docs. El desarrollo de software es una actividad donde el trabajar de manera distribuida es algo que sucede regularmente, se tienen varios ejemplos grandes de dicha situación como ser el conocido caso del sistema operativo Linux.

Entre los criterios que se toman para catalogar un groupware como tal, están el desarrollar la misma tarea y el estar presentes en espacio compartido. Por ejemplo, un sistema de tiempo compartido permite que múltiples usuarios realicen actividades, pero estos pueden tener metas diferentes, por lo tanto realizarán tareas diferentes, en cambio un sistema que permite editar un documento de manera conjunta (google docs), está enfocado a que los usuarios que ingresen al sistema tengan en mente una misma meta, lo cual se refleja en las actividades que se realicen. De igual forma un sistema de mensajería permite que dos usuarios puedan interactuar y enfocarse en una misma tarea, pero el ambiente y la colaboración no son muy notorios, en cambio el sistema que permite el diseño de la interface tiene que dar soporte para que los usuarios puedan percibir lo que sus compañeros están haciendo.

2.1.2. Workflow

Los Workflows son sistemas que ayudan a administrar y automatizar procesos de negocios en los que participan diferentes usuarios.

La WfMC (Workflow Management Coalition) define a los Workflows Como:

“La automatización de un proceso de negocio, total o parcial, en la cual documentos, información o tareas son pasadas de un participante a otro a los efectos de su procesamiento, de acuerdo a un conjunto de reglas establecidas.”

En la misma definición radica la diferencia entre groupware y workflow, dado que los groupware son sistemas orientados al trabajo conjunto, en cambio los workflow están orientados dar soporte al conjunto de personas que realizan distintas tareas siguiendo un flujo de trabajo. Debido a los diferentes tipos de procesos que se pueden dar soporte con un workflow se definen:

a) Workflows de Producción: o también llamados Workflows de transacciones; las transacciones son la base de las acciones dentro de las bases de datos, los workflow de producción en general automatizan procesos de negocios que tienden a ser repetitivos y con gran manejo de datos.

b) Workflows de Colaboración: Los workflow que están orientados a resolver procesos de negocios donde participan grupos de personas que tienen una meta común son denominados Workflows de colaboración.

c) Workflows de Administración: Son aquellos que están orientados a dar soporte a la los procesos de administración de una empresa o negocio.

Se puede afirmar que los Workflow son herramientas muy útiles para identificar y evaluar procesos, también son muy importantes al hacer un trabajo de reingeniería puesto que su principal enfoque es el integrar a las personas y los procesos que se llevan a cabo dentro de un negocio.

Después de haber definido la diferencia entre groupware and workflow y tomando en cuenta las características que ambas opciones presentan, se optó por los Groupware como el centro teórico del presente proyecto, dadas las facilidades en el apoyo a la colaboración dentro de un trabajo grupal, que es exactamente lo que se busca al momento de diseñar interfaces de usuario.

2.2. Awareness

En un WYSIWIS (What you see is what I see- Lo que ves es lo que veo) relajado, los usuarios pueden trabajar en diferentes partes de un espacio compartido. En dicha circunstancia es importante que el usuario sea consciente del estado de los otros usuarios, para que de esa manera pueda colaborar de manera natural y fluida con ellos. En este ámbito se habla de la conciencia de grupo (awareness) también definida como momento de comprensión de la interacción de otra persona con el espacio compartido. (Gutwin, Greenberg, y Roseman, 1996)

El awareness es fundamental en los sistemas colaborativos, puesto que incrementa significativamente la usabilidad del sistema (Gutwin y cols., 1996). Es usado principalmente para, simplificar la comunicación, coordinar acciones, ayudar a los usuarios a anticipar futuras acciones y entender la ayuda de los demás. (Gutwin y cols., 1996), la información de la conciencia de grupo consiste en los diferentes elementos listados en la siguiente figura 2.2.

La primera columna de la tabla lista básicamente las categorías de la información del awareness, incluyendo: con quien estamos trabajando, que es lo que están haciendo, donde estan, como estan ocurriendo dichos eventos, cuando estos eventos están sucediendo.

Category	Element	Specific Questions
Who	Presence	Is anyone in the workspace?
	Identity	Who is participating? Who is that?
	Authorship	Who is doing that?
	Presence History	Who was here, and when?
What	Action	What are they doing?
	Intention	What goal is that action part of?
	Artifact	What object are they working on?
	Action History	How did that operation happen?
Where	Location	Where are they working?
	Gaze	Where are they looking?
	View	Where can they see?
	Reach	Where can they reach?
	Location History	Where has a person been?
How	Action History	How did that operation happen?
	Artifact History	How did this artifact come to be in this state?
When	Event History	When did that event happen?

Figura 2.2: Composition of workspace awareness information. Fuente: Grudin and Greenberg 2002

En cada categoría existen diferentes elementos de conciencia, cada elemento describe una respuesta a cada pregunta que se puede realizar dentro de un espacio compartido las cuales se muestran en la tercera columna. (Gutwin and Greenberg 2002).

Para llenar la información requerida en el awareness, existen diferentes herramientas que son usadas comúnmente por los sistemas colaborativos:

Telepointer: El telepointer es la manifestación del puntero remoto de otro usuario, que se muestra en el sistema del usuario local.

Multi-user Scrollbar: el multi-user scrollbar es una extensión del scrollbar simple, que indica la posición de los demás usuarios.

Radar view: el radar view es una herramienta para entregar una información más detallada de la ubicación de un usuario, se usa generalmente cuando el sistema colaborativo presenta más de un espacio compartido para el trabajo.

En adición a las herramientas descritas existen otras no tan frecuentes que pueden ser de gran ayuda de acuerdo a las características del sistema en cuestión, dichas herramientas pueden ser el sonido, una lista de usuarios con la información de su sesión, video, áreas de intercambio de mensajes, entre otros.

2.3. AMENITIES: Metodología para el estudio y desarrollo de sistemas cooperativos

Cuando se realiza algún proceso que implica la colaboración entre personas, lo primero que se debe tener en cuenta es la estructura y la organización de dicho grupo de trabajo. Como resultado se tienen como tareas fundamentales el recoger los aspectos organizativos, cognitivos y de interacción que representen de manera adecuada las demandas del trabajo en concreto.

Existen propuestas dentro del diseño de sistemas colaborativos que ofrecen notaciones con diferente nivel de formalismo y abstracción para recoger aspectos relativos a dichos sistemas. Dentro de la amplia gama de propuestas se encuentran GTA [welie] y ConcurTaskTrees [Paterno,97]. La primera se centra en la descomposición de tareas y asignación de actividades a roles, en cambio CTT se centra más en aspectos de coordinación (uso temporal de espacios comunes), sin embargo estas propuestas no dejan de ser parciales y no cubren las características de los sistemas colaborativos, a su vez son difíciles de enlazar con técnicas de Ingeniería de Software.

Para trabajar sobre la complejidad inherente a los entornos colaborativos, se ha propuesto el uso de una metodología conocida como “AMENITIES”, que permite describir un sistema colaborativo mediante cuatro vistas que facilitan detectar los aspectos más relevantes de los sistemas de este tipo.

AMENITIES (acrónimo de A Methodology for aNalysis and Design of CooperaTive systEmS) [Garrido, 03] integra de modo jerarquizado varios modelos de comportamiento y tareas, con la idea de proporcionar una representación del sistema colaborativos en su conjunto y con diferentes vistas complementarias.

2.3.1. Vista de grupo

Para la vista de grupo, primeramente se debe identificar los aspectos propios del grupo u organización, y las restricciones que se imponen. Dichas organizaciones se articulan bajo “roles”, que determinan las relaciones entre los miembros del grupo y las tareas que están bajo su cargo.

Las relaciones, normalmente están condicionadas por una serie de restricciones impuestas al sistema colaborativo, y se pueden identificar como más importantes las siguientes:

- Capacidades: Son restricciones cognitivas que se imponen a cada actor para participar en un rol determinado. Estas capacidades determinan los conocimientos que se debe adquirir por un usuario para participar en un rol concreto.
- Leyes: Son restricciones impuestas por la organización e identifica las reglas que se deben preservar en un grupo. Normalmente estas reglas son la misma estructura social que se manifiesta en el grupo(jerarquía, democracia, etc.)

Ambas restricciones permiten modelar sistemas dinámicos, es habitual que tanto la estructura del grupo como su funcionamiento se modifique en el tiempo (los participantes pueden adquirir nuevas capacidades, variar en número de miembros que lo conforman o bien, modificar las leyes que rigen el grupo al aplicar nuevas estrategias de trabajo).

2.3.2. Vista de cognitiva

La vista cognitiva representa el conocimiento que posee o adquiere cada miembro del grupo en el escenario colaborativo. Este conocimiento queda reflejado mediante la descripción de las tareas que puede llevar a cabo.

La descripción de las tareas implica un análisis profundo de las actividades que se deben realizar en el grupo, la división del trabajo y determinar las interrelaciones que existen entre ellas. El análisis de tareas contempla todos estos pasos, y las divide en dos fases claramente diferenciadas.

En primer lugar se define la denominada interfaz del rol, que recoge las características más relevantes de las tareas a desempeñar por un rol junto a las interrelaciones con el resto de participantes (tareas) y entorno (mediante eventos).

Los aspectos más relevantes que identificamos en el interfaz del rol son:

- Identificar tareas a desempeñar
- Relación con otras tareas tales como:
 - Si puede ser interrumpida por otra tarea
 - Su naturaleza cooperativa
 - Mecanismo de activación y modos de sincronización

Este tipo de relaciones modelan el comportamiento tanto del usuario como del propio entorno. Las mismas se modelan mediante eventos que provocan cambios en el entorno (y en las políticas que determinan el comportamiento del sistema).

En la segunda fase, se describe y pormenoriza cada tarea mediante una descomposición jerárquica, que se completa con información y aspectos recogidos de otras vistas. En esta descripción de tareas usaremos notaciones que nos permite especificar secuencialidad, concurrencia, optatividad, decisiones, etc.

Posteriormente, se detalla las tareas tanto individuales como cooperativas, y en las cuales, puede aparecer información relativa a otras vistas.

2.3.3. Vista de interacción

La vista de interacción abarca otros aspectos que se deben estudiar como ser los procesos que implican un diálogo entre participantes para analizar sus características, concretamente:

- El modo de diálogo que se producen entre participantes
- Los requisitos que impone ese diálogo sobre los medios a utilizar

Este modo de diálogo se identifica mediante protocolos. Los protocolos se pueden analizar por separado dentro de la organización ya que en gran medida son independientes del dominio del problema, y por tanto, se pueden incorporar al análisis de tareas. Por ejemplo, se pueden identificar protocolos democráticos (toma de una decisión por mayoría), consenso (aprobación unánime de una decisión), jerárquica, etc.

2.3.4. Vista de información

En la vista de información, se debe recoger la información que es compartida en el escenario. Esta información se puede describir de manera implícita en las actividades y acciones o bien, de modo explícito como flujo de información entre actividades. La información que fluye a través del sistema colaborativo serán los documentos (los objetos que son gestionados en el sistema), eventos y recursos.

2.4. Refactorización

AMENITIES está centrada en el modelado inicial del sistema usando el punto de vista del usuario y teniendo muy en cuenta aspectos relacionados con el grupo (conciencia de grupo, relaciones entre usuarios, dinámica del grupo, representación de aspectos sociales, etc), dicha definición permite la posibilidad del desarrollo de un sistema colaborativo ya sea desde su inicio, o transformándolo a partir de un sistema ya realizado, agregando las características ya mencionadas, puesto que la dinámica de grupo representa la evolución del contexto en el que se va a realizar la colaboración entre los usuarios.

Durante el desarrollo de sistemas colaborativos, existen dos opciones, la creación de un sistema colaborativo o refactorizar una aplicación mono-usuario en una aplicación multi-usuario [Qian Xia, 06], para realizar la refactorización se debe tomar en cuenta los siguientes requerimientos:

- **Compatibilidad de la Aplicación:** Las características de la interfaz, funcionalidades y formatos de la aplicación mono-usuario original deben mantenerse.
- **Transparencia de la aplicación:** no deben realizarse cambios en el código fuente original.
- **Respuesta local rápida:** la interacción con el sistema en un medio local debe ser tan rápida como la respuesta del sistema original.
- **Colaboración sin restricción:** Se debe permitir a los usuarios cualquier operación sobre objetos de datos en cualquier momento, lo que implica un WYSIWIS (What you see is what i see) relajado y trabajo simultaneo.
- **Workspace Awareness:** El sistema debe soportar una variedad de características relacionadas a la conciencia de grupo dentro de los espacios de trabajo, de tal manera el usuario puede saber quien esta en su grupo de trabajo, donde se está trabajando y que es lo que se está trabajando.
- **Administración de sesión:** El sistema debe proveer una administración de sesión ligera y flexible, para soportar que usuarios administran el trabajo con poco esfuerzo.
- **Control de interacción flexible:** El sistema debe proporcionar una variedad de control de interacción paradigmas / políticas que van desde la interacción simultánea y libre, hasta la interacción secuencial y sincronizada con el fin de facilitar la eficacia de diferentes tareas de colaboración.

Debido a que la refactorización de un sistema puede llegar a tornarse bastante compleja, existen trabajos previos que sirven como marco de referencia para futuros trabajos, tal es el caso de CoWord y CoPowerPoint aplicaciones colaborativas realizadas a partir de Word y PowerPoint respectivamente [Qian Xia, 06].

2.4.1. Arquitectura centralizada y replicada

Al momento de realizar un sistema colaborativo, se debe escoger la arquitectura que soporta la aplicación, la arquitectura de los sistemas colaborativos puede ser clasificada en dos grupos, la arquitectura centralizada y la arquitectura replicada (Lauwers, 90).

Con la arquitectura centralizada, existe únicamente una instancia de la aplicación compartida ubicada en un sitio central. Los demás sitios colaborativos cuentan simplemente con sistemas clientes que cuentan con funciones limitadas de acuerdo al rol que desempeñan dentro del plan de trabajo. El cliente básicamente dispara eventos para la aplicación compartida, dichos eventos generan información en el sistema central de la aplicación, la cual luego es distribuida y mostrada en el o los sistemas cliente.

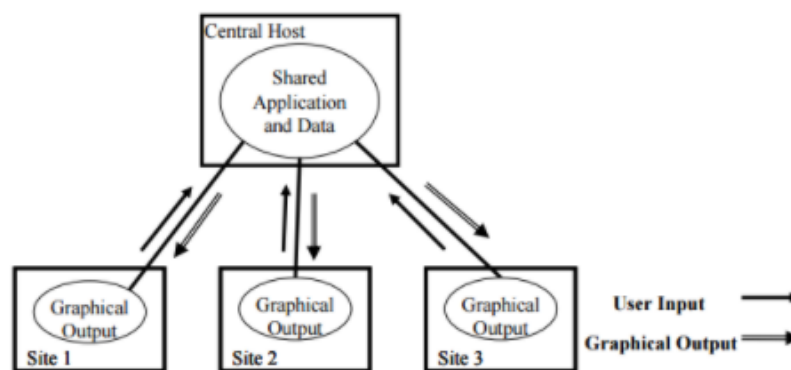


Figura 2.3: Arquitectura centralizada. Fuente: Lauwers 90

La principal ventaja de la arquitectura centralizada es su facilidad de implementación a comparación de la arquitectura replicada puesto que solamente se tiene que mantener una instancia de la aplicación compartida, los desarrolladores no tienen que poner mucha energía en la consistencia de la información. Pero la centralización también presenta desventajas, la más significativa es la lentitud en la respuesta local, cada evento local debe ser enviado a la aplicación compartida, la vista local no puede ser actualizada hasta que la información sea recibida correctamente en la aplicación central. La espera puede tornarse muy alta mientras mayor sea el retardo de la red.

Dichos problemas motivaron la arquitectura replicada, en la que cada uno de los sitios colaboradores tiene una instancia de la aplicación compartida, corriendo en el sitio local. La arquitectura replicada es capaz de realizar una respuesta rápida gracias a que los eventos disparados por el usuario pueden ser ejecutados de manera local antes de ser enviados a los sitios remotos.

De cualquier manera la arquitectura replicada también presenta problemas, el más significativo es mantener la consistencia. Si se permite a los usuarios interactuar con su aplicación local y replicar libremente, los eventos pueden ser ejecutados en diferente orden en los distintos sitios distribuidos. Sin embargo la arquitectura replicada facilita el WYSIWIS y también elimina el retardo en el tiempo de respuesta de la aplicación.

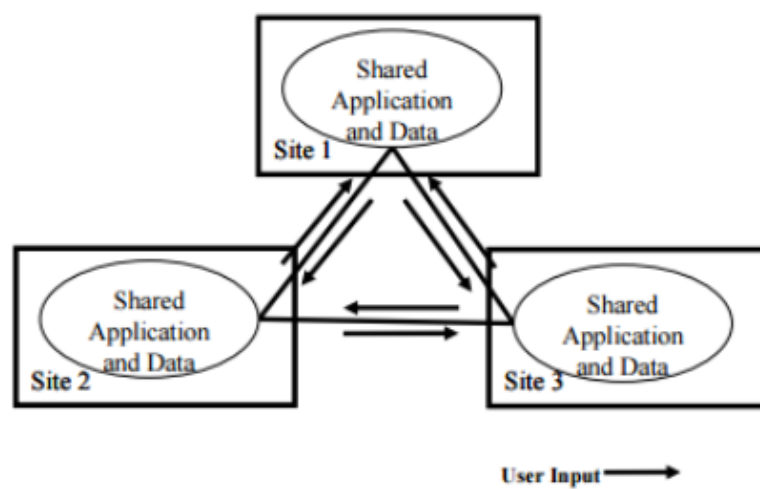


Figura 2.4: Arquitectura replicada. Fuente: Lauwers 90

Capítulo 3

Interfaces De Usuario

Dentro de este capítulo se hablará sobre algunas características importantes que se tienen que tomar en cuenta para el diseño de interfaces de usuario y algunas técnicas que se utilizan para fundamentar el desarrollo del presente proyecto.

3.1. Interfaz De Usuario

Se puede definir a la interfaz de usuario como el elemento encargado de permitir la comunicación del usuario con la máquina o el sistema. Dada esta definición, necesitamos describir dos elementos importantes para poder entender mejor a una interfaz de usuario:

- Elementos de entrada: Elementos de información ingresados por el usuario, como ser: órdenes, comandos o cualquier otro elemento que el usuario pueda ingresar como pauta de lo que quiere conseguir.
- Elementos de salida: Son los elementos obtenidos de las acciones ejecutadas por el usuario, que acercan al mismo a obtener lo que espera y lograr así su meta.

Las interfaces de usuario se pueden clasificar en tres tipos: PUI GUI, y TUI. Los cuales son descritos brevemente a continuación.

- **PUI (Physical User Interface - Interfaz de usuario física):** Son las interfaces de usuario físicas, generalmente presentes en sistemas físicos, como ser maquinaria, autos, u otros, son los medios mecánicos mediante los cuales un usuario puede comunicarse con un sistema determinado.
- **GUI (Graphical User Interface - Interfaz de usuario gráfica):** Son las interfaces gráficas generalmente presentes en sistemas computacionales o sistemas digitales, constan de elementos gráficos (botones, texto, imágenes, etc.) dentro de una pantalla en el computador.
- **TUI (Tangible User Interface - Interfaz de usuario tangible):** Las Interfaces de usuario tangibles emplean objetos reales para representación y control en un medio computacional. Este tipo de interfaz de usuario está presente generalmente en sistemas de entrenamiento o simulación, donde por ciertos motivos no se puede interactuar con un sistema real pero se intenta que la experiencia del usuario sea lo más parecida a ella.

Las GUI han ido evolucionando de la mano de la tecnología, podemos ver claramente esto en su historia, cuando se pasó de un texto plano siendo ejecutado como órdenes al computador, a un elemento visual orientado a satisfacer distintos tipos de necesidades de los usuario. Con la evolución también se generaron criterios que nos permiten diferenciar la calidad de las GUI, permitiendo de esta manera un mayor desarrollo y aprovechamiento de este elemento tan importante. Entre dichos criterios sobresalen enormemente dos, que son: la usabilidad y la naturalidad.

3.1.1. Usabilidad

La usabilidad, es una característica muy importante para obtener la aprobación del usuario en cualquier sistema, la misma se define como el elemento que mide el grado en el que el usuario puede realizar sus actividades, mientras ms sencillo le resulte al usuario realizar sus actividades más usable será la interfaz, por ende más usable será el sistema. La usabilidad abarca dos conceptos fundamentales:

- **Funcionalidad:** Que se refiere a las características del sistema que permiten al usuario realizar sus tareas.
- **Facilidad de uso:** Se provee características para que el sistema sea ms fácil de aprender y permite realizar tareas más fácilmente.

Cuando se usa el término usabilidad los programadores se enfocan especialmente en el primer concepto, dejando de lado factores esenciales como la experiencia del usuario cuando intenta usar el sistema, se olvida preguntarse, el usuario puede encontrar lo que busca?, El sistema le permite realizar su trabajo más fácil? Se puede decir que la usabilidad consiste de seis factores los cuales son presentados a continuación:

- Apto para el uso (Funcionalidad): El sistema soporta las tareas que tienen los usuarios en la vida real.
- Fácil de Aprender: El sistema es fácil de usar para distintos grupos de usuarios.
- Eficiencia en las Tareas: Cuan eficiente es el sistema para los usuarios frecuentes.
- Fácil de Recordar: Cuan fácil de recordar es para los usuarios ocasionales.
- Satisfacción Subjetiva: Cuán satisfecho est el usuario con el sistema.
- Entendible: Cuán fácil es entender lo que el sistema está haciendo. Este factor es muy importante en ciertas ocasiones como ser fallas del sistema o errores.

3.1.2. Naturalidad

La naturalidad es una característica que se enfoca más a cuán natural valga la redundancia es para el usuario realizar sus actividades, por ejemplo los usuarios en general están acostumbrados a salir de los programas mediante un botón en una de las esquinas superiores de la ventana, dicho botón está representado por una x, cambiar esto podría significar un choque con lo que el usuario

conoce, esto le restaría naturalidad a la interface que se está desarrollando, por lo tanto lo que se intenta con esta característica es presentarle al usuario elementos que ya conoce para poder estimular su memoria y facilitar su trabajo, pues se dice que la memoria es la forma más simple de inteligencia. Dadas las anteriores definiciones podemos continuar describiendo el proceso de diseño de una interfaz de usuario con algunas heurísticas para lograr un mejor resultado.

3.2. Diseño de interfaces gráficas de usuario

Soren Lauesen sostiene que, si consultamos a un programador sobre la interfaz de usuario, el responde que es un elemento sencillo, que se puede realizar después de haber concluido las tareas más importantes del sistema, en cambio si trasladamos la misma pregunta al diseñador de interfaces, este nos dice que es uno de los elementos más importantes dentro del desarrollo de un proyecto, y que por ende debe realizarse antes que las demás tareas, y en participación con programadores, clientes, diseñadores y otros. Podemos evidenciar que se tienen puntos de vista diferentes, y se puede afirmar que ambos tienen algo de razón, hacer una interfaz de usuario es una tarea relativamente sencilla, pero hacer una interfaz de usuario que agrade al cliente es ya una tarea mucho más complicada. Existen diversas técnicas que pueden ser usadas dentro del diseño de interfaces una de las más importantes y la que se tratara a lo largo de este artículo es el sketch (prototipado) y sus ventajas.

3.2.1. Sketch Design

Sketch es una técnica de diseño muy ventajosa en diversos aspectos, este método de diseño se basa esencialmente en el prototipado. El prototipado es una práctica muy usada en el ámbito de la arquitectura, por el costo que implica no se puede realizar pruebas reales en las que las personas puedan caminar, así que se realizan modelos en papel, maquetas, planos, etc. Para permitir que el o los usuarios puedan opinar sobre el producto antes de que el mismo esté terminado, esta práctica es muy útil a la hora de ahorrar tiempo y dinero. El sketch modeling se basa en desarrollar modelos mediante dibujos de la interfaz de usuario. Por qué realizar prototipos? Los prototipos son fáciles y rápidos de desarrollar, fáciles de cambiar y fáciles de desechar, permite ahorrar tiempo y dinero, permite contar con diferentes alternativas, solucionar problemas antes de que el código sea escrito y mantener el enfoque del diseño en el usuario. El prototipado de una interfaz de gráfica puede tener diversos tipos de fidelidad, la fidelidad de un prototipo se mide de acuerdo al nivel de detalle que se usó al momento de realizar el prototipo, existen 3 niveles de fidelidad: low fidelity, médium fidelity, high fidelity.

- **Low Fidelity** Son los prototipos de fidelidad baja, que son llamados trazos artísticos, el prototipo está formado por dibujos simples y de baja representacin, como se muestra en la siguiente figura 3.1.
- **Medium Fidelity** Los prototipos de fidelidad media son prototipos con una semejanza mayor a los modelos reales, aunque siguen siendo dibujos estos dibujos poseen un nivel mayor de semejanza y detalle. En la figura 3.2 se muestra un ejemplo de un prototipo de fidelidad media.



Figura 3.1: Ejemplo de un prototipo de baja fidelidad.

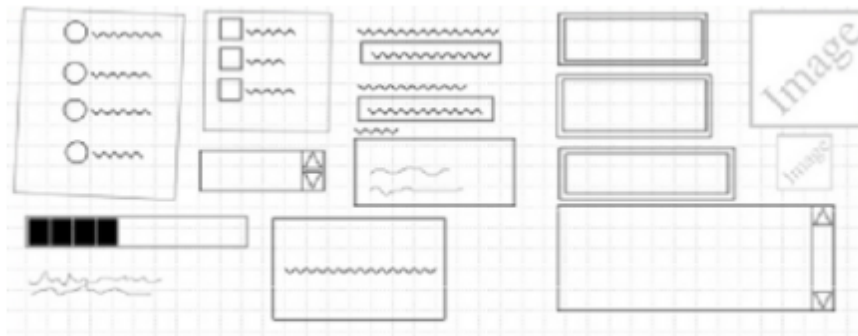


Figura 3.2: Ejemplo de un prototipo de fidelidad media.

- **High Fidelity** Son prototipos de una calidad más elevada, constan de un parecido mucho mayor a lo que se quiere representar, su detalle es mucho más alto por ende es más representativo. En la figura 3.3 podemos apreciar un ejemplo de prototipo de alto nivel de fidelidad.

El presente proyecto plantea realizar un sistema colaborativo para el diseño de interfaces de usuario mediante prototipos en alta fidelidad, de esta manera permitir a los usuarios ver el resultado más aproximado al resultado final.

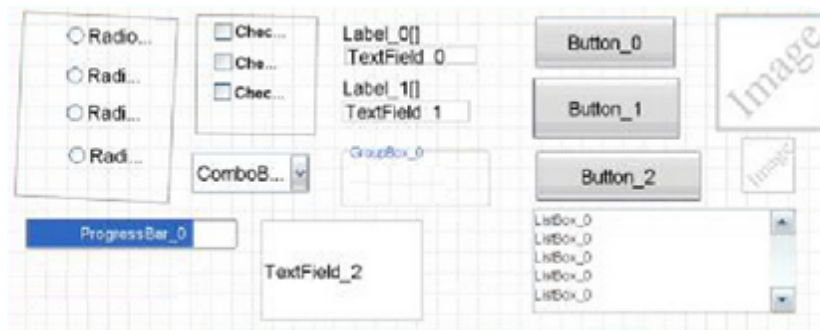


Figura 3.3: Ejemplo de un prototipo de fidelidad alta.

Capítulo 4

Adaptación

4.1. Adaptación transparente y transformación operacional

Los sistemas groupware en tiempo real son construidos generalmente guiados por uno de los siguientes enfoques: conciencia de colaboración (collaboration awareness) y transparencia de colaboración (collaboration transparency). Con el primer enfoque los groupware son desarrollados especialmente con el propósito de soportar la colaboración y con mecanismos internos de conciencia de colaboración desde su diseño. Con el segundo enfoque, los groupware que están basados en sistemas existentes o nuevos, presentan mecanismos externos que no fueron pensados en el diseño. [Qian Xia, 2006]. Una ventaja de la colaboración transparente es que permite a los usuarios realizar colaboraciones con las aplicaciones que le son familiares, ¿Quién abandonaría su editor de texto favorito para usar una aplicación de autoría colaborativa? - Who would abandon their favorite word processors to use a co-authorship application? [Grudin, 1994 b], asimismo el agregar funcionalidades de colaboración en aplicaciones mono-usuario reduce el esfuerzo de desarrollar groupwares.

4.2. Adaptación Transparente

A diferencia de otras técnicas de transformación de aplicaciones en aplicaciones compartidas, la adaptación transparente (TA) intenta transformar de forma transparente una aplicación mono-usuario en una versión colaborativa del mismo, en el caso específico del proyecto, se trabajó únicamente con ambientes homogéneos, lo cual significa que se requiere el uso de la misma aplicación convertida en las sesiones de colaboración. En los fundamentos de la adaptación transparente, se trabaja con la API (Application Programming Interface - Interfaz de programación de la aplicación) para de esta manera interceptar las acciones realizadas por los usuarios y así sin modificar el código fuente de la aplicación permitir la comunicación del sistema mono usuario con los demás miembros de la sesión. El proyecto sobre el que se trabajó no cuenta con una API como tal, para poder realizar la interceptación de los eventos, pero tras analizar la estructura del proyecto, se identificaron las operaciones claves a interceptar, es en este momento en el que se decidió usar herramientas de la Transformación Operacional (OT) para manipular e interceptar los eventos generados, en este caso modificando levemente el manejo de los eventos dentro de la aplicación de tipo mono usuario.

El núcleo de la aplicación de diseño de interfaces se centra en tres acciones fundamentales:

- Agregar componente.
- Editar componente.
- Eliminar componente.

Para poder llevar a cabo una adaptación transparente como tal se pide poner especial énfasis en los siguientes puntos:

Compatibilidad de la aplicación: Las características y funcionalidades de la interfaz de usuario así como los formatos de los documentos deben ser mantenidos.

Transparencia de la aplicación: No se deben realizar cambios al código original de la aplicación en lo posible. Esta característica no pudo ser completada a falta de una forma de comunicación con la aplicación sobre la que se realiza el trabajo, por lo que se hizo énfasis en modificar lo mínimo necesario para poder realizar la transformación de la manera más transparente posible.

Respuesta local rapida: La respuesta del sistema una vez modificado debe ser lo más parecida a la versión original. Para poder mantener esta característica puesto que como enfoque se escogió una arquitectura centralizada, se restringe que los miembros del equipo se encuentren de momento sobre la misma red.

Colaboración sin restricciones: Los usuarios deben ser libres de realizar la mayor cantidad de operaciones con los datos en cualquier momento, lo que implica un WYSIWIS (What you see is what I see - Lo que ves es lo que veo) relajado y trabajo concurrente. En un ambiente de WYSIWIS estricto, todos los integrantes del grupo de trabajo tienen que estar estrictamente viendo lo mismo en un mismo ambiente. En cambio en un ambiente de WYSIWIS relajado si bien los integrantes se encuentran sobre un mismo sistema, presentan la libertad de moverse y realizar acciones diferentes a las de sus compañeros de piso.

Conciencia de espacio de trabajo: El sistema deber soportar una variedad de caractersticas de conciencia de espacio de trabajo, para permitir a los usuarios conocer quien est trabajando con ellos y que estn realizando.

4.3. Modelo de adaptación

Para comenzar con la adaptacin del sistema, se utilizo el sistema para identificar las acciones fundamentales ya descritas, existen diferentes formas de realizar una misma acción dentro del sistema, y revisando la estructura de la forma en la que se manejan los eventos de los usuarios, se detectó que una misma acción realizada desde diferentes lugares dentro de la aplicación se traducen en diferentes acciones dentro de la misma 4.1, por lo que para comenzar con la transformación del proyecto primero se dio paso a una ligera refactorización del sistema.

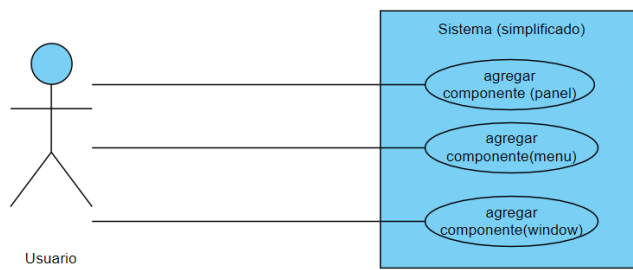


Figura 4.1: Estado inicial del sistema, ejemplo.

Bibliografía

Referencias

- Brown, J. S., Cashman, P. M., Malone, T., y Greif, I. (1985). Interfaces in organizations (panel session): supporting group work. En *Acm sigchi bulletin* (Vol. 16, p. 65).
- de Souza, C. R., Redmiles, D., Cheng, L.-T., Millen, D., y Patterson, J. (2004). Sometimes you need to see through walls: a field study of application programming interfaces. En *Proceedings of the 2004 acm conference on computer supported cooperative work* (pp. 63–71).
- Ellis, C. A., Gibbs, S. J., y Rein, G. (1991). Groupware: some issues and experiences. *Communications of the ACM*, 34(1), 39–58.
- Evolus. (2010). Pencil the open source gui prototyping tool.. (<http://pencil.evolus.vn/en-US/Home.aspx> en 7 mayo 2012)
- Garrido, J., Gea, M., Gutierrez, F., y Padilla, N. (2000). Designing co-operative systems for human collaboration. En *Designing cooperative systems: The use of theories and models: Proceedings of the 5th international conference on the design of cooperative systems (coop'2000)* (Vol. 58, p. 399).
- Gutwin, C., Greenberg, S., y Roseman, M. (1996). Workspace awareness in real-time distributed groupware: Framework, widgets, and evaluation. En *People and computers xi* (pp. 281–298). Springer.
- Wilson, P. (1991). *Computer supported cooperative work:: An introduction*. Springer Science & Business Media.