

Estimación de Completitud en Modelos de Requisitos Basados en Lenguaje Natural

Marcela Ridao
INTIA
Facultad de Ciencias Exactas
UNICEN, Tandil, Argentina
mridao@exa.unicen.edu.ar

Jorge H. Doorn
INTIA, Facultad de Ciencias Exactas
UNICEN, Tandil, Argentina
Depto. de Ingeniería, UNLaM, Argentina
jdoorn@exa.unicen.edu.ar

Abstract

La Ingeniería de Requisitos provee mecanismos para elicitar y especificar requisitos, procurando maximizar calidad y completitud. Sin embargo, obtener un modelo de requisitos completo es una meta inalcanzable, y meramente estimar el grado de completitud alcanzado es muy difícil. El mismo problema se presenta en diversas áreas del proceso de desarrollo de software, como las verificaciones de documentos de diseño y la prueba de programas, entre otras. La introducción de técnicas de predicción basadas en modelos estadísticos en el campo de la Ingeniería de Software lleva ya varios años, con muy buenos resultados. En el presente artículo se estudia experimentalmente el uso de los datos de captura y recaptura en el proceso de la Ingeniería de Requisitos, en particular en la construcción de Escenarios a partir del Léxico Extendido del Lenguaje. Se analiza también el impacto que la completitud del LEL tiene sobre la completitud del modelo de escenarios correspondiente.

1. Introducción

El objetivo principal de la Ingeniería de Requisitos es determinar lo que se pretende de un artefacto de software. Para ello, se proponen métodos, técnicas y herramientas cuyo fin es sistematizar el proceso de definición de requisitos [1] [2] [3] [4]. Además, puesto que tanto los clientes-usuarios como los ingenieros de requisitos deberían participar y colaborar en ese proceso, los métodos propuestos por la Ingeniería de Requisitos procuran ganar compromiso y facilitar la comunicación entre ellos. Si bien estos métodos intentan asegurar un máximo de completitud y calidad en los modelos resultantes, pretender obtener un documento de requisitos completo es una utopía. El ingeniero de requisitos se enfrenta con un Universo de Discurso (UdeD) que nunca parece terminar de conocer [5].

El problema de la completitud es una amenaza constante a la calidad de los requisitos. Un conjunto de requisitos incompleto podría producir un artefacto de software que no cumple con todas las expectativas del cliente-usuario.

El problema de la completitud en la Ingeniería de Requisitos, así como en la Ingeniería de Software tiene características comunes con los problemas que se presentan en otras áreas del conocimiento, como la Biología, la Medicina y la Ecología. Precisamente, fue en estas áreas donde se utilizó por primera vez el método de Captura-Recaptura para estimar el tamaño de una población cerrada de animales [6] [7]. Este método fue utilizado posteriormente en el campo de la Ingeniería de Software para estimar el número total de errores presentes en un artefacto de software, a partir de los resultados de un proceso de inspección sobre dicho artefacto. Este método puede aplicarse a todos los artefactos de software susceptibles de ser inspeccionados o probados, por ejemplo código, documentos de requisitos o de diseño [8] [9] [10] [11] [12].

En el proceso de obtención de los requisitos de un sistema de software, la validación constituye una de las tareas más complejas, ya que en muchos casos, se requiere que los clientes-usuarios posean conocimientos y habilidades específicas para poder comprender los modelos resultantes de la elicitación y especificación de los requisitos. El uso de representaciones basadas en lenguaje natural mejora la comunicación del ingeniero de requisitos con los clientes-usuarios. En particular, el uso de los modelos de LEL y Escenarios para la elicitación de requisitos y su utilización a través de todo el proceso de desarrollo de software facilita la validación con el cliente/usuario. El propósito principal del léxico es capturar el vocabulario de la aplicación y su semántica, mientras que los escenarios son usados para entender la aplicación y su funcionalidad [13].

Si bien el uso de este tipo de representaciones ayuda a la validación de los requisitos, esta actividad

habitualmente se reduce a la exhibición, total o parcial, de dichos modelos al cliente-usuario.

Usualmente, la validación ha sido vista como un conjunto de actividades donde es poco importante la presencia de procedimientos y métodos automáticos o semi-automáticos. En este trabajo, se pretende aplicar técnicas estadísticas para estudiar la completitud de modelos de requisitos basados en lenguaje natural. Esta idea introduce un grado mayor de formalidad que el habitual en la actividad de validación.

Se analizarán estadísticamente los modelos correspondientes a una estrategia que en primer término produce el léxico del UdeD y luego los escenarios para obtener conocimiento acerca del ambiente en el cual el software será usado. Se utilizan escenarios actuales para modelar el UdeD como es percibido y escenarios futuros para proyectar las situaciones que surgirán como consecuencia de la evolución. A partir de estos modelos, se obtienen los requisitos del sistema de software.

Se ha efectuado un estudio previo [5], aplicando un método cuantitativo para estimar la cantidad máxima de términos que podría contener el LEL de un UdeD determinado utilizando una estrategia de captura/recaptura. Este estudio, sin embargo, sólo atacó la primera parte del problema. En este artículo se aplicarán métodos cuantitativos a la segunda etapa del proceso de obtención de requisitos. Se estimará la cantidad máxima de escenarios de un UdeD utilizando también una estrategia de captura/recaptura.

El proceso de construcción del Léxico Extendido del Lenguaje y luego de los Escenarios es sólo el comienzo de una secuencia de actividades cuyo propósito consiste en elicitación primero conocimiento del UdeD, y luego el conjunto de los requisitos del sistema de software a ser desarrollado.

Este artículo pretende ser una primera contribución al objetivo más amplio de analizar el impacto que la completitud de cada uno de los modelos del proceso tiene sobre la completitud de los requisitos obtenidos al final del mismo. En una estrategia constituida por una sucesión encadenada de modelos, donde existe un claro vínculo entre una etapa y la siguiente, es importante predecir cómo impactará un LEL más o menos completo sobre el modelo de Escenarios, para luego analizar cómo impactará la completitud de este modelo en los siguientes, hasta llegar a los requisitos.

El artículo está organizado de la siguiente manera: La sección 2 describe las técnicas de predicción basadas en la captura y recaptura, especialmente el método Detection Profile Method (DPM), introducido por Wohlin y Runeson [14]; la sección 3 presenta un breve resumen del proceso de obtención de requisitos en el que se basó el experimento; la sección 4 presenta el caso de estudio utilizado, los resultados obtenidos en estudios anteriores, los datos obtenidos al estimar el número total de escenarios, y analiza el impacto que la completitud del

LEL tiene sobre los escenarios. Finalmente, se presentan las conclusiones obtenidas y se detallan los caminos que se pretenden seguir en un futuro cercano.

2. Estimaciones cuantitativas en universos cerrados

Los métodos de estimación de poblaciones cerradas de cualquier naturaleza, animales, errores en el software y en este caso términos de LEL o escenarios se basan en la idea de la captura y recaptura. Inicialmente, cuando se aplicó a poblaciones de animales, se capturó una cierta cantidad de animales en una primera cacería [6]. Obviamente, se utilizaron cacerías incruentas con el objeto de minimizar la perturbación a la población animal. A cada animal capturado se lo marcó y liberó sin ningún otro tipo de acción sobre el mismo. Transcurrido un cierto período de tiempo se realizó otra cacería, registrando la cantidad de animales que ya habían sido capturados anteriormente. En los casos en que se planeaba efectuar una o varias cacerías más, se marcó de manera diferente los animales capturados una sola vez de aquellos capturados dos veces. La hipótesis básica de esta estrategia consiste en que si el nivel de recaptura es alto, es decir si la mayoría de los animales es capturada en ambas oportunidades, entonces la población es reducida, mientras que si el nivel de recaptura es bajo, entonces la población es amplia. En estos usos iniciales se hipotetizó adicionalmente que todos los animales tenían la misma probabilidad de ser capturados y que todos los capturadores eran igualmente eficaces. Sin embargo, aun en las aplicaciones iniciales, ambas hipótesis fueron puestas bajo la lupa, dudándose de su veracidad.

Al aplicar este tipo de estrategias a las inspecciones de software, documentos de requisitos, documentos de diseño, etc. [8] [9] [10] [11] [12], el rol de los animales salvajes fue tomado por los defectos del objeto inspeccionado y el rol de los capturadores le correspondió a los inspectores. En este caso, las dudas acerca de la veracidad de la afirmación que todos los defectos tienen una dificultad similar de ser descubiertos y que los inspectores son igualmente idóneos es también cuestionada, aunque con más énfasis aún. Actualmente, se utilizan 4 modelos basados en diferentes hipótesis de detección:

- M0: Todos los defectos tienen la misma probabilidad de ser detectados y todos los inspectores tienen la misma habilidad para detectar defectos.
- Mt: Todos los defectos tienen la misma probabilidad de ser detectados pero la habilidad para detectar errores de los inspectores puede ser diferente.
- Mh: Los defectos pueden tener diferente probabilidad de ser detectados pero

todos los inspectores tienen la misma habilidad para detectar defectos.

Mth: Los defectos pueden tener diferentes probabilidades de detección y los revisores pueden tener diferentes habilidades para detectar errores.

Se ha comprobado que los modelos Mh y Mth son los más adecuados para el caso de defectos detectados en inspecciones [15].

Wohlin y Runeson [14] propusieron métodos alternativos que no se basan en las hipótesis de los modelos M0, Mh, Mt y Mth, sino que simplemente se basan en ajustar la curva que representa los datos de la cantidad de inspectores que encontró cada defecto. El fundamento de este método reside en que un ajuste de los datos experimentales con una curva teórica simple, como puede ser una exponencial decreciente, empuja dentro de los parámetros de la curva las variaciones de probabilidades del modelo Mth sin necesidad de suponer ninguna distribución de probabilidades en particular.

Considerando:

$$m_x = A \times e^{-bx}, \quad (1)$$

donde m_x indica el número total de inspectores que encuentran un defecto x , b describe el decrecimiento de la función exponencial, y A es una constante. Mediante técnicas de regresión es posible estimar los parámetros A y b . Una vez estimados dichos parámetros, el número total de defectos es determinado por el mayor valor de x para el cual la ecuación (1) produce un resultado mayor o igual que 0.5.

Se cuenta el número de inspectores que ha detectado cada defecto, y luego se ordenan estos datos en forma decreciente. Se grafica entonces el número obtenido para cada defecto. Ajustando los datos representados con una curva exponencial decreciente, es posible predecir el número total de defectos [14].

La estimación del número total de defectos mediante la propuesta de Wohlin y Runeson es mucho más sencilla que los métodos basados en hipótesis de distribuciones de probabilidades para los defectos y/o los inspectores. Además, se han efectuado estudios en los cuales se ha comprobado que la calidad de la estimación de DPM es igual o superior a la lograda con otros estimadores [9] [16].

Al aplicar este método a la estimación del número total de términos del LEL [5] y del número total de escenarios para un UdeD, se consideró el LEL y el modelo de escenarios obtenidos por nueve grupos diferentes, graficando el número de grupos que descubrió cada símbolo o cada escenario respectivamente. En la sección 4 se analizan estos datos.

3. Proceso de obtención de Requisitos usando LEL y Escenarios

Es de amplia difusión que para definir los requisitos de un sistema de software, se requiere un profundo conocimiento del UdeD. Esto incluye no sólo la visión actual del mismo, sino también los planes para el futuro. A menos que el ingeniero de requisitos tenga un conocimiento previo del dominio, puede suceder que se vea sobrecargado de datos de diferente naturaleza. Para afrontar este problema, es aconsejable utilizar un proceso gradual que evolucione desde el aprendizaje a la definición. Durante la fase de aprendizaje, el foco debe ubicarse en los hechos actuales que deberían ser organizados y registrados. Luego, deben comprenderse los planes a futuro; pero en lugar de registrarlos tal como se los obtiene, deberían ser usados para diseñar modelos del UdeD futuro [17].

En el ámbito de Ingeniería de Requisitos, los clientes-usuarios prefieren descripciones basadas en lenguaje natural en lugar de esquemas técnicos para comunicarse con los desarrolladores de software. Y mejor aún si el lenguaje usado en esas representaciones incluye tanta jerga del cliente-usuario que sea posible. Cuando esto sucede, el compromiso del cliente-usuario con el proyecto aumenta. Esta es la principal razón por la que muchos autores del área de Ingeniería de Requisitos, proponen estrategias basadas en lenguaje natural [18] [19] [20] [21].

En este trabajo se utilizan los modelos resultantes de la aplicación de una estrategia basada en lenguaje natural para la elicitación de requisitos, que utiliza dos representaciones principales: el Léxico Extendido del Lenguaje (LEL) y los Escenarios a través de todo el proceso de desarrollo de software. El principal propósito del léxico es capturar el vocabulario de la aplicación y su semántica, posponiendo la comprensión de la funcionalidad de la aplicación [22]. Los escenarios son usados para entender la aplicación y su funcionalidad: cada escenario describe una situación específica de la aplicación centrando la atención en el comportamiento [13]. Esta estrategia maneja las situaciones desde dos puntos de vista: como son en la actualidad, y como se espera que sean en el futuro. Las primeras son llamadas situaciones actuales, y las segundas, situaciones futuras.

Puesto que la estrategia descrita se aplica en un mundo cambiante, donde las reglas y procesos son más y más dependientes del software, es necesario considerar la idea de una baseline de requisitos que provea una base para administrar los cambios que se producen a lo largo del proceso de desarrollo del software. Esta baseline es el canal de comunicación con el mundo en el que el software se utiliza [17].

La baseline de requisitos evoluciona junto al proceso de desarrollo de software y acompaña las tareas de mantenimiento. Es importante enfatizar la importancia de la capacidad de referencias cruzadas entre la baseline de

requisitos y el proceso de desarrollo y mantenimiento [23]. Esto permite el seguimiento de los requisitos en cualquier punto durante el desarrollo y el mantenimiento hacia su punto de origen. En este contexto, los escenarios son descripciones de situaciones que evolucionan en el ambiente. Comienzan describiendo situaciones en el macrosistema, y luego, evolucionan a lo largo del proceso de construcción de software, describiendo cómo se modifican estas situaciones en el macrosistema.

En primer lugar, la estrategia produce el léxico del UdeD y luego los escenarios para obtener conocimiento acerca del ambiente en el cual el software será usado. Los escenarios actuales son usados para modelar el UdeD como es percibido y los escenarios futuros proyectan las situaciones que surgirán como consecuencia de la evolución. Los escenarios futuros utilizados en esta estrategia están más cerca de los casos de uso que se describen en la literatura, que a los escenarios vistos como instancias de casos de uso [23].

Una vez modelados los escenarios actuales y futuros, aquellas responsabilidades correspondientes a seres humanos o sistemas existentes en el UdeD actual, que se convierten en responsabilidades del nuevo sistema en el UdeD futuro, serán los requisitos del sistema de software.

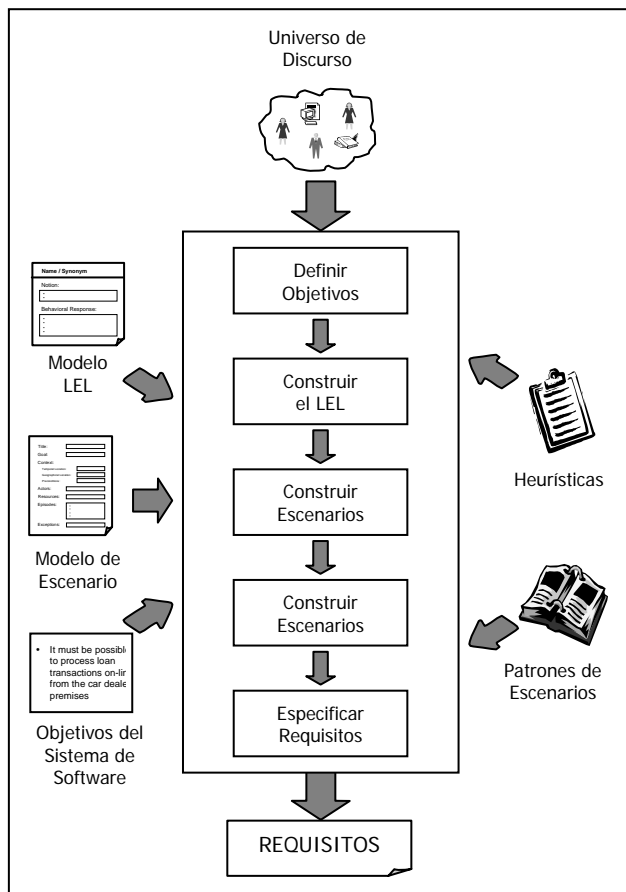


Figura 1. Proceso de obtención de Requisitos

En la figura 1 se presenta un esquema resumido de la estrategia descrita, donde se puede observar cuáles son las actividades que se llevan a cabo hasta llegar finalmente al modelo de requisitos.

4. Aplicación a un caso de estudio

El caso de estudio que se utilizó en este trabajo corresponde a un Sistema de Planes de Ahorro Previo para la Adquisición de Vehículos 0Km [24] [25]. Este sistema se dedica a la gestión y la administración de planes de ahorro para adquirir automotores 0Km. Funciona a través de grupos de una cierta cantidad de personas físicas o legales, los cuales participan mensualmente de los sorteos que la Administradora organiza, con el objeto de entregar una unidad por grupo. Los integrantes o adherentes de cada grupo pueden tratar de anticipar la entrega de la unidad que les corresponde, presentando ofertas de licitación. En cada acto de adjudicación, además de sortear el número del adherente favorecido de cada grupo, se abren las ofertas de licitación, si las hubiera, para determinar la mejor oferta y establecer así el beneficiario.

La complejidad de este caso reside en el gran número de situaciones que se generan desde la solicitud de ingreso de un adherente hasta el pago de la última cuota del plan, incluyendo una serie de situaciones especiales asociadas con el cambio del bien objeto del plan por parte del fabricante o por parte del adjudicatario y las situaciones que se derivan de demoras en los pagos, pagos por adelantado, renuncia, fallecimiento o expulsión de los adherentes, etc.

Para este caso de estudio, desarrollado en la Universidad Nacional del Centro de la Provincia de Buenos Aires, se utilizó como técnica de recolección de datos la lectura de documentos.

Nueve grupos, utilizando la misma técnica de elicitación, procedieron a la confección del Léxico Extendido del Lenguaje. Posteriormente, a partir de este modelo, cada uno de los grupos construyó un modelo de escenarios actuales.

El caso fue realizado con fines totalmente ajenos a este trabajo, y sólo tiempo después se decidió revisarlo con el propósito de estudiar la completitud. Esta secuencia de acontecimientos, garantiza absolutamente la ausencia de factores significativos que pudieran entorpecer la validez de los datos analizados.

4.1. Estimación del número de símbolos del LEL

En un trabajo previo se aplicó el método DPM [14] para estimar el número de símbolos del LEL, obteniendo el gráfico que se presenta en la figura 2. El número total de símbolos diferentes detectados por los nueve grupos fue de 118. Los datos obtenidos en el experimento fueron

ajustados con una curva exponencial decreciente con un coeficiente de determinación R^2 de casi un 90%. A partir de esta curva de ajuste se pudo estimar un número de 127 símbolos para el LEL correspondiente a este UdeD [5].

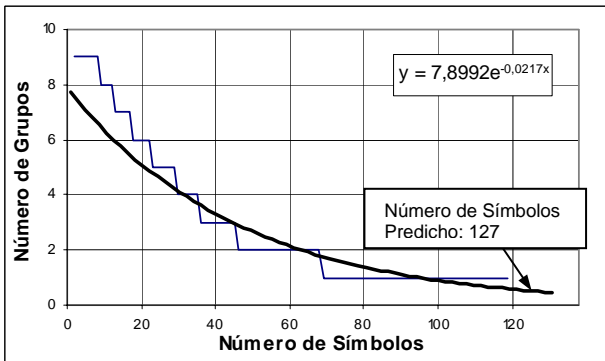


Figura 2. Predicción del número de símbolos del LEL a partir de los datos obtenidos por todos los grupos

4.2. Estimación del número de escenarios

En el desarrollo del caso de estudio, no todos los grupos mostraron la misma habilidad para aplicar la estrategia, tanto a la hora de detectar símbolos del LEL, como para construir los escenarios a partir del Léxico. Por este motivo, se optó por la utilización de DPM [14] para estimar el número total de elementos de cada modelo, en lugar de otro método de estimación.

Se efectuaron estimaciones para un grupo reducido de captore, en este caso aquellos grupos que mayor número de escenarios detectaron, obteniendo el resultado que se presenta en la figura 3. El número total de escenarios diferentes construidos por los tres grupos fue 92, y se estimó un total de 120 escenarios.

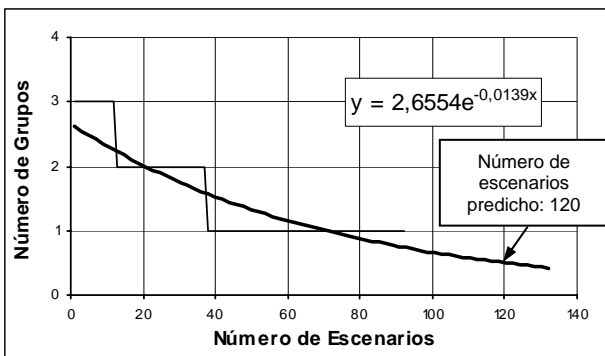


Figura 3. Predicción a partir de los escenarios obtenidos por los grupos 1, 5 y 7

Luego se efectuó la estimación para todos los grupos participantes, que detectaron un total de 114 escenarios diferentes. El número estimado aplicando DPM, en este

caso, fue de 126. En la figura 4 puede observarse el gráfico de los datos y la curva de ajuste correspondiente.

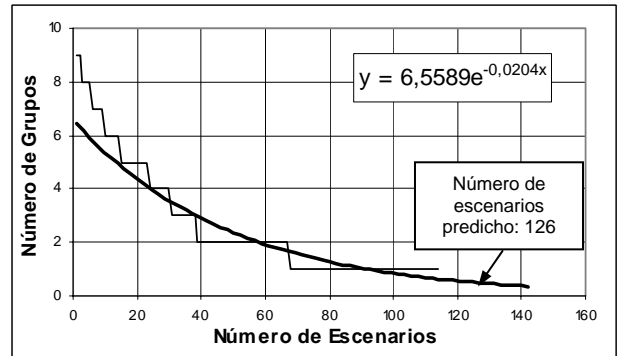


Figura 4. Predicción a partir de los escenarios obtenidos por todos los grupos

Del mismo modo que para las predicciones del número total de símbolos del LEL, se observa que al aumentar el número de captore, la cantidad total estimada parece ser más cercana al número real de escenarios. Cuando el número de observaciones independientes es 3, la predicción indica que quedan por descubrir 28 escenarios, ya que, para un total de 92 escenarios diferentes, se predice un total de 120. Para los 9 grupos, en cambio, el número de escenarios por descubrir, de acuerdo con la predicción, se reduce. En este caso, para un total de 114 escenarios diferentes descubiertos, la estimación indica un número total de 126, con lo que quedarían por descubrir sólo 12 escenarios.

Al aumentar el número de captore, aumenta también la precisión obtenida al ajustar los datos con una curva, ya que, al considerar sólo 3 grupos, el coeficiente de determinación R^2 es de 76%, mientras que para todos los grupos, este coeficiente alcanza casi un 90%

4.2. Análisis del impacto de la completitud del LEL sobre los escenarios

La estrategia aplicada en el caso de estudio consiste en una serie de modelos que se van construyendo en forma encadenada, hasta obtener al final una lista de requisitos. Dado que ésta debería ser lo más completa posible si se pretende obtener un producto de software de buena calidad, el objetivo final de una estimación en esta área debería ser el número total de requisitos. Por lo tanto, surge la siguiente pregunta: ¿la completitud de los requisitos depende de la completitud de cada uno de los modelos intermedios? Para saberlo, es necesario estudiar de qué manera impacta el grado de completitud de cada uno de esos modelos en los siguientes.

En particular, en este trabajo analizamos la relación que existe entre los símbolos del LEL y los escenarios actuales.

La heurística de derivación de escenarios establece como punto de partida para la creación de los escenarios actuales los impactos de los símbolos del LEL de tipo Sujeto [13]. Por esto, también se estudia la relación entre el número de impactos para símbolos de tipo sujeto detectado por cada grupo, y el número de escenarios obtenido por el mismo grupo. En la Tabla 1 se presentan los datos correspondientes.

Tabla 1. Número de símbolos, impactos para sujetos, y escenarios para cada grupo

Grupo	Número de símbolos	Número de impactos de símbolos de tipo Sujeto	Número de Escenarios
1	52	75	54
2	29	39	35
3	30	29	24
4	33	32	27
5	65	49	43
6	13	16	12
7	44	61	44
8	24	34	34
9	54	28	29

La figura 5 presenta la relación entre el número de símbolos del LEL y el número de escenarios obtenidos por cada grupo. La figura 6, muestra la relación existente entre el número de impactos de símbolos de tipo Sujeto y los Escenarios obtenidos.

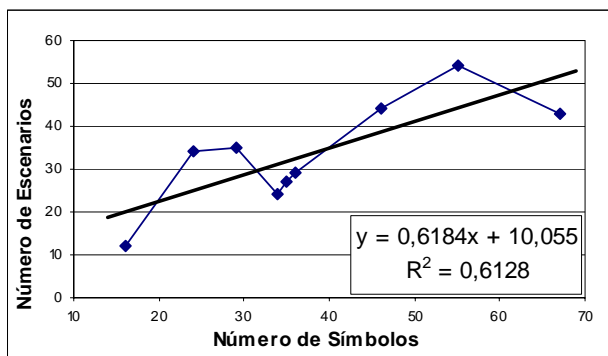


Figura 5. Relación entre número de símbolos del LEL y Escenarios

Se presenta en ambas figuras una línea de tendencia para los datos graficados, pudiéndose observar que existe una tendencia lineal en ambas figuras. Sin embargo, esta tendencia es mucho más evidente en el caso de la figura 6 que en la 5.

El análisis de los datos obtenidos nos permite concluir que, si bien la completitud externa del LEL es importante para lograr completitud en el modelo de escenarios, más

importante es la completitud interna de los símbolos del LEL. Es decir, no alcanza con detectar la mayor cantidad posible de símbolos que representan el lenguaje del UdeD, sino que cada uno de ellos, y en especial los sujetos, deben ser descriptos de la forma más completa posible.

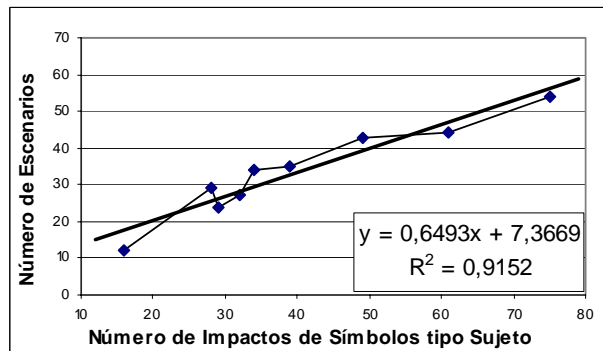


Figura 6. Relación entre número de impactos de símbolos de tipo Sujeto y Escenarios

5. Conclusiones y trabajo futuro

Se ha concluido con una certeza estadística del 61% que la cantidad de escenarios depende de la cantidad de símbolos del LEL y con un 91% de la cantidad de impactos correspondientes a los sujetos del LEL.

Se ha comprobado, entonces, que la completitud interna de los símbolos del LEL es tan o más importante que la completitud externa.

Ambas conclusiones directas impulsan fuertemente a rehacer, en el futuro, los estudios de completitud del LEL y de los Escenarios actuales, basándose en una granularidad más fina, es decir, analizando impactos únicos en el LEL y episodios únicos en los escenarios.

Un problema aún no resuelto, que debe ser atendido, es definir una métrica de granularidad precisa y heurísticas que permitan detectar, analizar y contar impactos incluidos en otros impactos, así como episodios incluidos en otros episodios. De este modo, será posible efectuar un estudio más preciso de la completitud interna del LEL y de los escenarios.

También queda por estudiar cómo impacta el grado de completitud del LEL y los Escenarios en los modelos siguientes, y en especial cómo impacta la completitud de cada uno de los modelos en la calidad de los requisitos obtenidos.

Un aspecto interesante a investigar es la extensión de la aplicación de estas técnicas a procesos de obtención de requisitos basados en otro tipo de representaciones, incluyendo eventualmente modelos no basados en lenguaje natural.

6. Referencias

- [1] Maculay, L.: "Requirements Capture as a Cooperative Activity". *Proceedings IEEE International Symposium on Requirement Engineering*. IEEE Computer Society Press, San Diego, Ca, pp. 174-181, 1993.
- [2] Reubenstein, H. B., Waters, R.C.: "The requirements apprentice: Automated assistance for requirements acquisition". *IEEE Transaction on Software Engineering*, Vol. 17, N° 3, pp 226-240, 1998.
- [3] Bubenko, J.A., Wrangler, B.: "Objectives driven capture of business rules and information systems requirements". *Proceedings de IEEE Conference on Systems, Man and Cybernetics*. Vol. 1, pp. 670-677, 1993.
- [4] Kotonya, G., Sommerville, I.: *Requirements Engineering. Processes and Techniques*. John Wiley & Sons, 1998.
- [5] Doorn, J., Ridao, M.: "Compleitud de Glosarios: Un Estudio Experimental". *Proceedings de WER03 - Workshop em Engenharia de Requisitos*, Piracicaba-SP, Brasil, Noviembre 27-28, pp. 317-328, 2003.
- [6] Otis, D.L., Burnham, K.P. White G.C., Anderson D.R.: "Statistical inference from Capture on Closed Animal Populations". *Wildlife Monograph*, 62, 1978.
- [7] G. White, D. Anderson, K. Burnham, and D. Otis.: "Capture-Recapture and Removal Methods for Sampling Closed Populations", Technical Report LA-8787-NERP, Los Alamos Nat'l Laboratory, 1982.
- [8] Briand, L.: El Emam, K., Freimut, B., Laitenberger, O.: "A Comprehensive Evaluation of Capture-Recapture Models for Estimating software Defects Contents". *IEEE Transactions on Software Engineering*, Vol. 26, N° 6, pp. 518-540, 2000.
- [9] Biffel, S.: "Evaluating defect estimation models with major defects". *The Journal of Systems and Software*, 65, pp. 13-29, 2003.
- [10] Kamel, A., Sorenson, P.: "The Application of Capture-Recapture Log-Linear Models To Software Inspections Data". *Proceedings of ISESE'03 - International Symposium on Empirical Software Engineering*, pp. 213-222, 2003.
- [11] Thelin, T.: "Team-based Fault Content Estimation in the Software Inspection Process". *Proceedings of ICSE'04 - 26th International Conference on Software Engineering*, pp 263-272, 2004.
- [12] Petersson, H., Thelin, T., Runeson, P., Wohlin, C.: "Capture-recapture in software inspections after 10 years research—theory, evaluation and application". *The Journal of Systems and Software*, 72, pp. 249–264, 2003.
- [13] Leite, J.C.S.P., Hadad, G.D.S., Doorn, J.H., Kaplan, G.N.: "A Scenario Construction Process". *Requirements Engineering Journal*, Vol.5, N° 1, pp. 38-61, 2000.
- [14] Wohlin, C., Runeson, P.: "Defect content estimations from Review Data". *Proceedings of the 20th International Conference on Software Engineering*, pp. 400-409, 1998.
- [15] Briand, L., El Emam, K., Freimut, B., Laitenberger, O.: "Quantitative evaluation of Capture Recapture Models to Control Software Inspections". *Proceedings of the 8th International Symposium on Software Reliability Engineering*, pp 234-244, 1997.
- [16] Thelin, T., Runeson, P.: "Fault Content Estimations using Extended Curve Fitting Models and Model Selection". *Proceedings of EASE'00 - International Conference on Empirical Assessment & Evaluation in Software Engineering*, 2000.
- [17] Leite, J.C.S.P., Doorn, J.H., Hadad, G.D.S., Kaplan, G.N., Ridao, M.: "Defining System Context Using Scenarios". *Perspectives of Software Requirements*. Kluwer Academic Publishers. Norwell, Massachusetts. USA. ISBN 1-4020-7625-8, 2004.
- [18] Jackson, M.: *Software Requirements & Specifications. A lexicon of practice, principles and prejudices*. Addison Wesley, ACM Press, 1995.
- [19] Leite, J.C.S.P., Franco, A.P.M.: "A Strategy for Conceptual Model Acquisition". *Proceedings of RE'93 - IEEE International Symposium on Requirements Engineering*; 1993 January 4-6; San Diego, CA.. Los Alamitos, CA: IEEE Computer Society Press, pp. 243-246, 1993.
- [20] Alspaugh, T. A., Antón, A. I., Barnes, T., Mott, B. W.: "An Integrated Scenario Management Strategy". *Proceedings of RE'99 - International Symposium On Requirements Engineering*; 1999 June 7-11; Limerick-Ireland. Los Alamitos, CA: IEEE Computer Society Press , pp. 142-149, 1999.
- [21] Carroll, J.: "Introduction: The Scenario Perspective on System Development". *Scenario-Based Design: Envisioning Work and Technology in System Development*, J. Carroll, ed.: John Wiley & Sons, New York, pp. 1-18, 1995.
- [22] Leite, J.C.S.P., Franco, A.P.M.: "O Uso de Hipertexto na Elicitação de Linguagens da Aplicação". *Anais de IV Simpósio Brasileiro de Engenharia de Software*. SBC, Brazil, pp. 134-149, 1990.
- [23] Breitman, K., Leite, J.C.S.P., Berry, D. "Supporting scenario evolution". *Requirements Engineering*, 10, pp. 112–131, 2005.
- [24] Mauco, V., Ridao M., del Fresno, M., Rivero, L., Doorn, J.H.: "Ingeniería de Requisitos, Proyecto: Sistema de Planes de Ahorro". Reporte técnico, ISISTAN, UNCPBA, Tandil, Argentina, 1997.
- [25] Rivero, L., Doorn, J., del Fresno, M., Mauco, V., Ridao, M., Leonardi, M.C.: "Una Estrategia de Análisis Orientada a Objetos basada en Escenarios: Aplicación en un Caso Real". *Proceedings of WER'98 - Workshop en Engenharia de Requisitos*, Maringá, Brasil, pp. 79-90, 1998.