

## Revisión con base cognitiva de un Proceso de Requisitos

Graciela D.S. Hadad<sup>1,2</sup>, Jorge H. Doorn<sup>1,2</sup>, María C. Elizalde<sup>1</sup>

<sup>1</sup> Escuela de Informática, Universidad Nacional del Oeste

<sup>2</sup> DIIT, Universidad Nacional de la Matanza

ghadad@uno.edu.ar, jdoorn@exa.unicen.edu.ar, melizalde@uno.edu.ar

### RESUMEN

En todas las etapas del desarrollo de software, están presentes variadas tareas cognitivas, inmersas en el proceso mismo. Estas tareas cognitivas han sido poco estudiadas, aún cuando afectan el desempeño de las personas intervinientes en el proyecto. Si bien todas las actividades del desarrollo de un sistema de software involucran tareas cognitivas, las correspondientes a las primeras etapas tienen una mayor presencia de las mismas. Es decir, durante el proceso de requisitos, la colaboración, la comunicación y la comprensión del contexto donde se desempeñará el sistema son partes inherentes del mismo. Es por ello que se considera que la psicología cognitiva, y particularmente la ergonomía cognitiva, pueden ayudar a lograr una mejor comprensión de las actividades humanas que se llevan a cabo en esta etapa del desarrollo de software. En el presente proyecto se planifica estudiar, revisar, evaluar y eventualmente proponer modificaciones a las heurísticas de un proceso específico de requisitos, focalizando en aspectos cognitivos de los involucrados, de manera tal de mejorar el desarrollo de las actividades del proceso y, por ende, mejorar la calidad del producto final.

**Palabras clave:** Ingeniería de Requisitos, Proceso de Requisitos, Ergonomía Cognitiva, Elicitación, Lenguaje Natural.

### CONTEXTO

La línea de investigación que se presenta es parte de los proyectos de investigación “Tratamiento de los Factores Situacionales y la Completitud en la Ingeniería de Requisitos” de la Universidad Nacional del Oeste (UNO) y “Reflexividad como herramienta en la

Ingeniería de Requisitos” de la Universidad Nacional de La Matanza (UNLaM).

### 1. INTRODUCCIÓN

En muy diversas actividades de la Ingeniería de Software, en general, y de la Ingeniería de Requisitos, en particular, es altamente recomendable utilizar heurísticas, especialmente en aquellas relacionadas con modelos construidos manualmente o en forma parcialmente automatizada. Frecuentemente estas heurísticas son adaptadas, no siempre en forma planificada, a circunstancias particulares del contexto de cada proyecto de desarrollo, tanto por razones relacionadas con el proyecto como por características de los usuarios de las mismas [1]. Sin embargo, rara vez se las cuestiona de raíz, procurando concentrar la atención en el sustento conceptual de las mismas.

Una consecuencia inmediata de lo anterior es preguntar cómo es que han sido construidas esas heurísticas y, por supuesto, la pregunta más general acerca de cómo debería ser construida cualquier heurística.

La respuesta a la segunda pregunta tiene la apariencia de ser sumamente simple y justamente esa apariencia es el origen de su dificultad intrínseca. En virtud del alto contenido de información de la mayoría de los modelos utilizados en las diferentes actividades de la Ingeniería de Requisitos, resulta ser que las heurísticas deberían guiar al ser humano en el desarrollo de las actividades cognitivas involucradas.

La mera introducción de la palabra *cognitiva* hace evidente lo erróneo de considerar simple la construcción de una heurística en ésta y en muchas otras áreas de la Ingeniería de Software.

Abordar la planificación y ordenamiento

de cualquier actividad intelectual, aún aquellas con componentes rutinarios, es difícil ya que se deben considerar las fortalezas y debilidades del ser humano en relación con dichas actividades.

El estudio del desempeño del ser humano en entornos deportivos, laborales y situaciones de todo tipo tiene larga data, aunque en la mayoría de los casos orientados a problemáticas de actividades motrices y uso de herramientas. Estos estudios han dado lugar a la disciplina denominada *ergonomía* [2, 3], también conocida como *factores humanos*.

Posteriormente, los estudios relacionados con los factores humanos han ido adquiriendo un perfil cada vez más relacionado con aspectos sensoriales y cognitivos [4], habiéndose centrado principalmente en las interacciones con sistemas complejos [5]. La introducción de estos enfoques en los sistemas de procesamiento de datos se produjo en el área de visualización y diseño de interfaces [4, 6]. Aún hoy en día, la mayoría de estos estudios siguen estando en las cercanías de aspectos sensoriales y no verdaderamente cognitivos.

Todo lo anterior hace evidente que este es un terreno prácticamente inexplorado cuyas características son simultáneamente muy promisorias y desafiantes. Las promesas residen en el hecho que al poner en el centro del problema las peculiaridades de la cognición humana se avizoran importantes mejoras en la forma de abordar la elicitación de conocimiento y en su posterior elaboración y registro, mientras que los desafíos residen en que los aportes de las ciencias cognitivas tienen aún pocos resultados de importancia pragmática.

Como consecuencia, el presente proyecto de investigación tiene esencialmente características exploratorias [7, 8]. En otras palabras, se trata de elaborar hipótesis que permitan guiar futuros proyectos en los que las mismas sean confirmadas, mejoradas o refutadas. Usualmente se suele considerar que la definición de las heurísticas es una tarea sencilla y casi irrelevante. Definitivamente este no es el caso. En diversos proyectos

previos, se ha comprobado lo irreductible que son algunos de las debilidades conocidas en el proceso de requisitos, especialmente aquellas vinculadas con los problemas de completitud [9, 10, 11, 12].

Las cuatro fuentes básicas de hipótesis [7] son: i) marcos conceptuales o teóricos, ii) referencias bibliográficas, iii) investigaciones previas, y iv) intuiciones, sospechas y experiencias de la vida diaria. Se estima que en el presente proyecto se deberá recurrir a todas ellas.

La necesidad de tener en cuenta las limitaciones del ser humano en cuanto a su capacidad de comprensión y análisis de problemas de toda naturaleza ha estado presente en la informática desde etapas muy tempranas y sigue estando. Posiblemente el ejemplo más relevante en este sentido sea la fuerte recomendación de guiarse por enfoques top-down en diversas áreas y en utilizar el método de los refinamientos sucesivos en particular [13]. Pese a lo valioso de estas contribuciones es relevante destacar que en el artículo original de Niklaus Wirth no hay mención alguna a la necesidad de concentrar la atención del programador en unos pocos aspectos en cada momento del desarrollo. Tampoco se indica que se esté enfrentando la creación de un programa para atender una necesidad perfectamente comprendida y especificada.

Al considerar el traslado de estos enfoques al área de la Ingeniería de Requisitos, en su muy valiosa contribución [14], Michael Jackson enfáticamente aconseja no usar enfoques top-down en virtud que hacen necesario fragmentar el sistema a estudiar en el peor momento posible, que es cuando menos se lo conoce. Es decir, en el momento en que existe mayor probabilidad de cometer errores por la falta de comprensión acabada del problema. Nuevamente se encuentra aquí un sustento, algo más explícito, de atender las necesidades cognitivas de las personas intervinientes.

Otros ejemplos similares son el uso de abstracción mediante lenguajes de modelado del dominio [15] y la descomposición mediante diagramas de flujos de datos [16].

Se pueden agregar otros muchos y muy variados ejemplos, en los que se pueden observar innovaciones relacionadas con aspectos cognitivos. Sin embargo, en casi todos los casos, las cuestiones cognitivas tienen un rol de telón de fondo. Es decir, son poco analizadas y su influencia sobre las innovaciones introducidas no ha sido efectivamente probada.

## 2. LÍNEAS DE INVESTIGACIÓN Y DESARROLLO

Una somera revisión de la literatura en lo que se refiere a las heurísticas que soportan fuertemente la gran mayoría de los métodos de la Ingeniería de Software muestra que las mismas están básicamente orientadas a describir el contenido de los modelos que a ayudar a elaborar conceptualmente ese contenido.

Si se analiza el Proceso Unificado en su flujo de trabajo Requisitos, éste propone la captura de requisitos centrándose en el modelado del dominio, del negocio y de los casos de uso, con heurísticas que atienden puramente a construirlos [17]. Por ejemplo, el modelo de dominio se describe mediante diagramas de clase UML cumpliendo con: “las clases del dominio aparecen como i) objetos del negocio que representan cosas que se manipulan en el negocio, ii) objetos del mundo real y conceptos que el sistema usará, y iii) sucesos que ocurrirán o han ocurrido en el entorno de trabajo” [17, pág.113]. Para la descripción de casos de uso, el Proceso Unificado menciona: “i) Debe definir el estado inicial como precondition, ii) Debe definir los posibles estados finales, iii) La interacción del sistema con los actores y qué cambios producen”, entre otros puntos [17, pág. 149]. En general, se observa que el Proceso Unificado no propone ninguna ayuda acerca de cómo seleccionar elementos del dominio, del negocio o del sistema, tales como clases, actores e interacciones relevantes, o como descartar lo superfluo.

En general, la literatura referida al Desarrollo Dirigido por Modelos (MDD) [18] considera que los dos primeros modelos CIM (Computational Independent Model) y PIM

(Platform Independent Model) en la cadena de transformación no pueden ser obtenidos en forma automatizada a partir de reglas de construcción (caso de CIM) ni de reglas de derivación del modelo previo (caso de PIM), sino que en ambos casos deben obtenerse a través de heurísticas, las que atienden aspectos netamente relacionados con el meta-modelo que los soporta sin guiar en cómo obtener el contenido. Es decir, en este paradigma aún no se han podido establecer reglas que automaticen la construcción de estos modelos ni que colaboren con la determinación de qué tipo de información del contexto de aplicación es necesario capturar, siendo considerada hoy en día una actividad “artesanal” [15]. Como declara Booch et al. [19] en su manifiesto de MDD solo se automatiza la transformación de modelos que no dependan de la “ingenuidad humana”.

Numerosos autores reportan que incrementa la probabilidad de éxito de un proyecto de software el estudio del dominio del problema y luego que los requisitos de ese sistema se plasmen en modelos creados en Lenguaje Natural [20, 21]. Las mejoras registradas en la calidad de los requisitos tienen su fundamento en que estas representaciones, tales como glosarios, casos de uso y escenarios, promueven la comunicación entre todos los involucrados en un proyecto de desarrollo de software y facilitan la validación de los requisitos elaborados. Una revisión hecha por Rolland et al. [22] muestra que, de doce enfoques propuestos en la literatura en el ámbito de la Ingeniería de Requisitos, todos ellos usan una notación de texto para describir escenarios, que en algunos casos se combinan con otros medios, tales como gráficos o imágenes. En un estudio posterior sobre la práctica en Ingeniería de Requisitos [23], se concluyó que el 51% de las organizaciones (sobre un total de 194) usa representaciones informales (por ejemplo, el lenguaje natural) y el 27% semi-formales, quedando los modelos formales relegados a un uso de apenas el 7% de las organizaciones.

En proyectos de investigación previos, se ha desarrollado una estrategia de IR basada en

modelos en lenguaje natural [24]. Aún cuando esta estrategia puede considerarse suficientemente madura, es deseable que la misma pueda ser mejorada en varios aspectos. Este proyecto está enmarcado en esa estrategia.

La estrategia procura alcanzar una profunda comprensión del dominio de la aplicación para definir una solución óptima a través de un sistema de software. Es decir, la estrategia presenta dos grandes etapas bien distinguibles: una de aprendizaje y la otra de definición. Cuando hay un conocimiento previo del dominio de la aplicación la primera fase se convierte en un proceso confirmatorio.

Los modelos que se utilizan en esta estrategia [24] son:

- un modelo léxico, LEL (Léxico Extendido del Lenguaje), el cual describe el vocabulario utilizado en el dominio de la aplicación,
- un modelo organizacional que describe los procesos actuales del negocio utilizando, denominado Escenarios Actuales,
- un modelo organizacional que describe los procesos del negocio proyectados en base al sistema de software a desarrollar, denominado Escenarios Futuros.

La mayoría de las heurísticas de esta estrategia de IR adolecen de las mismas debilidades que se han mencionado para el Proceso Unificado y para MDD, en el sentido que están fuertemente orientadas al contenido de los modelos que se construyen, con pocas pautas sobre cómo capturar y analizar la información a ser registrada en ellos.

### 3. RESULTADOS OBTENIDOS/ESPERADOS

En experiencias preliminares [25, 26] se han aplicado nociones de ergonomía cognitiva a la construcción del modelo Léxico Extendido del Lenguaje, poniendo especial énfasis en el uso de una heurística radicalmente diferente de la usual y orientada a facilitar la percepción del significado de los términos por parte del ingeniero de requisitos. Los resultados de este estudio fueron notoriamente prometedores, ya que se logró

una cantidad de términos notoriamente superior a la que se había obtenido anteriormente.

En la Tabla 1 se resumen los datos recogidos en una de estas experiencias [27]. Las omisiones indicadas en la misma están calculadas comparando el LEL obtenido con una versión anterior del mismo LEL creado por otros autores, habiendo sido ambas experiencias controladas en el sentido que ambos trabajos fueron realizados con el mismo material y con absoluto desconocimiento del otro trabajo y de la intención de hacer una comparación posterior.

**Tabla 1.** Omisiones en la versión de un LEL respecto a otra versión creada aplicando nociones cognitivas

Tipo de Término	Omisiones
Sujetos	17 %
Objetos	27 %
Verbos	62 %
Estados	44 %
Total	39 %

Es notable el déficit en la detección de los verbos con las heurísticas preexistentes. También se destaca que siempre se tuvo la suposición, poco fundada, que la mayor debilidad de la actividad de construcción del LEL residía en la identificación apropiada de los estados, hecho que parece no ser cierto.

Estos resultados son meramente indicativos, por dos razones: i) no tienen significación estadística, y ii) no se han establecido hipótesis que vinculen las visiones cognitivas con la heurística utilizada.

Tal como se mencionó en la introducción, desde el punto de vista epistemológico, el presente proyecto se puede catalogar como una investigación esencialmente exploratoria, en la que se aspira a: i) detectar las principales dificultades cognitivas que enfrenta el ingeniero de requisitos al elicitar y modelar el conocimiento que se adquiere del universo del discurso y al concebir el conjunto de servicios que deberá prestar el sistema de software a ser desarrollado, y ii) proponer nuevas heurísticas que no sólo describan los modelos a ser confeccionados sino que efectivamente contribuyan a facilitar la elaboración de la información que se registre en los mismos.

Obviamente, se espera que estos resultados puedan luego ser contrastados en investigaciones confirmatorias.

#### 4. FORMACIÓN DE RECURSOS HUMANOS

En el proyecto de la UNLaM participan tres investigadores, dos de ellos en formación. En el proyecto de la UNO participan en este tema otros dos investigadores, uno de ellos en formación.

La línea de investigación presentada aquí es parte directa de las tesis de maestría de la Ing. Renata Guatelli y de las tesis de doctorado de la Mg. Gladys Kaplan y la Ing. Andrea Vera, y colabora con la tesis de doctorado de la Lic. María Pepe.

#### 5. BIBLIOGRAFÍA

- [1] Hadad, G.D.S., Doorn, J.H., Ledesma, V.A. (2018) Dynamic Situational Adaptation of a Requirements Engineering Process, en: Encyclopedia of Information Science and Technology, Fourth Edition, IGI Global, cap. 646, pp. 7422-7434.
- [2] Murrell, K. (1965) Ergonomics Man in His Working Environment, Springer, pp 103-124.
- [3] Yong Zhou, Y., (1974) Applied Ergonomics Handbook, Elsevier, pp. 1-7.
- [4] van der Veer, G. (2008) Cognitive Ergonomics in Interface Design - Discussion of a Moving Science, Journal of Universal Computer Science, Vol. 1, N° 16, pp. 2614-2629.
- [5] Gersh, J.R., McKneely, J.A., Remington, R.W. (2005) Cognitive engineering: Understanding human interaction with complex systems, Johns Hopkins APL Technical Digest, Vol. 26, N° 4, pp. 377-382.
- [6] Kylie, M., Gomes, K.M., Riggs, S.L. (2017) Analyzing Visual Search Techniques using Eye Tracking for a Computerized Provider Order Entry (CPOE) Task, Human Factors and Ergonomics Society Annual Meeting, pp. 691-695.
- [7] Hernández Sampieri, R., Fernández Collado, C., Baptista Lucio, M.P. (2010) Metodología de la Investigación, 5ta. ed., Mc Graw Hill.
- [8] Popper, K. (1963) Conjectures and Refutations: The Growth of Scientific Knowledge, Routledge.
- [9] Doorn, J.H., Ridao, M.N. (2003) Completitud de Glosarios: Un estudio experimental, VI Workshop on Requirements Engineering, pp. 317-328.
- [10] Ridao, M.N., Doorn, J.H. (2006) Estimación de Completitud en Modelos de Requisitos Basados en Lenguaje Natural, XI Workshop on Requirements Engineering, pp. 146-157.
- [11] Doorn, J.H., Ridao, M.N. (2008) Completeness Concerns in Requirement Engineering, en: Encyclopedia of Information Science and Technology, Second Edition, IGI Global.
- [12] Hadad, G.D., Litvak, C. S., Doorn, J.H., Ridao, M. (2015) Dealing with Completeness in Requirements Engineering, en: Encyclopedia of Information Science and Technology, Third Edition, IGI Global.
- [13] Wirth, N. (1971) Program Development by Stepwise Refinement. Communications of the ACM, Vol. 14, N° 4, pp. 221-227.
- [14] Jackson, M., A. (1995) Software Requirements & Specifications: a lexicon of practice, principles and prejudices, Addison Wesley.
- [15] Pons, C., Giardini, R., Pérez, G., (2010) Desarrollo de Software Dirigido por Modelos: Conceptos teóricos y su aplicación práctica, Editorial de la Universidad Nacional de La Plata / McGraw-Hill Educación.
- [16] Jourdon, E. (1989) Modern Structured Analysis, Englewood Cliffs, NJ, Yourdon Press/Prentice Hall.
- [17] Jacobson, I., Booch, G., Rumbaugh, J., (2000) El Proceso Unificado de Desarrollo de Software, Pearson Educación, Madrid.
- [18] Karow, M., Gehlert, A. (2006) On the Transition from Computation Independent to Platform Independent Models, 12th American Conference on Information Systems, Mexico.
- [19] Booch, G., Brown, A., Iyengar, S., Rumbaugh, J., Selic, B. (2004) An MDA Manifesto, en Frankel, D. and Parodi J. (eds) The MDA Journal: Model Driven Architecture Straight from the Masters.
- [20] Ryan, K. (1993) The Role of Natural Language in Requirements Engineering, IEEE International Symposium on Requirements Engineering, San Diego, pp. 240-242.
- [21] Loucopoulos, P., Karakostas, V. (1995) System Requirements Engineering, McGraw-Hill, Londres.
- [22] Rolland, C., Souveyet, C, Ben Achour, C. (1989) Guiding Goal Modeling Using Scenarios, IEEE TSE, Vol. 24, N° 12, pp. 1055-1071.
- [23] Neill, C.J., Laplante, P.A. (2003) Requirements Engineering: The State of the Practice, IEEE Software, Vol.20, N° 6, pp.40-45.
- [24] Leite, J., Doorn, J., Kaplan, G., Hadad, G., Ridao, M. (2004) Defining System Context using Scenarios. Perspectives on Software Requirements, Kluwer Academic Publishers, cap.8, pp.169-199.
- [25] Litvak, C.S., Hadad, G.D.S., Doorn J.H. (2017) Nominalizations in Requirements Engineering Natural Language Models, en: Encyclopedia of Information Science and Technology, Fourth Edition, IGI Global.
- [26] Litvak, C.S., Hadad, G.D.S., Doorn, J.H. (2016) Procesamiento de Lenguaje Natural para Estudiar Completitud de Requisitos. Workshop de Investigadores en Ciencias de la Computación, pp. 498-502.
- [27] Litvak, C.S., Doorn J.H. (2016) Aporte del Procesamiento de Lenguaje Natural a la Ingeniería de Requisitos, 4to Congreso Nacional de Ingeniería Informática / Sistemas de Información, pp.463-467.