

ANEXO B: PRÁCTICAS

Práctica 1 – Análisis, visiones y descripción del dominio

Introducción

En esta práctica iniciarás las actividades para poder desarrollar una interfaz de usuario que satisfaga las necesidades de los usuarios. Para esto es necesario entender el dominio, el porqué un nuevo sistema es necesario, así como sus objetivos. Finalmente es necesario analizar a los usuarios, las tareas que realizan, así como los datos que manejan.

Toda esta información deberá estar documentada en un documento de visión, requisitos, modelos de datos, descripciones de tareas y otros artefactos que con anterioridad la organización haya realizado para desarrollar el sistema. Sin embargo, es necesario que tú como diseñador entiendas los detalles de las funciones y características del sistema que se desea desarrollar. Así, para iniciar el proyecto, primero se realizará una investigación sobre el dominio de la aplicación.

Objetivos de aprendizaje

Desarrollar en el estudiante las habilidades para entender un dominio de aplicación con el que no está familiarizado y aplicar la técnica de investigación documentan para:

- Definir el alcance del sistema de software.
- Realizar una descripción del dominio que incluya datos y tareas del usuario.

Descripción de la práctica

1. Investiga cuáles son los objetivos de negocio de la organización en donde se usará el software a desarrollar. Idealmente esta información se podrá encontrar en la visión y misión de la organización. Puedes utilizar un motor de búsqueda si es que la organización cuenta con sitio web.
2. Ahora reflexiona y realiza una lista de cómo la organización alcanza o trata de alcanzar sus objetivos de negocio. Es probable que haya indicios de que la organización haya hecho esfuerzos anteriores. Investiga más información para complementar tu lista y si es posible incluye tareas que realicen las personas de la organización.
3. Investiga si la organización ya utiliza algún o algunos sistemas para alcanzar su objetivos. ¿Cuáles son sus problemas? ¿Existen algunas limitantes? Realiza un análisis de la interfaz de usuario de los sistema que existan para identificar posibles requisitos para el nuevo sistema.
4. Organiza tus hallazgos de tal manera que sean fácil de consultar y que puedan servir para las siguientes actividades del desarrollo.
5. Ahora realiza una lista de los hallazgos con cuestiones pendientes de entender y de las que haya posiblemente poca información.
6. Finalmente, realiza un presentación con una extensión de 8 a 9 láminas con los hallazgos más importantes.

Materiales

Para esta práctica necesitarás lo siguiente:

- Software para desarrollar presentaciones.
- Procesador de textos.

Bibliografía

Lauesen, S. (2005). *User Interface Design: A Software Engineering Perspective*. Addison-Wesley Longman Publishing Co., Inc.

Práctica 2 – Entrevistas e inicio del diseño de interfaz de usuario

Introducción

En esta práctica realizarás el guion o guiones de las entrevistas necesarias para conocer a detalle las tareas que realizan actualmente las personas a las que va dirigido el producto de software, así como los datos que utilizan además de aclarar los elementos que no hayan quedado claros en la investigación realizada en la Práctica 1. La técnica de entrevista es muy utilizada en las primeras actividades del desarrollo de software y para poder obtener los mejores resultados es necesario prepararla con tiempo y contemplando todas las cuestiones de la logística necesaria para realizarla.

Objetivos de aprendizaje:

Desarrollar en el estudiante las habilidades para llevar a cabo una entrevista con interesados en un productos de software realizando lo siguiente:

- Obtener información sobre el dominio del sistema.
- Conocer las tareas que se llevan a cabo en el dominio.
- Validar los hallazgos de la Práctica 1.

Descripción de la práctica

1. El primer paso es determinar a quién se entrevistará. Preferentemente deberá ser por lo menos una persona de cada uno de los grupos a los que vaya dirigido el software. Es importante asegurarte de que dicha persona trabaje en el día a día de la organización.
2. Escribe una breve explicación de porqué es necesario entrevistar a cada una de las personas. Dicho texto lo utilizarás como una introducción a momento de realizar la pregunta. En ella debe quedar claro los objetivos del sistema a desarrollar (los que se tengan al momento), el porqué lo estamos entrevistando, el qué se hará con las respuestas y explicar las medidas de confidencialidad de la información recabada
3. Ahora inicia con el guion de entrevista. Plantea preguntas que abarquen las tareas que realizan, así como problemas en las áreas de trabajo o cuestiones que se pueden mejorar. El objetivo deberá identificar las tareas y los problemas principales.
4. Es momento de redactar preguntas más detalladas para obtener descripciones de las tareas, así como los datos que se utilizan. Procura que las preguntas indaguen sobre: el área de trabajo, el perfil del usuario, las tareas que realizan, los problemas actuales y otras personas que sería deseable entrevistar.
5. Prepara la versión final de tu o tus guiones para obtener retroalimentación de tu profesor.
6. Aplica la entrevista utilizando tu guion. Toma todas las medidas logísticas para que tu entrevista sea exitosa.
7. Ahora, con la información recabada tanto en esta práctica como en la anterior elabora el documento en donde se plasmen los objetivos de negocio que la solución ayudará a alcanzar, los requisitos funciones, las interfaces técnicas, así como los requisitos de calidad . Consulta la bibliografía de la práctica para encontrar ejemplos e información sobre los elementos que se solicitan.
8. Finalmente, elabora el modelo de datos, la descripción de datos, la descripción de tareas y el área de trabajo como se muestra en las secciones 5.2, 5.3 y 5.4 del libro de Soren Lauesen respectivamente. Cuida de utilizar tanto los formatos sugeridos por el autor como de validar tus productos con expertos en el dominio.

Materiales

Para esta práctica necesitarás lo siguiente:

- Procesador de textos.

Bibliografía

Lauesen, S. (2005). *User Interface Design: A Software Engineering Perspective*. Addison-Wesley Longman Publishing Co., Inc.

Práctica 3 – Diseño de la interfaz de usuario

Introducción

En esta práctica continuarás con el diseño de la interfaz de usuario del sistema a desarrollar. Para esto, será necesario establecer en un documento en donde se plasme el diseño de ventanas virtuales. Las ventanas virtuales es una presentación de datos persistentes orientada a usuarios en donde se muestran datos, pero no botones u otras funciones. Posteriormente identificar las funciones que se encuentran asociadas a las ventanas y tareas identificadas.

Objetivos de aprendizaje:

El estudiante podrá desarrollar las habilidades desarrollar interfaces de usuario equilibradas.

Descripción de la práctica

1. De las tareas importantes y frecuentes identifica los datos que el usuarios le gustaría ver para completar cada tarea. Esto nos servirá para la planeación de las ventanas.
2. Agrupa los datos en pocas ventanas y señala el contenido de cada una. Repite para cada tarea.
3. Si alguna tarea reutiliza datos que ya están planteados en las ventanas identificadas considera reutilizar dichas ventanas. Si se utiliza información adicional, considera agregarla a las ventanas que ya se tenga identificada.
4. Continúa con los pasos anteriores hasta haber cubierto todas las tareas y hayas considerado las reglas de diseño propuestas por Soren Lausen en la sección 6.1.
5. Una vez terminada la planeación, por cada ventana realiza una ventana gráfica detallada. Recuerda que solo se deben presentar datos (no botones, menús u otras funciones).
6. Agrega información real a todas las ventanas. Considera utilizar datos extremos para evaluar si la ventana se ve bien aún en estos casos.
7. Para verificar tu trabajo, por cada ventana hay que imaginar que eres el usuario y que quieres realizar la tarea. Ahora identifica si falta alguna información. Si es así, corrige. Adicionalmente se recomienda realizar una verificación CREDO.
8. Para continuar con el diseño de funciones, toma una de las tareas importantes y frecuentes. Ahora imagina que estás realizando la tarea. Identifica cuáles son las acciones (buscar, crear elementos, guardar datos, etc.) que son necesarias para realizar la tarea, qué ventanas son necesarias ver para que la tarea prosiga.
9. Cuando hayas identificado un botón, dale un nombre y especifica qué es lo que debería realizar. Puedes utilizar minidescripciones. Asócialo con la ventana respectiva.
10. Recorre todas las variantes de la tarea para que puedas identificar todas la funciones. Realiza lo mencionado en el paso 9 cuantas veces sea necesario.
11. Recorre cada una de las tareas identificadas y repite los pasos 9 y 10 cuantas veces sea necesario.

Materiales

Para esta práctica necesitarás lo siguiente:

- Procesador de textos.

Bibliografía

Lauesen, S. (2005). *User Interface Design: A Software Engineering Perspective*. Addison-Wesley Longman Publishing Co., Inc.

Práctica 4 – Diseño de prototipo de Interfaz de usuario

Introducción

A continuación, culminarás el diseño de la interfaz de usuario mediante el desarrollo de un prototipo y la interfaz de usuario que utilizará el sistema. Aquí, tanto las ventanas virtuales como las funciones se conjuntarán en primer lugar en un prototipo para su evaluación y cristalización en la interfaz de usuario.

Objetivos de aprendizaje:

El estudiante desarrollará las habilidades para realizar un prototipo de la interfaz de usuario y su evaluación para culminar con una propuesta de interfaz de usuario.

- Definir la interacción de los usuarios con el sistema mediante un prototipo.
- Evaluar de manera temprana la usabilidad del sistema.

Descripción de la práctica

1. Decide el tipo de prototipo a desarrollar en cuanto a su fidelidad, así como las herramientas de prototipado apropiadas.
2. Realiza una versión de cada una de las ventanas. Revisa que se refleje todo lo que has identificado en las prácticas anteriores.
3. Agrega los menús necesarios a todas las ventanas.
4. Ahora realiza las ventanas que contendrán las ventanas de error.
5. Finalmente agrega todos los combo boxes, los diálogos, ayudas de texto y otros elementos de la interfaz gráfica que el usuario pueda necesitar.
6. Procura evaluar tu prototipo junto con los interesados. Revisa las recomendaciones para la evaluación en la bibliografía de la práctica.
7. Corrige los errores identificados en el prototipo y prosigue con el diseño de la interfaz de usuario real.

Materiales

Para esta práctica necesitarás lo siguiente:

- Herramienta de prototipado.
- Procesador de textos.

Bibliografía

Lauesen, S. (2005). *User Interface Design: A Software Engineering Perspective*. Addison-Wesley Longman Publishing Co., Inc.

Práctica 5 – Desarrollo del modelo de casos de uso y del modelo de conceptos de negocio

Introducción

En todo proceso de Ingeniería de Software, la etapa de requerimientos resulta importante. No importa cuanto tiempo se les dedique a los requerimientos, nunca es suficiente para tener todo correcto ni completo. De ahí la importancia de que los componentes puedan adaptarse al cambio; es decir, que sean fáciles de modificar.

En la etapa de definición de requerimientos de acuerdo con el proceso de Cheesman & Daniels, se desarrollarán los artefactos mínimos para continuar con los siguientes flujos de trabajo. Como lo marcan los autores “tener claro el propósito y metas del software antes de iniciar su construcción”.

En esta práctica, desarrollarás los pasos necesarios para obtener los dos principales artefactos de este flujo de trabajo: el modelo de casos de uso y el modelo de conceptos de negocio.

Objetivos de aprendizaje:

Desarrollar en el estudiantes las habilidades para realizar la etapa de requerimientos del desarrollo basado en componentes para el proyecto del curso, obteniendo los siguientes artefactos:

- Modelo de casos de uso
- Modelo de conceptos del negocio

Descripción de la práctica

Parta esta práctica a continuación los pasos para elaborar los siguientes artefactos:

- Modelo de casos de uso
- Modelo de dominio, ya que el modelo de tipos de negocio es algo similar

Se asume que ya estás al tanto de los procesos de negocio que el sistema debe soportar. Idealmente deberías contar con un modelo de procesos elaborado con un diagrama de actividades. Si no lo tienes, no te preocupes. Únicamente asegúrate de que tienes toda la información cerca de los procesos del negocio y de que la has entendido correctamente.

Pasos para elaborar el modelo de conceptos de negocio.

1. Lista todos los conceptos relacionados con el problema, piensa que estás elaborando un mapa conceptual.
2. Emplea un diagrama de clases para representar los conceptos del negocio, pero ten cuida ya que no se está modelando software.
3. Para identificar conceptos, recuerda usar las designaciones.
4. Ayúdate de una herramienta como StarUML o Enterprise Architect para elaborar este modelo.

5. En el diagrama, pon relaciones entre los conceptos, pero como es una especie de mapa mental, no puede haber agregaciones o composiciones, aunque sí puedes, identifica de una vez la herencia.
6. Revisa tu modelo, por si hubiera conceptos o relaciones faltantes.

Pasos para elaborar los casos de uso.

1. Los casos de uso corresponden a responsabilidades de parte del sistema de realizar alguna funcionalidad asociada a los procesos del negocio, así que revisa los procesos del negocio para identificar los casos de uso.
2. Dibuja los distintos casos de uso que hayas identificado en el paso anterior. Te puedes ayudar de StarUML o Enterprise Architect.
3. Los pasos o funcionalidad que no realiza los procesos de negocio deben ser realizados por actores.
4. Revisar casos en que se necesiten actores más genéricos (un administrador de . . .) para ponerlo en el modelo de conceptos de negocio agregando una relación de herencia hacia el actor en específico.
5. Revisar los eventos que disparan los pasos del proceso de negocio para identificar casos de uso.
6. Revisar los casos de uso iniciales contra el modelo de conceptos, mediante las siguientes preguntas:
 - a. ¿Los conceptos representados por las cajas se crean y destruyen? ¿El software necesita saber de ello? Si es así ¿Cómo se da cuenta? ¿El concepto tiene atributos que pueden cambiar?
 - b. Las relaciones (asociaciones) entre los conceptos ¿Pueden cambiar en e tiempo? Si es así ¿El software necesita saberlo? ¿Cómo se da cuenta de que cambian?
7. Elabora una redacción como la de las tablas 4.1 y 4.2 del libro.
8. Elabora ahora el modelo de casos de uso.
9. Elabora las descripciones de los CU. Recuerda que estos CU son de requerimientos o análisis. En la descripción incluye: ID, nombre del CU, Actor(es), Objetivo, Descripción, Flujo normal, Flujo alterno, Flujo de excepción.
10. Como paso final, elabora la calidad de servicio se cada CU, piensa en tiempo de respuesta, número de usuarios soportados, seguridad y facilidad de uso.

Materiales

Para esta práctica necesitas contar con:

- Herramienta de modelado, ya sea StarUML o Enterprise Architect (EA).
- Procesador de textos.

Bibliografía

Cheesman, J., & Daniels, J. (2000). *UML Components: A Simple Process for Specifying Component-Based Software*. Addison-Wesley Longman Publishing Co., Inc.

Práctica 6 – Identificación de componentes

Introducción

La identificación de componentes es la primera fase del flujo de trabajo de especificación. Parte del flujo de trabajo de requisitos y recibe como entradas tanto el modelo de conceptos de negocio, así como el modelo de casos de uso. De acuerdo con Cheesman & Daniels, el objetivo de esta fase es crear un conjunto inicial de interfaces y especificaciones de componentes, así como una primera versión de la arquitectura de componentes. En esta práctica, desarrollarás los pasos necesarios para obtener los artefactos anteriormente mencionados, así como el modelo de tipos de negocio.

Objetivos de aprendizaje:

Desarrollar las habilidades en el estudiante para descubrir la información e interfaces que se necesitan manejar, los componentes que se necesitan para proveer dicha funcionalidad y cómo estos encajan obteniendo los siguientes artefactos:

- Modelo de tipos de negocio
- Interfaces de sistema
- Interfaces de negocio
- Especificaciones iniciales de componentes

Descripción de la práctica

Es importante recordar que el enfoque de la propuesta de Chessman & Daniels se centra en sistema de negocio, es decir, el aspecto independiente de la interfaz de usuario. Una aplicación es dividida en capas y es la capa de diálogo de usuario la que actúa como iniciadora de las operaciones de las interfaces de sistema y es la que implementa la lógica del caso de uso.

Las interfaces de sistema emergen de un análisis del modelo de casos de uso, específicamente de las interacciones del sistema. Cuando un usuario inicia un caso de uso, la lógica del caso de uso crea y despliega la GUI adecuada y se guía al usuario paso por paso. Cuando la lógica del caso de uso necesita información que desplegar o necesita notificar al sistema la acción del usuario, ésta llama a la operación adecuada la cual usa operaciones definidas en la lógica de negocio para llevar a cabo su función.

Por otro lado, para concentrarnos en la información que necesitará manejar el sistema se refine el modelo de conceptos de negocio para obtener una representación del punto de vista del sistema, es decir, el modelo de tipos de negocio.

Pasos para identificar las interfaces de sistema y operaciones.

7. Define un tipo diálogo y una interfaz de sistema por caso de uso.
8. Recorre cada descripción de caso de uso, paso por paso, y considera si hay responsabilidades del sistema que se deban ser modeladas. Si es así, represéntala como una o más operaciones en la interfaz adecuada.
9. Si hay varios pasos que son todos responsabilidad del sistema estos se pueden representar como una sola operación, pero no lo hagas si crees que se puedan necesitar por separado más adelante (en una extensión, por ejemplo).

10. Ayúdate de una herramienta como StarUML o Enterprise Architect para elaborar este modelo, recuerda utilizar un diagrama de clases con los estereotipos adecuados.
11. Si necesitas un ejemplo, puedes revisar la sección 5.2.1 y 5.2.2 del libro.

Pasos para identificar las interfaces de negocio.

11. Realiza una copia del modelo de conceptos de negocio. Nombra a la copia como Modelo de tipos de negocio. Recuerda que necesitarás ambos diagramas, por lo que no debes eliminar el modelo de conceptos de negocio. Estos se relacionarán con una dependencia <<trace>>
12. En este nuevo diagrama clases, refina el modelo de tipos de negocio y especifica reglas de negocio adicionales con restricciones. Recuerda que aquí los tipos de negocio representan datos (estados) que el negocio necesita mantener y monitorear. Además, pueden corresponder a aspectos NO físicos del negocio como procesos.
13. Identifica los **Core Business Types**, márcalos con el estereotipo <<core>>. Recuerda que un tipo Core se caracteriza por a) tener un identificador de negocio usualmente independiente de otros y b) tienen existencia independiente, es decir, no tienen asociaciones obligatorias a excepción de las de categorización (TipoDeHabitacion con Habitación, por ejemplo).
14. Identifica las interfaces de negocio para los tipos Core y agrégalas al modelo de tipos de negocio.
15. Sigue refinando el modelo para indicar las responsabilidades de las interfaces de negocio. Remueve o adiciona elementos hasta que el alcance sea el correcto.
16. También agrega atributos en cada tipo y multiplicidad de acuerdo con las restricciones.
17. Si necesitas un ejemplo de como se debería ver tu diagrama, consulta la figura 5.8 del libro. Por otro lado, estudia el ejemplo en la sección 5.3.3 para ver como se manifiestan reglas de negocio en el diagrama.
18. Crea una interfaz de negocio por cada tipo Core en el modelo de tipos de negocio. Cada interfaz de negocio maneja la información representada por el tipo Core y los tipos de detalle. Dado que manejan un conjunto de tipos Core habrá que nombrarlas como lxxxMgt, por ejemplo, IClienteMgt para un tipo core Cliente. Tu diagrama deberá parecerse al que aparece en la figura 5.9 en este momento.
19. Asigna quien será responsable de las asociaciones inter-interfaces, es decir, que interfaz almacenará dicha información. Para esto deberás agregar la navegabilidad. A final de todo el proceso anterior obtendrás el diagrama de responsabilidad de interfaz.

Pasos para crear las especificaciones de interfaz iniciales:

1. Una vez que estamos satisfechos con el diagrama de responsabilidad de interfaz, crea otro conjunto de interfaces de negocio en el paquete de especificaciones de interfaz. Cada una con su propio subpaquete que corresponde a las interfaces de negocio creadas en el modelo de tipos de negocio.
2. Copia en cada subpaquete todos los tipos en el modelo de tipos de negocio de los que es responsable la interfaz. Esto será la base para el modelo de información de interfaz para cada interfaz.
3. Las especificaciones de sistema se deben crear directamente en el paquete de especificaciones de interfaz. El detalle de la estructura lo puedes observar en la figura 5.11

Pasos para la arquitectura de especificación de componentes

1. Identificar si existen entradas existentes para este proceso. Las posibles entradas son las interfaces en el modelo de interfaz, especificaciones de componentes que se quieran reutilizar, una arquitectura de especificación de componentes que necesitemos adaptar o patrones de arquitectura de especificación de componentes que queramos utilizar.
2. Crea una especificación de componentes por cada especificación de interfaz de paso anterior, sin embargo, habrá casos en los que queramos soportar múltiples interfaces en una sola especificación de componentes. Consulta la sección 5.6 para verificar dichas circunstancias.

A partir del conjunto de especificaciones de componentes juntos con sus interfaces soportadas y dependencias de interfaces, podemos vincular los componentes mediante sus interfaces y dependencias directamente. Un ejemplo de este paso lo puedes observar en las figuras 5.13 y 5.14 del libro.

Materiales

Para esta práctica necesitas contar con:

- Modelo de conceptos de negocio
- Modelo de casos de uso
- Herramienta de modelado, ya sea StarUML o Enterprise Architect (EA)
- Procesador de textos

Bibliografía

Cheesman, J., & Daniels, J. (2000). *UML Components: A Simple Process for Specifying Component-Based Software*. Addison-Wesley Longman Publishing Co., Inc.

Práctica 7 – Definir la interacción de componentes

Introducción

Una vez realizada la identificación de componentes y de sus interfaces, ahora es necesario llevar a cabo el modelado de la interacción entre componentes. Este modelado permitirá, a su vez, descubrir y definir las operaciones de las interfaces de negocio. En este modelado se usarán los diagramas de comunicación. Estos diagramas emplean objetos de componentes que corresponden a las interfaces de los componentes. Los diagramas de colaboración se emplean de manera similar a cuando se modela la interacción entre objetos.

Un artefacto adicional es el diagrama de especificación de componente, que permite definir las restricciones de la arquitectura de componentes.

Objetivos de aprendizaje:

Desarrollar en el estudiante las habilidades para realizar el modelado de la interacción entre componentes que a su vez permite definir las operaciones de las interfaces de negocio. En este proceso habrá realizado lo siguiente:

- Descubrir operaciones de negocio.
- Refinar interfaces y operaciones.
- Refinar especificación de componentes y arquitectura.

Descripción de la práctica

En la práctica anterior, se obtuvieron las interfaces del sistema, las interfaces del negocio (cada uno con su diagrama de responsabilidad de interfaz) y la arquitectura inicial de componentes. En esta práctica definiremos la interacción entre componentes. Esta interacción se definirá en términos de diagramas de comunicación. Para ello se empezará por definir las operaciones que se identificaron para las interfaces del sistema.

Se deben realizar los pasos de abajo para cada una de las operaciones definidas en una interfaz de sistema:

1. Inicia un diagrama de comunicación. Recuerda que el diagrama de comunicación muestra distintos flujos de acuerdo con el flujo de ejecución de la operación. El primer paso es el llamado a la operación de la interfaz del sistema, por lo que es necesario definir la firma.
2. Para definir la firma, identifica de una vez el parámetro(s) de la operación e inclúyelo(s) en la llamada a la operación de la interfaz del sistema en el inicio del diagrama. Puede suceder que algunos de los parámetros es un tipo de dato estructurado previamente definido. La definición de los tipos estructurados de datos se describe en el paso 6.
3. Dado que estás haciendo una invocación deberás colocar como primer objeto en el diagrama de colaboración la interfaz de sistema que corresponde. Revisa tus diagramas. Esta interfaz tiene el nombre de la forma IXXXXX. Para completar el nombre del primer objeto hazlo de la siguiente forma: objectname/rolename:classname, donde:
 - a. objectname no se coloca ya que los objetos son anónimos en este momento
 - b. rolename es la interfaz de sistema identificada previamente (IXXXXX) a la cual pertenece la operación.

- c. `classifiername` es el componente al que pertenece la interfaz de negocio.
4. Para identificar las interfaces que debes usar, puedes emplear el diagrama de la arquitectura inicial de componentes desarrollado en la práctica anterior.
 5. A continuación, analiza el o los valores de retorno de la operación e identifica sus tipos correspondientes.
 6. Cuando algunos valores de retorno están relacionados entre sí se debe a que pertenecen a un mismo elemento o entidad como por ejemplo un cliente, alumno, proveedor, hotel, etc. (según el dominio), define un tipo de dato estructurado (*structured data type*). Si es una lista de varias ocurrencias entonces será una lista de tipos de datos estructurados. Por ejemplo, una lista de información de hotel: id del hotel, nombre del hotel y una lista de tipos de habitaciones.
 7. Cuando hayas identificado tipos de datos estructurados, entonces crea el tipo de dato estructurado como una clase en un diagrama de clases y agrega el estereotipo `<<data type>>`.
 8. Nombra la clase como corresponda. Si, por ejemplo, se recupera la información de detalle de un cliente, estudiante o cualquier otro elemento, se sugiere llamar a esta clase `elementoDetalles`.
 9. Analiza el comportamiento de la operación de interfaz de sistema e identifica el siguiente objeto del diagrama de comunicación que, seguramente corresponde a una interfaz de negocio.
 10. Nombra al objeto según corresponda, siguiendo las reglas descritas en el paso 3 anterior.
 11. Identifica ahora la operación que esta interfaz de negocio debe poseer y colócala en la invocación. Recuerda que debes respetar la numeración, conforme se hace en un diagrama de comunicación.
 12. Repite los pasos 5 a 8 para esta nueva operación.
 13. Completa la firma para la interfaz de negocio. No olvides ir redactando todas las firmas de las interfaces de sistema y de las interfaces de negocio.
 14. Continúa analizando la operación de la interfaz de sistema y decide si necesitas invocar alguna operación de otra interfaz de negocio. Si es así, repite los pasos 4 a 13.
 15. Antes de dar por terminada la definición de la interacción, revisar casos especiales. Por ejemplo ¿Qué pasa si el cliente o usuario que solicita una reservación o un servicio es nuevo? Se deberá entonces agregar las operaciones necesarias a las interfaces de negocio correspondientes.

Una vez concluida la definición de la interacción, se debe elaborar el diagrama de especificación de componentes. Los pasos son los siguientes:

16. En un diagrama de clases coloca las clases que corresponden a las interfaces de negocio y a las interfaces que se detectaron como ya existentes. No olvidar que deben llevar su estereotipo `<<interface type>>`

17. Coloca una clase a la que pertenecen las interfaces del sistema. Esta clase representa el sistema que se está modelando y corresponde al <<comp spec>> previamente incluido en la arquitectura inicial de componentes desarrollada en la práctica anterior.
18. Pon las asociaciones entre el <<comp spec>> y los <<interface type>> que se requieran, pero sin olvidar la navegabilidad hacia los <<interface type>>
19. Pon la multiplicidad del lado de la flecha de la navegabilidad agregar la restricción frozen.

Para realizar las consideraciones respectivas a las referencias intercomponentes, revisa la sección 6.2.2 del libro.

Como último paso, se debe revisar si las responsabilidades de una interfaz deben dividirse en dos o más. A este proceso se le llama factorización (*factoring*). El objetivo es muy similar al de *subtyping*, es decir, separar el comportamiento general del más especializado. Revisa las recomendaciones de la sección 6.4.1 para llevarlo a cabo, si es necesario.

Las operaciones detectadas para cada una de las interfaces de negocio, has debido hacerlo en el modelo de responsabilidad de interfaces. El diagrama de especificación de componente es un diagrama nuevo. Integra estos artefactos en el modelo que ya tienes hecho del proyecto. Genera los archivos imagen o pdf necesarios para que integres un documento llamado Practica07_NumeroDeMiequipo.doc o docx o pdf y súbelo a Eminus antes de la fecha y hora indicada

Materiales

Para esta práctica necesitarás contar con:

- Herramienta de modelado, ya sea StarUML o Enterprise Architect (EA)
- Procesador de textos

Bibliografía

Cheesman, J., & Daniels, J. (2000). *UML Components: A Simple Process for Specifying Component-Based Software*. Addison-Wesley Longman Publishing Co., Inc.

Práctica 8 – Especificación de componentes

Introducción

En esta práctica trabajarás con la actividad final del flujo de trabajo de especificación de la propuesta de Chessman & Daniels. En ella se dan los pasos finales para la creación de especificaciones de componentes, es decir, la definición de los contratos de uso y de realización a partir del modelo de tipos de negocio, las interfaces, las especificaciones de componentes y la arquitectura de componentes desarrollados en la fase anterior.

Objetivos de aprendizaje:

Desarrollar en el estudiante las habilidades para realizar las actividades de la etapa de especificación de componentes para el proyecto del curso, obteniendo las versiones finales de los siguientes artefactos:

- Modelo de información de cada interfaz
- Interfaces
- Especificación de componentes
- Arquitectura de componentes

Descripción de la práctica

Recuerda que una interfaz es un conjunto de operaciones. Cada operación define una función o servicio que ofrecerá el objeto componente a un cliente y representa un contrato de “grano fino” entre ellos. Las operaciones relacionadas se agrupan en una interfaz ya que es muy probable que si un cliente usa una operación utilice las demás. Este agrupamiento lo tienes que decidir tú considerando que se favorezca la gestión de dependencias. Así que el primer paso para la especificación de interfaces es la especificación de operaciones.

Pasos para la definición de los modelos de información para interfaces:

1. Hay que definir un modelo de información de cada interfaz. Recuerda que este modelo es una representación abstracta de la información que manejará el componente a través de dicha interfaz.
2. Realiza una copia del modelo de tipos de negocio en cada interfaz de negocio. Así, en el paquete de la interfaz deberá estar esta copia.
3. Posteriormente elimina los *types*, asociaciones y atributos que no necesite tu interfaz de negocio. Recuerda de lo que debe ser responsable la interfaz y elimina en consecuencia.
4. Cuando un *type* propiedad de una interfaz hace referencia a un *type* propiedad de otra, el *tipo* de dato referenciado debe aparecer en ambos modelos de información de interfaz.
5. El modelo de información de interfaz final se debe parecer al que se muestra en la Figura 7.4 del libro. Observa que se han agregado una asociación directa del *type* “Reservation” a la interfaz. Este *type* en el diagrama original, relacionada el *type* Hotel (que ya estaba dentro de la interfaz) con el *type* Cliente (referenciado fuera de la interfaz). Realiza lo mismo con los *types* que has agregado y modifica las asociaciones derivadas a asociaciones regulares.

6. Finalmente, agrega los atributos que te simplifiquen la especificación de operaciones. En el ejemplo de la figura anterior, se agregó un atributo “claimed” de tipo booleano para facilitar la especificación de la operación “beginStay”.
7. Recuerda que una invariante es una condición agregada a un *type* que debe ser cierta para todas las instancias del *type*. Expresa la invariantes en tu modelo. Algunas se podrán representar de manera gráfica utilizando la notación de UML, por ejemplo, con la multiplicidad. En otras ocasiones será mejor escribir las invariantes en lenguaje natural o en OCL

Pasos para la especificación de pre y postcondiciones:

8. A partir del modelo de información de interfaz (*interface information model*) dibujar diagramas de instancia para aclarar el cambio de estado. Esto ayudará a la especificación de cada operación.
9. La parte del “antes” del diagrama de instancia ayuda a definir la precondición. Utiliza tus diagramas para especificar todas las precondiciones.
10. La parte “después” del diagrama de instancia te dirá cuales son las postcondiciones para que las definas. Define ahora tus postcondiciones para terminar la especificación de tu operación.

Pasos para la especificación de interfaces de sistema:

11. Ahora, para las especificaciones de sistema, considera el *business type model* de cada una de ellas. A diferencia de las interfaces de negocio, verifica que *types* necesita realmente la interfaz y cuales no. Verifica también los atributos de cada *type* que son necesarios. Los que no sean necesarios, elimínalos.
12. Cada interfaz deberá tener su propio diagrama de especificación de interfaz por separado.
13. Para las restricciones inter-interfaces, recordemos que las interfaces se han definido de manera independiente entre ellas, aunque puede haber referencias entre ellas. Debemos asegurar que los *types* usados y referenciados sean los mismos en todo momento. Para ello, agregar las restricciones de manera textual empleando OCL como se ejemplifica al final de la página 141.
14. Recuerda que en el capítulo 5 se vio como asignar reglas de negocio al modelo de tipos de negocio. Estas reglas deben establecerse en la especificación de interfaz. Revisa donde aplican, conforme al uso esperado de las interfaces en los varios sistemas en que serán usadas. Las opciones son:
 - a. Sólo en las interfaces de sistemas
 - b. Sólo en las interfaces de negocio
 - c. En ambas

Pasos para la especificación de componentes y sus restricciones:

15. Para cada interacción entre componentes habrá que incluir las restricciones necesarias para dicha interacción.

16. De los diagramas de interacción desarrollados, se deberá, para la especificación de cada componente, marcar solo el fragmento que esté directamente relacionado.
17. Revisa los modelos de información de las interfaces y determina si se necesita hacer alguna factorización.

Por último, revisa el diagrama de componentes en el que elaboraste la arquitectura inicial de componentes en la práctica 6 (Identificación de Componentes) y verifica que las interfaces provistas y requeridas sean las adecuadas, conforme a las interfaces de negocio y de sistema recién especificadas.

Materiales

Para esta práctica necesitarás:

- Herramienta de modelado, ya sea StarUML o Enterprise Architect (EA)
- Procesador de textos

Bibliografía

Cheesman, J., & Daniels, J. (2000). *UML Components: A Simple Process for Specifying Component-Based Software*. Addison-Wesley Longman Publishing Co., Inc.