

UNC-ANALISTA: HACIA LA CAPTURA DE UN CORPUS DE REQUISITOS A PARTIR DE LA APLICACIÓN DEL EXPERIMENTO MAGO DE OZ

CARLOS MARIO ZAPATA*
CAROLINA PALACIO**
NATALÍ OLAYA***

RESUMEN

La obtención de requisitos es una de las etapas más importantes en el proceso de desarrollo del software. Una buena comprensión de los requisitos puede conducir a mejores productos de software que satisfagan las necesidades de los interesados. Sin embargo, el proceso de captura de requisitos se torna difícil para el analista debido, en gran parte, al carácter presencial que tienen las reuniones que se realizan con tal fin y a la dificultad que presentan algunas personas para expresar sus ideas de forma clara. En este artículo se presenta UNC-Analista, una propuesta de experimento Mago de Oz enfocado al diseño de un sistema de diálogo controlado que posibilite la labor del analista durante el proceso de obtención de requisitos. Con este sistema será posible capturar un corpus de requisitos, que servirá como base para la construcción futura de un sistema automático para la obtención de requisitos.

PALABRAS CLAVE: requisitos; software; corpus; experimento Mago de Oz; sistemas de diálogo.

* Ph. D. (C) en Ingeniería. Grupo en Ingeniería de Software. Profesor Asistente de la Escuela de Sistemas. Facultad de Minas. Universidad Nacional de Colombia. cmzapata@unalmed.edu.co

** Estudiante de Ingeniería de Sistemas. Grupo en Ingeniería de Software. Escuela de Sistemas. Facultad de Minas. Universidad Nacional de Colombia. cpalaci@unalmed.edu.co

*** Estudiante de Ingeniería de Sistemas. Grupo en Ingeniería de Software. Escuela de Sistemas. Facultad de Minas. Universidad Nacional de Colombia. nolaya@unalmed.edu.co

Este artículo se realizó en el marco de los proyectos de investigación Construcción automática de esquemas conceptuales a partir de lenguaje natural, financiado por la DIME, y Definición de un esquema preconceptual para la obtención automática de esquemas conceptuales de UML, financiado por DINAIN y administrado por la DIME.

ABSTRACT

Requirements elicitation is one of the most important phases in software development process. A good requirements understanding can lead to better software products, achieving satisfaction of stakeholder needs. However, requirements-capture process is sometimes difficult for analysts, because of the face-to-face character of the meetings required for it and because of difficulties of people for expressing clearly their ideas. In this paper we present UNC-Analista, a proposal for Wizard-of-Oz experiment focused on the design of a dialogue-controlled system for helping the analyst labor in the requirements elicitation process. With this system will be possible to capture a requirements corpus, for leading a future development of an automatic requirements elicitation system.

KEY WORDS: requirements; software; corpus; Wizard-of-Oz experiment; dialogue systems.

1. INTRODUCCIÓN

La obtención de requisitos es una de las etapas fundamentales del proceso de análisis, diseño y construcción de una pieza de software, pues contribuye a la realización de una primera descripción del problema que permite establecer las motivaciones para su solución mediante sistemas informáticos (Zapata y Arango, 2004).

Una adecuada comprensión de los requisitos puede conducir al desarrollo de mejores aplicaciones de software que cumplan con las necesidades y expectativas de los interesados. Sin embargo, la identificación de los requisitos es un gran problema con el cual los analistas han tenido que enfrentarse durante el proceso de análisis. Esta situación se debe en gran parte a que la información se extrae de personas (interesados) que, por lo general, no tienen claras sus verdaderas necesidades y expectativas sobre el software que precisan para resolver los problemas de su organización. Por consiguiente, la información obtenida por el analista puede variar dependiendo de la persona a la que se esté consultando, así como de la selección apropiada de las preguntas para la captura de requisitos por parte del analista. Este carácter subjetivo del proceso de obtención depende de la habilidad del analista para identificar las necesidades de los interesados.

Para llevar a cabo este procedimiento, en la ingeniería de requisitos se han propuesto varias técnicas que guían al analista en el proceso de establecer una comunicación con el interesado y su equipo de trabajo (Sommerville, 2001; Pressman, 2005; Goguen y Linde, 1993; Raghavan *et al.*, 1994 y Leffingwell y Widrig, 1999). Estas técnicas implican una alta inversión de tiempo de ambas partes (analista e interesado), debido al carácter presencial que tienen la mayoría de las reuniones que se realizan durante este proceso y por ser diálogos persona-persona que poseen todas las imprecisiones que el lenguaje natural presenta. El análisis de la información resultante lo realiza en estas técnicas únicamente el analista, con el fin de traducirla a los diferentes diagramas que representan dicha información; estos diagramas difícilmente pueden ser validados por los interesados, por su desconocimiento del lenguaje técnico en que están elaborados. Este problema podría aliviarse parcialmente si se automatizara la actividad comunicativa analista-interesado.

En los últimos años, las investigaciones orientadas al procesamiento del lenguaje natural por medio de la lingüística computacional se han incrementado considerablemente (Mitkov, 2003), con el objetivo de incluir el lenguaje natural en la interacción hombre-máquina. Esto permite al ser humano sentirse más cómodo en el momento de establecer



una comunicación con un sistema informático. Para lograr esto se han propuesto trabajos en el diseño, desarrollo y evaluación de sistemas que incluyen interacciones hombre-máquina, en los cuales los experimentos Mago de Oz (Fraser y Gilbert, 1991) han sido de gran importancia. Sin embargo, pocas han sido las aplicaciones de software encaminadas a la obtención de requisitos que aplican este experimento; algunas de ellas se pueden encontrar en Salber y Coutaz (1993), Coutaz *et al.* (1994), Sommerville y Sawyer (1997) y Mørch *et al.* (2005).

En este artículo se presenta UNC-Analista, una propuesta para la captura de un corpus de requisitos (un conjunto de especificaciones en un lenguaje controlado generado a partir de diálogos analista-interesado) aplicando para ello el experimento Mago de Oz. Esta propuesta se hace con el fin de recabar información que pueda ser importante en la definición de las especificaciones de un sistema automático para la captura de requisitos de cualquier aplicación informática.

Este artículo está organizado de la siguiente manera: en la sección 2 se describen algunas de las técnicas más tradicionales empleadas por los analistas para capturar los requisitos de los interesados. En la sección 3 se describe el experimento Mago de Oz, el cual ha sido empleado en ingeniería de software para la creación y evaluación de prototipos. En la sección 4 se propone una adaptación del experimento Mago de Oz para la captura de un corpus de requisitos de una pieza de software. En la sección 5 se presenta un caso de estudio. En la sección 6 se concluye y se presenta el trabajo futuro que se deriva de esta propuesta.

2. TÉCNICAS TRADICIONALES PARA LA CAPTURA DE REQUISITOS

La obtención de requisitos es el proceso mediante el cual los interesados en un sistema de software descubren, revelan, articulan y entienden

sus requisitos (Raghavan *et al.*, 1994). Una correcta captura de los requisitos del interesado facilitará la incorporación de cambios posteriores en el sistema; por lo tanto, se hace necesario que los analistas empleen técnicas que les permitan establecer una buena comunicación con los interesados y su equipo de trabajo.

A continuación, se hace un compendio de las técnicas tradicionales para la captura de requisitos, a partir de los trabajos de Sommerville (2001), Pressman (2005), Goguen y Linde (1993), Raghavan *et al.* (1994) y Leffingwell y Widrig (1999).

2.1 Entrevistas

Es la más tradicional de las técnicas de obtención y consiste en reuniones analista-interesado en las cuales se suceden preguntas y respuestas para extraer el dominio de la aplicación (Goguen y Linde, 1993). En Pressman (2005) se presentan conjuntos de preguntas que se pueden utilizar en el desarrollo de esta técnica, que tiene una alta participación del analista y se realiza en conjunto con otras técnicas.

2.2 Técnicas para facilitar las especificaciones de una aplicación (TFEA)

Estas son una variación de las entrevistas buscando identificar el problema, proponer elementos de solución, negociar diferentes enfoques y especificar un conjunto preliminar de requisitos de la solución (Pressman, 2005). A pesar de ir un paso más allá de las entrevistas convencionales, precisan aún de una alta participación del analista.

2.3 Despliegue de la función de calidad (DFC)

Esta técnica incluye entrevistas y documentación de la organización con las cuales se construye la tabla de opinión del interesado. Esta tabla se analiza

con diagramas, matrices y métodos de evaluación para extraer los requisitos esperados e intentar obtener requisitos innovadores (Pressman, 2005). También esta técnica requiere una alta interacción con el analista.

2.4 Tormenta de ideas (brainstorming)

Es una técnica de reuniones en grupo cuyo objetivo es la generación de ideas en un ambiente libre de críticas o juicios. Puede ayudar a generar una gran variedad de vistas del problema y a formularlo de diferentes formas, sobre todo al comienzo del proceso de captura, cuando los requisitos son todavía muy difusos (Raghavan et al., 1994). Sin embargo, también se requiere participación intensiva del analista.

2.5 Juego de roles

En su forma más simple, consiste en que el desarrollador, el analista y cada uno de los miembros del equipo de desarrollo del software toman el lugar del interesado y ejecutan la actividad de trabajo que éste desempeña. Ellos experimentan las inexactitudes y problemas ligados con el sistema que se está especificando. Se busca suministrarle al analista una perspectiva nueva del problema que le permita la obtención de los requisitos del sistema por construir (Raghavan et al., 1994). Esta técnica también presenta alta participación de los involucrados.

2.6 Introspección

Esta técnica recomienda que el analista se ponga en el lugar del interesado y trate de imaginar cómo diseñaría éste la aplicación de software. Basado en estas suposiciones, el analista entrega recomendaciones al interesado sobre la funcionalidad que debería tener dicha aplicación (Goguen y Linde, 1993). El problema radica en que un analista no es un tipo normal de interesado, pues posee un conocimiento

técnico más elevado; por ello, es posible que entre las recomendaciones haya cosas que el interesado aún no necesita o que incluso no sabe que necesitará en un futuro. En este caso, el discurso se refiere más a la solución que al dominio del problema.

2.7 Casos de uso o escenarios

Son descripciones que incluyen actores, eventos, operaciones y objetivos de esas operaciones, generalmente ligados con el funcionamiento de una solución informática (Leffingwell y Widrig, 1999; Raghavan et al., 1994; Pressman, 2005; Sommerville 2001). Los escenarios como técnica de obtención poseen dos limitaciones: exigen una alta participación del interesado en su elaboración y necesitan que él realice una concepción completa de la solución informática, que sólo sería posible al final del proceso de obtención de requisitos.

2.8 Desarrollo conjunto de aplicaciones (joint application development –JAD–)

Se realiza en un conjunto de reuniones que cuentan con recursos especiales (diagramas, transparencias, multimedios, herramientas CASE, etc.), trabajando directamente sobre los documentos de requisitos por generar (Raghavan et al., 1994; Goguen y Linde, 1993). Además de exigir una alta intervención del analista, se precisa que el interesado conozca elementos técnicos como diagramas, generalmente reservados a los analistas.

2.9 Tablero de historias (storyboarding)

Son simulaciones que utiliza el analista para tratar de establecer con el interesado cómo será la aplicación de software que necesita para solucionar sus problemas. Se suelen utilizar videos, animaciones, programas de presentaciones o incluso lenguajes visuales de programación (Raghavan et al., 1994).



Los tableros de historias también se suelen referir a la solución informática que en etapas iniciales del desarrollo aún no existe.

2.10 Prototipado

Es una técnica similar a la anterior (Sommerville, 2001; Raghavan *et al.*, 1994) y comparte con ella sus mismas limitaciones. En este caso se desarrolla la solución informática con una funcionalidad restringida para que sea el interesado el que la valide.

2.11 Dificultades que presentan las técnicas tradicionales de obtención de requisitos

Las técnicas presentadas muestran que la obtención de requisitos ha sido tratada ampliamente. Aun así, estas técnicas presentan algunas dificultades, como las que se enuncian a continuación:

- Relación persona-persona con alta participación del analista.
- Conocimientos técnicos por parte del interesado (que comúnmente no los posee).
- Revisión de la documentación (que generalmente se realiza de manera manual).
- Alto conocimiento de la solución (que en las etapas iniciales del desarrollo, cuando se realiza el proceso de obtención de requisitos, aún no se conoce con certeza).

El hecho de que el diálogo se entable entre personas implica que se desarrolla en un lenguaje natural, con todas las imprecisiones y malas interpretaciones que pueden tener lugar en este tipo de diálogos. Con base en las técnicas presentadas es posible un manejo más cuidadoso de la información asociada con la aplicación de software, lo cual paulatinamente puede conducir dicha aplicación hasta el código fuente. Los interesados en general no tienen

Tabla 1. Técnicas tradicionales de obtención de requisitos y sus problemas.

Técnica	Problema			
	Requiere alta participación del analista	Requiere conocimientos técnicos en el interesado	Incluye revisión manual de documentos	Se refiere a la solución
Entrevistas	X			
TFEA	X			X
DFC	X		X	
Tormenta de ideas	X			
Juego de roles	X			
Introspección	X			X
Casos de uso	X	X		X
JAD	X		X	X
Tablero de historias	X			X
Prototipado	X			X

el conocimiento técnico suficiente para acompañar al equipo de desarrollo hasta esa instancia.

Ahora, si bien las técnicas presentadas facilitan la interacción analista-interesado, aún se requiere mucho tiempo para llegar a especificar claramente lo que el interesado espera de la aplicación de software. Técnicas como el prototipado, por ejemplo, son mucho más visuales, pero también necesitan tiempo y esfuerzo que podrían no verse recompensados si no se aclara rápidamente la idea que tiene el interesado.

En la tabla 1 se presenta un compendio de los problemas asociados con las técnicas tradicionales para la obtención de requisitos.

2.12 Otros elementos teóricos por considerar

Sommerville y Sawyer (1997) citan una técnica no convencional para la elaboración de prototipos de aplicaciones de software empleando el experimento Mago de Oz. En esta técnica, el mago (un analista que se oculta tras una interfaz gráfica de usuario) “simula” la interacción con una aplicación informática que aún no existe realmente, con el fin de que el interesado pueda observar y corregir el funcionamiento de dicha aplicación. Si una aplicación se fuera a construir para tratar de solucionar algunos de los problemas anotados para las técnicas de obtención de requisitos, el experimento Mago de Oz podría contribuir a que el prototipo de esa aplicación permitiera estudiar el comportamiento de los interesados ante una aplicación real de obtención de requisitos. Además, la aplicación del experimento podría conducir a la recopilación de un corpus (un conjunto o cuerpo especial de información) de requisitos que se empleen en procesos reales de captura de requisitos.

El uso de lenguajes controlados puede permitir el almacenamiento en el corpus de los diferentes requisitos capturados mediante el diálogo con el interesado. El UN-Lencep (Zapata *et al.*, 2006) puede

ser el lenguaje controlado que permita almacenar los requisitos capturados mediante la aplicación de diálogo que se propone en este artículo, que en adelante se denominará UNC-Analista. Los discursos expresados en UN-Lencep tienen, además, como una característica importante el hecho de que sus sentencias permiten la generación automática de algunos de los diagramas de UML (Zapata *et al.*, 2006b).

Un discurso en UN-Lencep tiene la apariencia de un conjunto de frases del tipo “sujeto-verbo-complemento”, en el cual el verbo puede ser de tipo “estructural” (específicamente los verbos ser y tener) o de tipo “dinámico” (con verbos de acciones o actividades que se realizan en el mundo); ejemplos de estas frases son “el propietario es una persona” y “la secretaria asigna la cita”, respectivamente. Además, existen otros tipos de frases como las implicaciones “CUANDO sujeto-verbo dinámico-objeto, ENTONCES sujeto-verbo dinámico-objeto” (por ejemplo, “cuando la secretaria asigna la cita, entonces el propietario cumple la cita) y los condicionales “SI {CONDICIÓN}, ENTONCES sujeto-verbo dinámico-objeto”, donde {CONDICIÓN} es una expresión que incluye uno o más entes del mundo separados mediante operadores de comparación (por ejemplo, “SI mascota.estado = ‘enfermo’, ENTONCES veterinario receta medicamento”).

Como punto de partida para entender cómo el experimento Mago de Oz puede contribuir en la captura de un corpus de requisitos en UN-Lencep, en la siguiente sección se presentan los lineamientos básicos de dichos experimentos.

3. EL EXPERIMENTO MAGO DE OZ

Además de los problemas anotados en la sección anterior, en la obtención de requisitos la interacción interesado-analista tiene una dificultad adicional: las ambigüedades propias del lenguaje natural del interesado y su falta de formalismo. Si se tratara de automatizar el proceso, algunas de



esas limitaciones se podrían superar. Eso haría que la obtención de requisitos se realizara de manera similar a como se usan los sistemas de diálogo automático propuestos por la disciplina Interacción Hombre-Máquina (Human-Computer Interaction -HCI-) (Allen, 1995).

En un sistema de diálogo automático se establece una comunicación hombre-máquina. El diálogo se produce en un lenguaje más sencillo y preciso que el utilizado en diálogos persona-persona. Por esta razón, durante el diseño de tales sistemas se hace necesaria la utilización de interacciones ficticias entre el interesado y el sistema, con las cuales se logra orientar a las personas que participan en la evaluación de un prototipo como potenciales interesadas en un acto comunicativo semejante al de la aplicación real del sistema de diálogo. Para este caso en particular, estas interacciones se recogen en un corpus construido mediante un experimento denominado “Mago de Oz” (Fraser y Gilbert, 1991), en el que una persona simula el comportamiento total o parcial del sistema futuro, con el fin de recoger muestras de intervenciones del interesado con el sistema antes de disponer del sistema mismo. La labor del operador humano, el mago, consiste en simular el sistema automático sin que el interesado se percate de ello, para entregarle respuesta a sus consultas. La interacción se produce en condiciones similares a las del sistema en operación.

La estrategia del mago puede variar dependiendo del dominio donde se aplique. Por lo general, consiste en ir supervisando la información generada durante el proceso de diálogo. Para ello, el mago debe ejecutar algunas de las siguientes acciones: completar información, confirmar datos, comunicar que la información no es válida o no ha sido entendida, consultar la base de datos, entre otras. La importancia de la estrategia radica en sistematizar el comportamiento y las respuestas del mago. Así, a lo largo de la interacción, el interesado suministra información necesaria para poder llevar a cabo la consulta. Si dicha información es insuficiente o no

es muy precisa, el mago le solicita al interesado más información en un nuevo turno de diálogo y así sucesivamente, hasta que el mago considere que la información disponible es suficiente para realizar una consulta a la base de datos y posteriormente mostrar la respuesta de ella.

En Fraser y Gilbert (1991) se presentan algunos elementos y variables que se deben tomar en cuenta en la realización de experimentos Mago de Oz. De ellos se resaltan los escenarios y los objetivos.

Los escenarios describen las condiciones y circunstancias concretas para que el interesado sepa cómo actuar ante el sistema al momento de dialogar con él. En la definición del escenario se incluye un objetivo (la información que debe obtener el informante), una situación que motiva el interés en la información, así como un esquema que condiciona la petición.

Mediante los escenarios se evalúa el funcionamiento de un prototipo del sistema; simulan una situación que puede ocurrir en el funcionamiento real de un sistema de diálogo y se diseñan para proporcionar a las personas que participarán en las pruebas para evaluar el prototipo del sistema una indicación o unas pautas sobre cuál ha de ser su papel. Los escenarios se elaboran teniendo en cuenta la relación entre los participantes en el acto comunicativo: el interesado que solicita una determinada información y el mago que se la facilita. Suelen definirse tres tipos de escenarios: cerrados, semicerrados y abiertos. En los escenarios cerrados se presenta una situación totalmente dirigida, donde el interesado debe pedir una determinada información a partir de las instrucciones que se le proponen en el escenario, ya que debe ceñirse a la situación propuesta en éste. En los escenarios semicerrados el interesado conoce parte de la información que necesita para realizar determinado tipo de consulta, teniendo una mayor libertad para solicitar la información. Por último, en los escenarios abiertos el interesado desconoce la mayoría de los detalles de la consulta que ha de realizar.

Naturalmente, la persona que actúa de mago necesita una etapa previa de entrenamiento y un conocimiento muy detallado de los escenarios que se hayan elaborado. En este sentido, los escenarios abiertos son los más difíciles de abordar, ya que el interesado puede preguntar cualquier detalle sobre la información que quiere solicitar.

El experimento Mago de Oz se suele emplear en la creación y evaluación de prototipos con los siguientes objetivos (Llisterri *et al.*, 2003):

- Probar la eficacia del sistema. Se pueden modificar las estrategias de diálogo de acuerdo con los resultados de iteraciones sucesivas del experimento.
- Analizar la manera como interactúan las personas y la máquina, con el fin de extraer información lingüística necesaria para el sistema real en el momento de interpretar los mensajes del interesado. Las características de las conversaciones naturales no se conservan cuando la comunicación se lleva a cabo con un computador. Por tal motivo, se debe establecer una comparación entre el corpus persona-persona y el corpus persona-sistema informático. Para esto deben tenerse en cuenta la duración total de la interacción, el tiempo de espera, el número de turnos utilizados desde que el interesado solicita la información hasta que se le facilita y el número de llamadas que se desvían porque el mago no es capaz de contestarlas.
- Suministrar entrenamiento para el módulo de reconocimiento, ya que en la mayoría de las aplicaciones se necesita el reconocimiento del habla espontánea. El reconocedor puede mejorar si se incorpora información sobre ciertos fenómenos lingüísticos propios de este estilo de habla: repeticiones, inserciones que truncan el discurso, relajaciones en la articulación de los sonidos, alargamientos de vocales, ruidos del interesado, vocalizaciones, etc.

En la sección 2.11 se mencionaron como problemas de las técnicas tradicionales de obtención

de requisitos el carácter natural del diálogo que se entabla entre analista e interesado (que requiere mucho tiempo de ambos para precisar las necesidades de los interesados), la capacitación técnica que se exige al interesado, la validación manual del proceso y la necesidad de concebir una solución en etapas preliminares del desarrollo. Aunque las investigaciones orientadas al procesamiento del lenguaje natural con la lingüística computacional han tenido un gran auge en las últimas décadas, pocas han sido las aplicaciones realizadas con el experimento Mago de Oz para tratar de solucionar algunos de esos problemas. Entre ellas se pueden contar:

- Salber y Coutaz (1993) y Coutaz *et al.* (1994) describen una plataforma Mago de Oz genérica denominada NEIMO, que se emplea para la evaluación de la usabilidad de una pieza de software.
- En Sommerville y Sawyer (1997) se discute una técnica de elaboración de prototipos para cualquier tipo de sistema, empleando para ello el experimento Mago de Oz. Además, Mørch *et al.* (2005) usan el experimento Mago de Oz para simular con interesados humanos el comportamiento de agentes de software.

En la siguiente sección se presenta la propuesta de aplicación del experimento Mago de Oz a la captura de un corpus de requisitos del software, con el fin de superar las limitaciones anotadas para las técnicas tradicionales de obtención de requisitos.

4. APLICACIÓN DEL EXPERIMENTO MAGO DE OZ EN LA CAPTURA DE UN CORPUS DE REQUISITOS CON EL PROGRAMA UNC-ANALISTA

En los trabajos mencionados en la sección 3 no se tomaron en cuenta las posibilidades de



generalización que posee el trabajo de obtención de requisitos de diferentes aplicaciones de software, las cuales, si bien poseen dominios diferentes y características propias, también poseen similitudes en cuanto al proceso de captura, pues en todos los casos se trata de conocer la organización y el problema para presentar un conjunto de soluciones y modelarlas. Con base en esto, se presenta una propuesta para realizar un experimento Mago de Oz que tome esas similitudes y las plasme en un conjunto de interfaces; éstas constituirán el escenario para que un analista realice el trabajo de mago e interactúe con los interesados a través de ellas y no directamente, con el fin de evitar las ambigüedades del lenguaje natural. De esta manera se recopilará un corpus de requisitos expresados en el lenguaje controlado UN-Lencep; el punto de partida serán las interacciones interesado-analista que servirán en el futuro para predecir las respuestas que debería entregar un sistema de diálogo automático. La idea del sistema automático será generar una relación directa entre el interesado y la máquina, estableciendo una interacción semejante a la de una situación comunicativa real. El sistema contará con un módulo capaz de realizar las preguntas apropiadas para obtener los requisitos necesarios en la elaboración de determinado software.

Como producto final se deberá entregar una especie de modelo verbal, pero en un lenguaje controlado (en esencia, más “preciso”) en el cual se encuentren todos los requisitos establecidos por el interesado. Esto implica un entrenamiento muy sofisticado para dicho módulo de preguntas, de tal forma que éste pueda llevar a cabo la actividad comunicativa elaborada hasta ahora sólo por el analista. Para tal entrenamiento es indispensable la existencia de ejemplos de diálogo de la tarea sobre la cual se desea realizar el sistema. Así, la recolección de diferentes muestras de entrevistas realizadas por analistas a los interesados durante el proceso de obtención de requisitos se hace primordial para la abstracción del prototipo de las preguntas que el sistema, haciendo el papel del analista, debe formular.

UNC-Analista es un sistema que se centra en la adaptación y posterior utilización del experimento Mago de Oz para la captura de un corpus de requisitos. Ese corpus se puede utilizar posteriormente como principal fuente de información para la alimentación del módulo automático de preguntas de un sistema de diálogo en lenguaje controlado orientado a la obtención de requisitos de los interesados.

En este experimento el analista se hace pasar por el sistema; de esta forma entabla una comunicación por red, basada en una serie de preguntas, con un interesado; establece los requisitos para realizar una aplicación determinada, mientras éste se encuentra convencido de que lo hace con un sistema automático. El hecho de que el interesado crea que la comunicación la está entablando con una máquina, lo restringe a usar un lenguaje más preciso (comprensible por el sistema) en sus respuestas que el que usaría si se tratase de una persona. Este es un factor importante en el momento de llevar a cabo la futura automatización.

Para la aplicación de este experimento primero se diseñó un sistema de diálogo simple en el que el mago y el interesado pueden establecer una comunicación. Este sistema cuenta con diversas interfaces en las cuales el mago tiene la opción de construir en el esquema que él elija las preguntas que considere adecuadas, ya sean abiertas o cerradas (de selección múltiple), así como los mensajes de alerta cuando encuentra inconsistencias en la información brindada por el interesado; tal esquema se establece con el fin de hacer más real el hecho de que el interesado está interactuando con una máquina y no con una persona.

El mago debe ser un analista con una amplia experiencia en ingeniería de requisitos, que haya trabajado en la obtención de requisitos de diferentes proyectos. Además, debe estar previamente entrenado para realizar de forma rápida el diseño de las preguntas en el sistema, que de manera casi instantánea deben llegar al interesado.

La aplicación del experimento deberá estar delimitada por una serie de escenarios que tienen que ver con la información que se desea obtener, restringiendo el tipo de preguntas que el mago está obligado a formular (orientadas al fin de la organización, al conocimiento de los problemas que se deseen tratar, al objetivo del sistema que se desea construir, al conocimiento de los principales actores participantes, etc.). Los escenarios sirven para reglar las conversaciones analista-interesado y así descubrir patrones de uso de las preguntas utilizadas y garantizar completitud en los requisitos. Tales patrones permitirán el reconocimiento de las preguntas esenciales durante la comunicación analista-interesado, de las cuales se puedan extraer de forma exitosa los requisitos más importantes para un sistema automático.

Al final, la información obtenida se guardará en archivos diferentes que conformarán el corpus de requisitos. Uno de esos archivos almacenará el discurso resultante en UN-Lencep y el tipo de escenario elegido (que se determina con la primera pregunta sobre el dominio de la aplicación), mientras que la forma de realización de las preguntas y las correspondientes respuestas se guardarán en otro archivo.

La descripción de las interfaces gráficas de usuario (GUI por su sigla en inglés) del sistema es la que sigue.

EL MAGO

Menú principal. En esta interfaz el mago elige el escenario en el que desea trabajar, así como la estructura que considere más adecuada para la pregunta que va a formular (figura 1):

Preguntas cerradas:

- Selección múltiple con botones de radio (sólo se puede escoger una opción)

- Selección múltiple con cajas de cotejo (se pueden escoger varias opciones)
- Falso o verdadero



Figura 1. Menú principal de UNC-Analista para el mago

Preguntas abiertas:

- Tipo ensayo (el interesado puede extenderse en su respuesta)
- Tipo línea (la respuesta debe ser concisa)

En esta interfaz el mago ve las respuestas dadas por el interesado.

Diseño de preguntas. Al elegir el tipo de pregunta que desea formular, aparece una interfaz donde se tiene una plantilla editable con un diseño ya establecido sobre la cual el mago puede formular y posteriormente enviar la pregunta al interesado (figura 2).



Botones de Radio

Formule aquí su pregunta de Selección Múltiple utilizando botones de radio (Sólo se puede escoger una opción)

Opción1

Opción2

Opción3

Opción4

Cancelar Enviar Pregunta

a)

Cajas de Cotejo

Formule aquí su pregunta de Selección Múltiple utilizando cajas de cotejo (Se puede escoger más de una opción)

Opción1

Opción2

Opción3

Opción4

Cancelar Enviar Pregunta

b)

Falso o Verdadero

Formule aquí su pregunta de Falso o Verdadero utilizando botones de radio (Sólo se puede escoger una opción)

SI

NO

Cancelar Enviar Pregunta

c)

Tipo Discurso

Formule aquí su pregunta de Contexto Libre Tipo Discurso (El usuario puede extenderse en su respuesta)

Cancelar Enviar Pregunta

d)

Tipo Línea

Formule aquí su pregunta de Contexto Libre Tipo Línea (El usuario debe responder concretamente)

Cancelar Enviar Pregunta

e)

Figura 2. Interfaces gráficas de usuario correspondientes al menú de diseño de preguntas

EL INTERESADO

Interfaz de comunicación. En esta única interfaz el interesado establece el diálogo con el supuesto sistema (el mago); puede observar las preguntas que haga el “sistema” y puede digitar o seleccionar (en caso de que la pregunta sea cerrada) la respuesta correspondiente (figura 3).

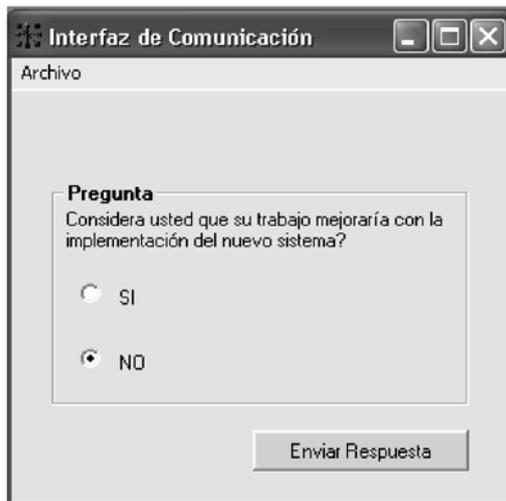


Figura 3. Interfaz de un UNC-Analista correspondiente al interesado

Con el fin de suministrar tiempo al mago para hacer la siguiente pregunta, el sistema puede mostrar interfaces de advertencia como la que se muestra en la figura 4.

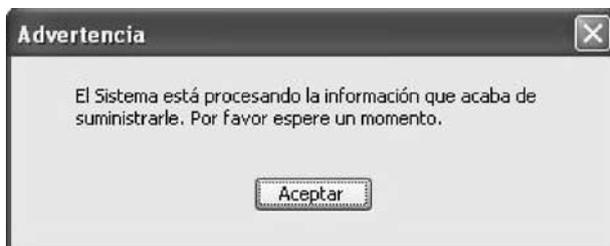


Figura 4. Ventana de advertencia en UNC-Analista

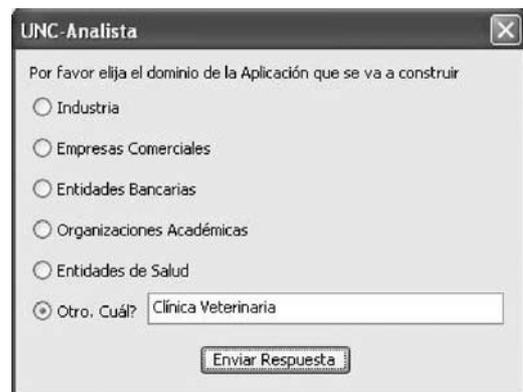
Las preguntas del analista deberán guardar coherencia con las respuestas suministradas por el interesado, pues, de lo contrario, el interesado podría sospechar que no es un sistema automático el que está haciendo las preguntas.

Como resultado de la interacción en UNC-Analista, se obtiene un conjunto de frases en el lenguaje controlado UN-Lencep (Zapata *et al.*, 2006), que después se pueden emplear en la generación de esquemas conceptuales de UML según el proceso descrito en Zapata *et al.* (2006b).

5. CASO DE ESTUDIO

En esta sección se simula en UNC-Analista parte del diálogo que se genera entre el mago y un interesado que desea que se le desarrolle una aplicación que le facilite los procesos de registro, asignación de citas, diagnóstico y manejo de medicamentos en una pequeña clínica veterinaria. En la figura 5 se pueden apreciar las interfaces gráficas de usuario resultantes de ese diálogo en UNC-Analista. El proceso es el siguiente:

- Tanto el mago como el interesado ingresan al sistema UNC-Analista simultáneamente pero en diferentes terminales. El mago ingresa en la interfaz correspondiente a la figura 1 y el interesado ingresa en una interfaz de advertencia similar a la de la figura 4, pero cuyo mensaje es “Ingreso Satisfactorio” y allí permanecerá hasta que el mago inicie la interacción.



a)



UNC -Analista [X]

Cuáles de los siguientes actores están presentes en la Clínica Veterinaria?

- Veterinario
- Secretaria
- Asistente
- Propietario
- Mascota
- Otro. Cuál?

Enviar Respuesta

b)

UNC -Analista [X]

Enuncie una función de la Secretaria

Enviar Respuesta

c)

UNC -Analista [X]

Qué pasa antes de que la Secretaria asigne una cita?

- Alguien pide la cita
- Alguien trae una mascota

Enviar Respuesta

d)

UNC -Analista [X]

Usualmente quién pide la cita?

- Propietario
- Secretaria
- Veterinario

Enviar Respuesta

e)

UNC -Analista [X]

Es cierto que una mascota pertenece a un propietario?

- Sí
- No

Enviar Respuesta

f)

UNC -Analista [X]

Cuáles de las siguientes características poseen las mascotas?

- Identificación
- Nombre
- Apellido
- Otro. Cuál?

Enviar Respuesta

g)

UNC -Analista [X]

La Historia Clínica posee alguna característica?

- Sí
- No

Enviar Respuesta

h)

Figura 5. Sucesión de mensajes en UNC-Analista

- El mago presiona el botón “Selección múltiple con botones de radio” e ingresa en la interfaz definida en la figura 2a), y digita la información que se presenta en la figura 5a). Una vez ingresa la información, presiona el botón “Enviar Pregunta” y la interfaz correspondiente se muestra al interesado.
- El interesado presiona el botón “Enviar Respuesta” y le aparece la ventana de advertencia de la figura 4.
- El mago recibe la respuesta y presiona el botón “Selección múltiple con cajas de cotejo” para activar la interfaz que corresponde a la figura 2b). Una vez en esa interfaz, digita la información pertinente para obtener la apariencia correspondiente a la interfaz de la figura 5b) y luego presiona el botón “Enviar Pregunta” para hacerla llegar al interesado.

El proceso continúa hasta obtener todas las interfaces de la figura 5 y hasta que el mago obtenga todas las respuestas a sus preguntas. Nótese que las preguntas deben guardar coherencia en la medida en que se van ejecutando, puesto que un error de este estilo en la elaboración de las preguntas puede hacer que el interesado descubra que se trata de un analista humano y decida no seguir colaborando con el experimento. Es importante resaltar que, a medida que transcurren las interacciones, el mago comienza a disponer de un menú contextual, tal como se muestra en la figura 6. Ese menú puede ser invocado en cualquier momento por el analista y en él se incorporan los términos que el interesado ha ido ratificando con sus respuestas. Así, por ejemplo, cuando el interesado oprime “Enviar Respuesta” al finalizar la interacción d) en la figura 5, el menú contextual posee los siguientes términos: Clínica Veterinaria, Veterinario, Secretaria, Propietario, Mascota, citas y asignar. Este menú es importante para garantizar la coherencia del discurso por parte del analista, pues una vez se incorpore un término por primera vez, el menú contextual permitirá escribirlo siempre de la misma manera.



Figura 6. Menú contextual invocado por el mago

Una vez se cumplan las interacciones con las interfaces de la figura 5, UNC-Analista deberá formar un discurso en UN-Lencep, que se almacenará en el corpus de requisitos. Para el ejemplo de la figura 5 ese discurso tiene la siguiente redacción:

- Cuando el propietario pide una cita, entonces la secretaria asigna la cita.
- Una mascota pertenece a un propietario.
- Una mascota posee identificación, nombre e historia clínica.

6. CONCLUSIONES Y TRABAJO FUTURO

El experimento del Mago de Oz, cuando se adapta a la captura de requisitos del software, permite la obtención de requisitos más precisos que los que resultan durante las entrevistas en las técnicas tradicionales; esto se logra debido a que el interesado, al creer que está etablando una comunicación con “el supuesto sistema”, se ve obligado a restringir



más el lenguaje que usa para responder las preguntas formuladas por el mago.

Los escenarios establecidos en el experimento establecen restricciones y reglas en las conversaciones analista-interesado y facilitan de esta manera el análisis posterior al experimento. Se busca descubrir los patrones de uso de las preguntas y garantizar la completitud en los requisitos.

La manera en que se realiza la interacción entre el mago y el interesado, por medio de los tipos de preguntas que se seleccionan, su diligenciamiento, envío y posterior respuesta, no sólo sirve como base para la construcción de un sistema automático de obtención de requisitos, sino que además el conocimiento de esa interacción puede ser de gran utilidad para los analistas a la hora de formular preguntas en una entrevista dirigida en el marco de una técnica de obtención de requisitos convencional.

En este experimento se reconoce el diálogo analista-interesado como un acto de comunicación técnica que se puede automatizar con una herramienta de captura de diálogo como UNC-Analista. Además, se pueden tener algunas respuestas predefinidas que reemplacen al mago en el sistema completamente automático; la información correspondiente para esas preguntas predefinidas podría surgir a partir de conjuntos de preguntas y respuestas analista-interesado recolectadas por la primera etapa de UNC-Analista, que emplea el experimento Mago de Oz, para la generación de un corpus de requisitos.

El alto grado de capacitación que debe poseer el mago se constituye en la principal amenaza a la validez del experimento, dado que es él quien permite que el interesado crea que está interactuando con una aplicación automática y no con un analista humano.

Después de la captura del corpus de requisitos con el experimento del Mago de Oz y del posterior análisis de los resultados obtenidos, el trabajo futuro consiste en el diseño y construcción del sistema de diálogo para la obtención automática de los requi-

sitos, cuya base sea el patrón de uso de preguntas, establecido a partir del experimento. Tales patrones de uso se pueden determinar por medio de métodos estadísticos aplicados al corpus (producto del experimento) que indiquen la frecuencia o el grado en el cual cierta pregunta puede ser indispensable para encaminar a una información relevante en la especificación de requisitos de determinado software. De este modo, se puede lograr que el módulo de preguntas del sistema sea capaz de seleccionar, sin intervención alguna del analista, las preguntas indicadas para ser formuladas al interesado durante el proceso de obtención de requisitos, y poder así llevar a cabo de forma automática tal proceso.

Una segunda línea de trabajo futuro tiene que ver con la incorporación del UNC-Analista en el proceso de otras herramientas que también se construyen en la actualidad, como el UNC-Diagramador, una herramienta CASE basada en UN-Lencep y esquemas preconceptuales. De esta manera se podrían obtener incluso diagramas de UML tomando como punto de partida las preguntas y respuestas que se realizan en UNC-Analista mediante la interacción analista-interesado.

REFERENCIAS

- Allen, J. 1995. Natural language understanding. The Benjamin/Cummings, Redwood City,
- Coutaz, J.; Salber, D.; Balbo, S. and Jambon, F. 1994. Supporting usability evaluation through software engineering tools. Proceedings of the Workshop on Models for Developing High Impact Formative Evaluation Methods at the ACM Conference on Human Factors in Computing Systems CHI'94, Boston.
- Fraser, N. and Gilbert, G. 1991. Simulating speech systems. Computer Speech and Language, 5:81-99.
- Goguen, J. and Linde, Ch. 1993. Techniques for requirements elicitation. Proceedings of Requirements Engineering '93, 152-164.
- Leffingwell, D. and Widrig, D. 1999. Managing software requirements: a unified approach. Addison Wesley, Reading.
- Llisterri, J.; Carbó, C.; Machuca, M. J.; De la Mota, C.; Riera, M. y Ríos, A. 2003. El papel de la fonética en

- el desarrollo de las tecnologías del habla. Memorias de las VII Jornadas de Lingüística. Cádiz, Servicio de Publicaciones de la Universidad de Cádiz.
- Mørch, A.; Dolonen, J. and Nævdal, J. 2005. An evolutionary approach to prototyping pedagogical agents: from simulation to integrated systems. *Journal of Network and Computer Applications*, 29:177-199.
- Mitkov, R., 2003. *The Oxford handbook of computational linguistics*. Oxford University Press, New York.
- Pressman, R. 2005. *Software engineering: a practitioner's approach*, 6th ed. McGraw-Hill, New York.
- Raghavan, S.; Zelesnik, G. and Ford, G. 1994. *Lecture notes on requirements elicitation*. Educational Materials, CMU/SEI-94-EM-10, Software Engineering Institute, Carnegie Mellon University.
- Salber, D. and Coutaz, J. 1993. Applying the Wizard of Oz technique to the study of multimodal systems. Len J. Bass, Juri Gornostaev and Claus Unger (eds), *Human-Computer Interaction Selected Papers*, Berlin, Springer Verlag, *Lecture Notes in Computer Science*, 753:219-230.
- Sommerville, I. 2001. *Software engineering*, 6th ed. Addison-Wesley Longman, Boston.
- Sommerville, I. and Sawyer, P. 1997. *Requirements engineering: a good practice guide*. John Wiley and Sons, Chicester.
- Zapata, C. M. y Arango, F. 2004. Alineación entre metas organizacionales y elicitación de requisitos del software. *DYNA*, 143:101-110.
- Zapata, C. M.; Gelbukh, A. y Arango, F. 2006. UN-Lencep: Obtención automática de diagramas UML a partir de un lenguaje controlado. Memorias del 3er Taller en Tecnologías del Lenguaje Humano del Encuentro Nacional de Computación, San Luis Potosí, septiembre de 2006.
- Zapata, C. M.; Gelbukh, A. and Arango, F. 2006b. Pre-conceptual schema: a UML isomorphism for automatically obtaining UML conceptual schemas. *Research in Computing Science: Advances in Computer Science and Engineering*, 19, 2006:3-13.