

Definición del modelo del negocio y del dominio utilizando Razonamiento Basado en Casos.

Autora: MSc. Martha D. Delgado Dapena.
Centro de Estudios de Ingeniería de Sistemas.
e-mail: marta@ceis.ispjae.edu.cu

Resumen

El resultado final de un proyecto de software es un producto que toma forma a lo largo del desarrollo del proyecto. La calidad del producto final, está estrechamente ligada a la calidad del proceso de desarrollo de software, entre otros aspectos.

En el mundo actual en el que se necesita obtener productos con alta calidad, que integran variadas tecnologías y cuyo tiempo de elaboración sea mínimo, se impone mejorar el proceso de desarrollo de software, incrementando la productividad de los equipos involucrados en el desarrollo de los proyectos.

No basta con utilizar una metodología adecuada, pues en un proyecto intervienen múltiples especialistas, no todos con la misma experiencia sobre el tema específico que se aborda. Es necesario contar con herramientas que permitan aprovechar la experiencia de otros desarrolladores en el diseño de sistemas con características similares. Desarrollar sistemas que utilicen el Razonamiento Basado en Casos puede contribuir favorablemente en logro de este objetivo.

Este artículo hace una breve incursión en el Razonamiento Basado en Casos y explica dos sistemas desarrollados en el Centro de Estudios de Ingeniería de Sistemas que utilizan esta técnica para obtener el Modelo del Negocio y Modelo del dominio respectivamente.

Palabras claves: Razonamiento Basado en Casos, Inteligencia artificial, Ingeniería de Requerimientos, Ingeniería de Software, Calidad de Software.

Introducción

Desde el surgimiento mismo de la computación y a lo largo de toda su evolución se ha intentado modelar o simular el pensamiento humano y los procesos que ocurren en él. En los inicios solo se trataba de representar en las computadoras el pensamiento estructurado, los algoritmos de cálculos que podían definirse claramente como un conjunto de pasos que podían ser interpretados por las máquinas y de cierta forma sustituir o contribuir a elevar la eficiencia del ser humano en este tipo de actividades. Con el tiempo se fueron introduciendo técnicas de inteligencia artificial mucho más ambiciosas, los robots empleados en diferentes esferas de la vida económica y social son una muestra de ello.

Se hace necesario, entonces, combinar herramientas con la experiencia de los especialistas para afrontar los nuevos retos que impone diseñar sistemas eficientes y novedosos en las condiciones actuales, sumamente cambiantes. Esta necesidad es aún más imperiosa cuando cada individuo en una organización tiene su propia forma de abordar los problemas y no siempre tiene todo el conocimiento y experiencia necesarios sobre el negocio.

Contar con una base de ejemplos que ayude en la construcción de software basada en la experiencia de otros especialistas que han abordado proyectos con características similares, mejoraría considerablemente la calidad del producto final y permitiría a los analistas hacer sugerencias que contribuyan a mejorar los procesos productivos y de negocios.

Este trabajo explica dos propuestas que permiten aprovechar el conocimiento acumulado por los desarrolladores de software en nuevos proyectos, utilizando el Razonamiento Basado en Casos (RBC), para obtener un Modelo del Negocio y un Modelo Conceptual partiendo de un conjunto de especificaciones iniciales brindadas por los analistas.

Con esta información, de carácter preliminar, los grupos de desarrollo pueden simplificar de manera notable algunos aspectos de las fases de modelación iniciales que permitan aprovechar el trabajo y las experiencias previas y aceleren por vías probadas el proceso total de elaboración de software.

En este trabajo se describen las características generales del funcionamiento de dos sistemas que se desarrollan en el Centro de Estudios de Ingeniería de Sistemas (CEIS), que aborda estos temas de la Ingeniería de Software, utilizando la técnica de inteligencia artificial de RBC.

1. Definición del Modelo del negocio, una tarea compleja

Cualquier metodología de Análisis y Diseño para el desarrollo de sistemas tiene como punto de partida la captura de requisitos, obtenidos por los analistas en interacción con los usuarios, que mas tarde serán analizados y plasmados en herramientas propias de cada metodología de manera que cubran las expectativas de los usuarios y que se ajusten a las tendencias actuales de desarrollo de aplicaciones.

La obtención de requerimientos es un paso muy importante para el posterior desarrollo de las siguientes etapas [7], pues un error en estas fases iniciales puede dar al traste con un sistema que no cumpla las expectativas de los usuarios y difícilmente aporte valor agregado al negocio para el que debe ser concebido.

No basta con utilizar una metodología adecuada, pues en un proyecto intervienen múltiples especialistas, no todos con la misma experiencia sobre el tema específico que se aborda, es necesario establecer controles de calidad, que contribuyan a detectar fallas o demoras en la ejecución de los proyectos tempranamente, de forma que se puedan tomar las medidas correspondientes.

El éxito de esta etapa depende en gran medida de la experiencia de los analistas y grupos de proyectos, así como de las habilidades desarrolladas por éstos en el desarrollo de sistemas con características similares, mientras más conocimiento sobre este tema tenga el analista más fácil le será proponer una solución adecuada y adaptarse a los cambios del medio, aún cuando este proceso de analizar sus experiencias anteriores es algo que realizará casi de manera intuitiva y hasta sin darse cuenta.

El objetivo de los procesos de desarrollo de software es guiar a los desarrolladores para la obtención de sistemas que se ajusten a las necesidades de los clientes. La eficiencia se mide en términos de costo, calidad y tiempo de desarrollo. El paso entre la determinación de las necesidades de los clientes y la implementación de éstas de manera eficiente no es trivial y está influenciada por múltiples factores [13].

Este es un proceso complejo que depende de la comunicación entre clientes y especialistas y de la relación dentro de los grupos de trabajo, entre otros factores. Un sistema que no responda a las necesidades de los clientes no cumple los requerimientos mínimos de calidad. Por todo esto se hace necesario contar con herramientas que ayuden a los analistas a obtener un Modelo del Negocio y Modelo conceptual que se ajuste a las necesidades de los clientes y que utilice la experiencia acumulada hasta el momento en sistemas con características similares [6]. Una posible solución es utilizar técnicas de inteligencia artificial, que permitan acumular experiencia y contar con ella en el momento adecuado.

En el Centro de Estudios de Ingeniería de Sistemas (CEIS) se está utilizando esta técnica para abordar la captura de requisitos en el desarrollo de proyectos. Se desarrollan dos sistemas que hacen uso de RBC para abordar ésta área de la Ingeniería de Software.

Se trabaja en el desarrollo de un sistema en conjunto con la Universidad Peruana de Ciencias Aplicadas (UPC) que partiendo del listado de clientes y trabajadores del negocio y sus necesidades, propone el listado tentativo de casos de uso y actores del sistema, utilizando la notación UML [4]. Este software contará con una base de ejemplos a partir de los cuales se obtendrán nuevas soluciones. Esta herramienta puede ser utilizada también en metodologías estructuradas para describir el modelo del negocio [7].

El segundo sistema, desarrollado en el CEIS, permitirá obtener un listado preliminar de clases para una metodología orientada a objetos, partiendo del listado de requerimientos candidatos. Este listado preliminar, también llamado Modelo del Dominio [7], se basará en una base de ejemplos que será accedida utilizando RBC [8].

El uso de estas herramientas pueden disminuir el tiempo de modelación del sistema y producir software más eficiente y con mayor calidad, ya que partirá de requerimientos iniciales que han sido obtenidos a partir de la experiencia propia del grupo y del trabajo realizado por otros grupos de proyecto con anterioridad.

Para una empresa productora de software pueden ser de mucha utilidad sistemas con estas características, pues proveerá a la empresa de un banco de ejemplos basados en su propia experiencia de desarrollo y que podrá ser utilizado por nuevos analistas, lo que contribuirá a la formación de éstos en las políticas de la organización.

2. ¿Qué es el Razonamiento Basado en casos?

Esta técnica de inteligencia artificial intenta llegar a la solución de nuevos problemas, de forma similar a como lo hacen los seres humanos [14].

Cuando un individuo se enfrenta a un nuevo problema comienza por buscar en su memoria experiencias anteriores similares a la actual y a partir de ese momento establece semejanzas y diferencias y combina las soluciones dadas con anterioridad para obtener una nueva solución. Este proceso es intuitivo y la persona lo realiza prácticamente sin darse cuenta.

Una vez que la persona tiene situadas un grupo de situaciones anteriores similares a la actual, analiza las variantes que se presentan en la nueva situación y cómo puede dar respuesta a estos cambios.

De manera resumida el proceso ocurre como sigue:

- El individuo buscó en su memoria casos similares.
- Intenta inferir una respuesta a partir del caso mas similar que encontró.

- Tuvo que realizar algunas concesiones y ajustes para adaptar el caso anterior a la situación actual.

Finalmente la solución obtenida no es igual a la anterior, pero cumple dos aspectos muy importantes, el primero da respuesta al nuevo problema y el segundo, ha enriquecido su experiencia anterior con la nueva solución.

El funcionamiento del RBC parte de estos principios y para ello comprende cuatro actividades principales [1, 10]:

- Recuperar los casos más parecidos.
- Reutilizar el o los casos para tratar de resolver el nuevo problema.
- Revisar y adaptar la solución propuesta, en caso de ser necesario.
- Almacenar la nueva solución como parte de un nuevo caso.

Un nuevo problema se compara con los casos almacenados previamente en la base de casos y se recuperan uno o varios casos. Posteriormente se utiliza y evalúa una solución, sugerida por los casos que han sido seleccionados con anterioridad, para ver si se aplica al problema actual [5, 11, 12].

A menos que el caso recuperado sea igual al actual, la solución probablemente tendrá que ser revisada y adaptada, produciéndose un nuevo caso que será almacenado.

La elaboración de un sistema que emplea el RBC presenta dos problemas principales: el primero saber cómo almacenar la experiencia de tal forma que ésta pueda ser recuperada en forma adecuada y el segundo conseguir utilizar la experiencia previa en un problema actual [2, 3].

La forma de representar y almacenar estas experiencias se realiza a través de casos. Un caso mantiene todos los atributos y características relevantes de un evento pasado. Estas características servirán como índices para la recuperación del caso futuro [16].

De acuerdo a la naturaleza del problema tratado se define la representación del caso, es decir, cuáles son los atributos importantes, qué problemas serán tratados, cuál es la solución propuesta, etc. Además es necesario definir el o los mecanismos de recuperación de casos [15].

3. Utilización de la base de ejemplos para la definición del Modelo del Negocio y Modelo del Dominio

A continuación se analiza cómo utilizar la base de ejemplos para obtener el modelo del negocio y el modelo del dominio en las aplicaciones descritas con anterioridad.

Se supone que se desea diseñar un sistema para una Clínica Multidisciplinaria, donde los pacientes pueden reservar citas para una especialidad determinada y a partir de ahí se hace un seguimiento de las consultas realizadas a éste, así como de los análisis indicados por el médico.

En el momento de reservar la cita para un paciente y una especialidad dados se asigna un médico de esa especialidad que tenga espacio en su agenda y se le informa al paciente el día, la hora y el nombre del médico que lo atenderá en su cita. En el momento de la consulta el médico puede decidir indicar al paciente que se realice análisis para chequear los resultados en la siguiente consulta y verificar la evolución de la enfermedad. Para el caso de los análisis de laboratorio hay que registrar la orden de análisis en el momento en que el paciente acude a la clínica a realizárselos y posteriormente se almacenarán los resultados, lo que se hará por mediación de un operador del laboratorio.

Después de todas las entrevistas y estudios necesarios realizados por lo analistas (haciendo una simplificación del negocio) se obtiene un listado de clientes y trabajadores del sistema [5] con lo que ellos necesitan del sistema, como se muestra a continuación:

Cliente / Trabajador	¿Qué necesita del sistema?
Paciente	Reservar cita para especialidad
Operador de laboratorio	Registrar orden de análisis de laboratorio
	Registrar resultados de análisis de laboratorio

Tabla 1. Relación de los clientes del sistema con lo que ellos necesitan de éste.

Para la tabla anterior se podría obtener un Diagrama de Casos de uso [9], que modele los procesos descritos con anterioridad y que tendría una forma similar al mostrado en la figura 1, considerando como actores el contenido de la columna Cliente/Trabajador.

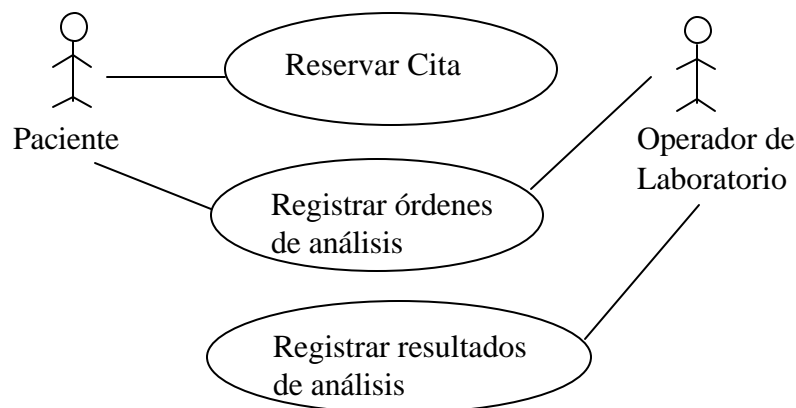


Figura 1. Diagrama de casos de uso para el ejemplo de la clínica.

Nótese que en este diagrama no se incluye ningún proceso que se encargue de crear la Historia Clínica para un nuevo paciente, en este caso el sistema que utiliza RBC puede proponer un caso de

uso que sería una extensión del caso de uso “Reservar Cita”, eso lo puede hacer el sistema porque parte de la experiencia de otros ejemplos en la base de casos que incluyen este proceso de creación de la historia clínica, pero solo en los casos de aquellos pacientes que acuden por primera vez a consultarse.

Si los analistas no cuentan con un sistema de este tipo, llegarán a la solución, pero pueden incluir el nuevo caso de uso en otro momento, cuando debía ser considerado desde el principio. Este es un caso muy simple, pero pudieran ser casos con relativa complejidad y actividades muy específicas con las que los desarrolladores de software no estén familiarizados, en estos casos sería mucho mas evidente la necesidad de una herramienta con tales facilidades.

El sistema, para el ejemplo, también puede proponer, partiendo de los requerimientos candidatos, un listado preliminar de clases, como las que siguen:

- Paciente,
- Historia Clínica,
- Descripción de Análisis de laboratorio.
- Orden de análisis de laboratorio.
- Médico,
- Especialidad,
- Agenda y
- Cita reservada.

Después de la propuesta del sistema, el analista puede hacer ajustes a la solución e incorporar nuevas consideraciones de acuerdo a la situación concreta que está modelando y esa nueva solución se almacenará en la base de ejemplos como un nuevo caso para condiciones diferentes y con el que se podrá construir una nueva solución para otro problema del mismo tipo, es decir la nueva solución será considerada experiencia acumulada a partir de este momento.

La base de ejemplos se enriquecerá y con ella el conocimiento y la experiencia disponible para el desarrollo de nuevos proyectos de software.

3.1. Dos propuestas de sistemas para obtener Modelo del Negocio y Modelo del dominio

En el CEIS se trabaja en el desarrollo de un sistema en conjunto con la Universidad Peruana de Ciencias Aplicadas (UPC) que partiendo del listado de clientes y trabajadores del negocio y sus necesidades, propone el listado tentativo de casos de uso y actores del sistema –Modelo del Negocio- y para ello cada caso almacenado en la Base de Casos contiene la siguiente información:

- Usuarios,
- Trabajadores,

- Necesidades de cada usuario/trabajador,
- Actores resultantes,
- Casos de uso resultantes,
- Relación de cada actor con los casos de uso,
- Requerimientos funcionales del sistema.

Para recuperar los casos se ha considerado una función objetivo en la que se maximizan, entre otros, los valores de los siguientes atributos:

- Cantidad de usuarios/trabajadores iguales entre el caso actual y el almacenado.
- Cantidad de necesidades comunes.

Cada uno de estos atributos tendrá un peso específico, que denotará el nivel de importancia de éste dentro de la función objetivo, en el momento de seleccionar el modelo de negocio que mas se parece a la situación actual, suministrada al sistema por el analista o grupo de proyecto.

El caso seleccionado se le muestra al analista para que realice los ajustes que entienda necesarios y el nuevo caso se almacena, teniendo en cuenta los índices establecidos en la aplicación para el almacenamiento de los modelos de negocio.

El segundo sistema, desarrollado en el CEIS, permitirá obtener un listado preliminar de clases para una metodología orientada a objetos, partiendo del listado de requerimientos candidatos. Este listado preliminar, también llamado Modelo del Dominio, utiliza una base de ejemplos con la siguiente información:

- Área en la que clasifica el sistema,
- Dominio dentro del área,
- Requerimientos funcionales,
- Requerimientos no funcionales,
- Listado preliminar de clases y
- Relación entre el requerimiento y las clases.

En la Base de Casos cada problema será almacenado de acuerdo al Área y el dominio donde clasifique, por ejemplo el Área puede ser Medicina y el Dominio Laboratorio o Consulta Externa.

Para recuperar los casos se ha considerado una función objetivo en la que se maximizan, entre otros, los valores de los siguientes atributos:

- Cantidad de requerimientos iguales entre el caso actual y el almacenado.
- Cantidad de requerimientos significativos iguales.

El usuario del sistema -el analista- en el momento en que define los requerimientos de su problema debe indicar cuáles de ellos deberán ser considerados como significativos. Los requerimientos

significativos son aquellos cuya importancia es medular en el desarrollo del software, en general son requerimientos funcionales.

El caso seleccionado se le muestra al analista para que realice los ajustes que entienda necesarios de la misma forma que en el primer sistema.

Conclusiones

El RBC es una de las áreas de mayor crecimiento en el campo de los sistemas basados en conocimiento. Este paradigma es utilizado frecuentemente por los seres humanos para resolver un sinnúmero de situaciones, siendo ésta una de las principales razones de su aceptación en la comunidad de investigadores de Inteligencia Artificial. La calidad de este tipo de sistemas depende de las experiencias que almacene y de su capacidad para comprender, adaptar, evaluar y reparar nuevos casos.

Este trabajo puede ser un punto de partida para introducir el RBC como ayuda a los desarrolladores de sistemas y contribuir así a lograr software con mayor calidad aprovechando la experiencia acumulada, de manera que se puedan construir soluciones cada vez más complejas y con la rapidez que exige el creciente ritmo de la tecnología de la información.

Este tipo de sistemas contribuyen de manera importante a elevar la calidad del proceso de desarrollo de software y por consiguiente la calidad del producto final, el cumplimiento de los cronogramas previstos y otros aspectos que influyen considerablemente en la eficiencia de las empresas de software en el mundo actual.

De gran utilidad serían estos sistemas en empresas desarrolladoras de software, pues la dotarían de una base de ejemplos con soluciones dadas en la propia empresa que ayudarían a futuros desarrolladores a familiarizarse en poco tiempo con las características y políticas definidas en el trabajo de ésta.

Bibliografía

1. Althoff, K. : “Case-Based Reasoning. Handbook of Software Engineering and Knowledge Engineering.”, kaiserslautern, Alemania. Fraunhofer Institute for Experimental Software Engineering (IESE), 2001.
2. Althoff, K. : “Potential uses of case-based reasoning in the experience-based construction of software systems.”, kaiserslautern, Alemania. Proceedings of the 5th German Workshop in Case-Based Reasoning, Centre for learning Systems and Applications, University of Kaiserslautern, 1997.
3. Bergmann, R. : “Developing Industrial Case Based Reasoning Applications. The INRECA Methodology.”, 1999, Berlín, Alemania. Springer-Verlag.
4. Booch, G. : “The Unified Modeling Language. User Guide”, 1999 Addison Wesley Longman Inc.
5. Cuenca, J. : “Sistemas Inteligentes. Conceptos, Técnicas y Métodos.”, 1998, Publicación de la Facultad de Informática de la Universidad Politécnica de Madrid, Madrid, España.
6. Fowler, M.: “UML Distilled”, segunda edición, 2000 Addison Wesley Longman Inc.
7. Jacobson, I. : “El Proceso Unificado de Desarrollo de Software”, 2000 Addison Wesley Longman Inc.
8. Kolodner, J.: “Case-Based Reasoning”, 1993 Morgan Kaufmann Publishers Inc.
9. Larman, G.: “UML y Patrones”, 1999 Prentice Hall Hispanoamericana, S.A.
10. Manjares, A. : “Razonamiento basado en casos.” Universidad Nacional de Educación a Distancia, Departamento de Inteligencia Artificial, Madrid, España, 2001.
11. Rich, E. : “Inteligencia Artificial”, 1994 McGraw-Hill / Interamericana de España, segunda edición.
12. Riesbeck, Ch.: “Inside Case-based Reasoning”, 1989 by Lawrence Erlbaum Associates Inc.
13. Rumbaugh, J.: “The Unified Modeling Language. Reference Manual”, 1999 Addison Wesley Longman Inc.
14. Schank, R.: “Inside Case-based Explanation”, 1994 by Lawrence Erlbaum Associates Inc.
15. Turban, E: “Decision Support Systems and Intelligent Systems”, 1998 Prentice Hall Inc, Quinta edición, New Jersey, Estados Unidos.
16. Watson, I. : ”Applying Case-Based Reasoning: Techniques for Enterprise Systems.”, 1997 Morgan Kaufmann Publishers, Inc.