

# Aplicaciones de la Teoría de Constructos Personales a la Elicitación de Requisitos

Bruno González-Baixauli<sup>1</sup>, Miguel A. Laguna<sup>1</sup>, Julio Cesar Sampaio do Prado Leite<sup>2</sup>

<sup>1</sup>Departamento de Informática. Universidad de Valladolid  
{bbaixauli, mlaguna}@infor.uva.es

<sup>2</sup>Departamento de Informática. PUC do Rio de Janeiro  
www.inf.puc-rio.br/~julio/

**Abstract.** La Ingeniería de Requisitos es un área considerada clave en la Ingeniería del Software. Dentro de esta área, uno de los principales enfoques es el de la orientación a objetivos. Según este enfoque, que se basa en ideas de inteligencia artificial, los objetivos son un concepto natural y de alto nivel de abstracción para la elicitación y representación de requisitos. Además, permiten una sistematización del manejo de los requisitos no funcionales. Sin embargo, la elicitación de los objetivos mismos y sus relaciones apenas se ha desarrollado. El problema de toda elicitación es recoger toda la información precisa, sobre todo aquella que está oculta por ser difícil de expresar. Dentro de la Psicología, existen diversos métodos para obtener este conocimiento. En este artículo se exploran varias aplicaciones de una teoría psicológica, la Teoría de Constructos Personales para la elicitación de objetivos. Esta teoría tiene la ventaja frente a otras de tener una base estadística y, por tanto, ser más rigurosa e independiente de quien la ponga en práctica, y su posibilidad de soportar su automatización gracias a la técnica del *Repertory Grid*.

## 1 Introducción

La Ingeniería de Requisitos (IR) trata la cuestión clave en todo proceso de producción, el producir y, por tanto es uno de los principales campos de trabajo de la Ingeniería del Software. Esto se debe a que los requisitos representan uno de los principales problemas durante el desarrollo del software y, su no cumplimiento, una de las principales quejas de los clientes. Dentro de esta rama de la Ingeniería del Software, los objetivos (*goals*) aparecen como una de las propuestas de modelado para fijar el proceso de definición de requisitos. Los objetivos proporcionan una visión intencional con varias ventajas respecto a otros mecanismos como su mayor estabilidad [16] (es más difícil que cambien las razones por las que se necesita un sistema, que la forma en que se realizan sus funciones) o su posibilidad de analizar distintas alternativas y seleccionar la mejor a partir de los requisitos no funcionales [2]. Además, este análisis puede ser realizado adaptando técnicas de Inteligencia Artificial, mediante un razonamiento formal sobre los modelos de objetivos [6].

Uno de los principales desafíos de la IR es la manera de elicitar estos requisitos. Aunque los objetivos proporcionan una manera más natural de expresarlos, hay poco trabajo realizado en su elicitación. Los requisitos de un sistema se obtienen básicamente a partir del Universo de Discurso<sup>1</sup> (UdeD). En el UdeD están todas las fuentes de información, incluso los interesados<sup>2</sup>. Obtener o elicitar los requisitos de los interesados es una tarea difícil, tanto por el conocimiento tácito [7] como por las diversas opiniones, probablemente contrapuestas, que se pueden obtener [10]. En psicología, el conocimiento tácito es también conocido como conocimiento oculto o profundo y existen diversas teorías cuyo fin es sacarlo a la luz.

La Teoría de Constructos Personales (TCP), desarrollada por Kelly en 1955 [9] es una de las teorías psicológicas más completas que tratan de este tema. Kelly afirma que las personas utilizan dimensiones cognitivas denominadas “constructos” para evaluar su experiencia personal. Por tanto, dichos constructos son los elementos de la visión del mundo de la persona, a partir de los cuales juzga y toma decisiones. Un resultado práctico de la teoría es la técnica del “*repertory grid*”, que permite relacionar constructos con elementos por medio de una matriz o rejilla y obtener nuevo conocimiento creando reglas entre constructos.

Este artículo explora las posibles aplicaciones de esta teoría y, en particular de la técnica del *repertory grid* a la elicitación de objetivos. Está organizado como sigue: primero se revisan los principales enfoques de IR orientada a objetivos. En la sección 3, se presenta la Teoría de Constructos Personales y el “*repertory grid*”. En la sección 4 se muestran las aplicaciones que estimamos más prometedoras de dicha técnica y en la última sección se termina con las conclusiones y trabajo futuro.

## 2 Ingeniería de Requisitos Orientada a Objetivos

La Ingeniería de Requisitos orientada a objetivos (*goal-oriented requirement engineering*). El enfoque que promueve es la utilización de objetivos como base para obtener los requisitos, incorporando de esta forma un punto de vista intencional, es decir, el propósito del sistema. Introducir un punto de vista intencional permite que los interesados expresen sus necesidades de una manera más natural, centrándose en lo que quieren (sus objetivos) frente a la manera de alcanzarlos (los requisitos convencionales). A partir de los objetivos, los requisitos se pueden derivar como maneras de alcanzar esos objetivos.

Los objetivos se suelen estructurar en árboles And-Or descomponiéndose en otros obligatorios u optativos. Así, es posible estudiar alternativas en los requisitos y verificar la completitud de un conjunto de requisitos con respecto a los objetivos planteados. Además este estudio puede hacerse de manera formal si los objetivos están suficientemente formalizados. Por tanto, los objetivos ayudan a identificar,

---

<sup>1</sup> El contexto total en el cual el software se desarrollará y operará. El UdeD incluye todas las fuentes de información y personas relacionadas con el software. Es la realidad acotada por el conjunto de objetivos establecidos por aquellos que demandan una solución software [11].

<sup>2</sup> Los interesados o *stakeholders* son todas aquellas personas involucradas en el proceso de desarrollo del software. Pueden ser clientes, usuarios, financiadores, etc. y muchos otros papeles, incluyendo incluso el de los ingenieros del software.

justificar y organizar los requisitos. Otras ventajas de los objetivos son que los requisitos resultantes son más independientes de la tecnología, que el tratamiento de requisitos no funcionales (RNF) y conflictos es más natural y que guían el desarrollo software permitiendo trazar los artefactos software hasta sus objetivos finales [16].

Existen diversos enfoques dentro de la IR orientada a objetivos, que distan mucho de ser homogéneos, si bien los conceptos básicos que utilizan son similares. Los enfoques se diferencian básicamente en dos factores: la actividad de IR a la que se enfocan (elicitación, modelado o análisis de requisitos) y el grado de formalismo que soportan. Por ejemplo, la propuesta de KAOS (*Knowledge Acquisition in Automated Specification of Software*) [4] está enfocada al modelado de requisitos con un alto grado de formalismo, lo que le permite realizar un proceso de análisis formal de los requisitos. En cambio, el *NFR Framework* [2], aunque también está enfocado al modelado de requisitos, se centra en RNF, y utiliza un enfoque cualitativo mucho menos formal. Un enfoque que se basa en el *NFR Framework* es *i\** [15], centrado en la elicitación de requisitos en las fases iniciales de la IR, es decir en el modelado de negocio. Otras propuestas buscan integrar otras técnicas como los escenarios en el modelado de objetivos como el caso de GBRAM (*Goal Based Requirements Analysis Method*) [1]. Su principal interés es que define un proceso en el cual los escenarios se crean a partir de los objetivos y, a su vez ayudan a descubrir objetivos ocultos.

Un problema de estas propuestas es que no proporcionan un enfoque sistemático para la elicitación de los elementos, ni de las relaciones entre ellos. Por lo general se dan consejos, pero no técnicas orientadas específicamente a la obtención de los objetivos de los interesados. Sí que existen propuestas que utilizan casos de uso o escenarios con objetivos, pero en la mayoría de los casos los objetivos no tienen la importancia que se les da en la IR orientada a objetivos.

En las siguientes subsecciones se presentan dos de los enfoques orientados a objetivos más importantes, que serán la base para explorar las aplicaciones de la TCP. Se han elegido estos enfoques debido a la importancia que dan a los RNF que, como veremos, es el concepto más cercano al de constructo de la TCP.

## 2.1 NFR Framework

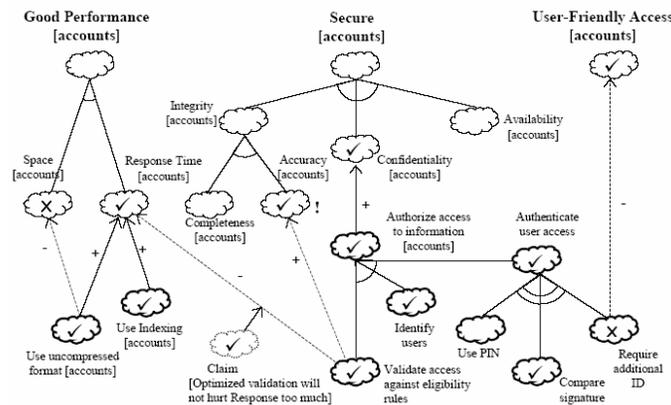
El propósito fundamental del *NFR Framework* es, como su nombre indica, tratar con los requisitos no funcionales. Los RNF son considerados una parte fundamental de la Ingeniería del Software desde hace tiempo, pero solo han sido considerados a partir de la fase de diseño, por eso existen pocas propuestas para introducirlos de una manera sistemática en el desarrollo a partir de requisitos [3]. El *NFR Framework* aboga por analizarlos en la fase de requisitos por medio de objetivos. El primer problema que aparece es que los objetivos tradicionales se deben poder verificar si se han alcanzado, pero los RNF son mucho más difusos, ya que no suelen tener una definición completa. Por tanto, no siempre se puede afirmar si se ha logrado el objetivo al depender de muchos factores. Para solucionar esto Chung et al. [2] definen un nuevo concepto: el *softgoal*, es decir, un objetivo para el cual su satisfacción no siempre se puede establecer de manera clara. Así, un *softgoal* puede considerarse alcanzado o no, pero también parcialmente satisfecho y parcialmente insatisfecho. Por tanto, no se habla de satisfacción sino de grado de satisfacción.

A partir de este concepto se define un modelo, el *Softgoal Interdependence Graph* (SIG). Este modelo permite relacionar *softgoals* por medio de las relaciones típicas dentro de los modelos de objetivos (descomposición en árboles And-Or) y por medio de un conjunto de nuevas relaciones denominadas enlaces de correlación. Estos enlaces son la base para calcular el grado de satisfacción y se resumen en la Tabla 1.

↑ -	↑ -	↑ ?	↑ +	↑ ++
BREAK	HURT	UNKNOWN	HELP	MAKE

**Tabla 1.** Enlaces de correlación. Adaptado de [2].

Otros elementos que se utilizan en los SIG son las operacionalizaciones (*operationalization*), que permiten enlazar los requisitos no funcionales con los funcionales y las afirmaciones (*claim*) que modifican correlaciones según sea cierta o no la afirmación. Además, es posible indicar la prioridad de los *softgoals* por medio de signos de exclamación. Un ejemplo de uno de estos modelos se presenta en Fig. 1. Es importante señalar que en el *NFR framework* no se especifica un modelo de requisitos funcionales, pero lo más natural es utilizar también objetivos en este modelo como en [12].



**Fig. 1.** Ejemplo de SIG. Las nubes de trazo fino representan *softgoals*, las de trazo grueso operacionalizaciones y las punteadas afirmaciones. Las relaciones de un arco son *And* y las de 2 son *Or*. Adaptado de [2].

Gracias a estas relaciones es posible especificar explícitamente como unos *softgoals* afectan a otros. De esta forma, dando valores a algunos *softgoals* y propagándolos según las correlaciones, se puede obtener el grado de satisfacción del resto. En la propuesta inicial [2], los valores son cualitativos y su propagación se hace de manera manual, pero hay trabajos en los que se automatiza, como en [6] donde incluso se propone un enfoque cuantitativo y se permite el uso de relaciones asimétricas en las que solo se propaga la parte positiva o la negativa.

## 2.2 I\* Framework

i\* es un enfoque, propuesto por Yu [15], que utiliza la idea de *softgoal*. La principal particularidad del modelado de negocio sobre otros campos de la IR es la importancia de los agentes. Un agente se define como una entidad que existe en la organización que tiene objetivos y que puede realizar tareas o utilizar recursos para alcanzar dichos objetivos o ayudar a otros agentes a alcanzar sus objetivos. En esta definición se incluyen los elementos de modelado de esta propuesta: objetivos (*goals* y *softgoals*), agentes, tareas (manera de alcanzar un objetivo) y recursos (objeto que es necesario para realizar una tarea o para alcanzar un objetivo).

A partir de estos elementos se definen dos modelos en los que los agentes son la pieza fundamental. En el primero, el *Strategic Diagram* (SD) se modelan las dependencias entre actores. Estas dependencias pueden deberse a objetivos (*goals* o *softgoals*), a tareas o a recursos y dan una visión global del sistema. En cuanto al segundo, denominado *Strategic Rationale* (SR) lo que se modela es el razonamiento interno del agente mediante objetivos y tareas.

En la Fig. 2 se puede ver un ejemplo de un modelo SR que modela el negocio de la multinacional IKEA. Un modelo SD es similar al anterior, pero sin mostrar los objetivos y tareas internos de los agentes.

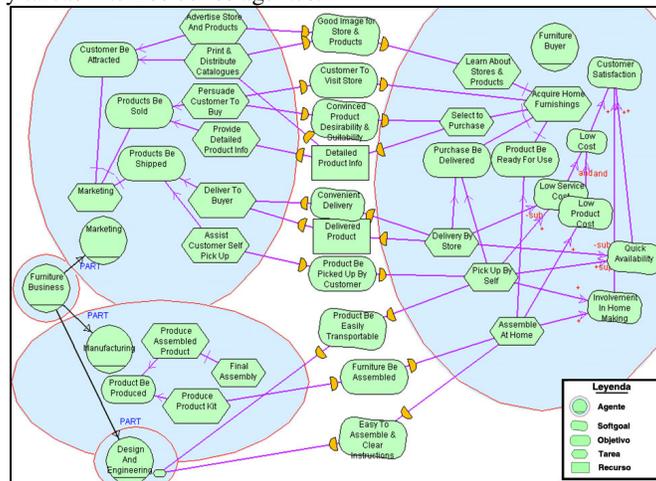


Fig. 2. Modelo SR de i\* para el modelo de negocio de IKEA. Adaptado de [17]

## 3 Teoría de Constructos Personales

La Teoría de Constructos Personales (TCP) es una teoría psicológica constructivista propuesta por Kelly en 1955 [9] que trata de explicar la visión del mundo de una persona por medio de constructos. Un constructo es un elemento de conocimiento con dos polos opuestos. Según Kelly, el conjunto de constructos de una persona y las

relaciones entre ellos da la base de los juicios y comportamientos de dicha persona, puesto que le permiten hacer hipótesis y llegar a conclusiones.

La principal ventaja de esta teoría psicológica frente a otras que también permiten obtener parte del conocimiento oculto de las persona es que es una teoría rigurosa. Por tanto, es más precisa y las conclusiones se ven menos afectadas por personas externas o por el experto.

Además, Kelly desarrolló una técnica basada en esta teoría, el *Repertory Grid* o rejilla, para elicitar el conocimiento oculto de una persona. Esta técnica, que ha sido automatizada en el campo de la gestión de conocimiento [5], presenta un gran potencial para la elicitación de objetivos por permitir elicitar el conocimiento difícil de expresar y porque saca a la luz relaciones entre los distintos constructos.

### 3.1 Repertory Grid

La técnica del *Repertory Grid* es básicamente una entrevista en la que el sujeto de la entrevista debe evaluar un conjunto de elementos sobre la base del conjunto de constructos. Por tanto, los elementos representan elementos de interés, que pueden ser tanto concretos como abstractos, y los constructos representan las maneras por las cuales los elementos son juzgados. Los resultados de la evaluación se muestran en una matriz de constructos (filas) por elementos (columnas).

La rejilla propuesta por Kelly originalmente solamente admitía constructos bipolares, pero varias propuestas posteriores permiten que sean multivaluados. La ventaja de utilizar constructos multivaluados es que dan una mayor libertad para evaluar los constructos y permiten un análisis más complejo. En la Tabla 3 hay un ejemplo de rejilla multivaluada resumida a partir de un ejemplo de [5]. Es un ejemplo clásico de psicología en el cual el sujeto de estudio evalúa a sus conocidos, donde los constructos son características de las personas como honesto/deshonesto y los elementos son personas como su padre, o conceptos abstractos como el “yo deseado”.

		Elementos					
		<i>Jose</i>	<i>Victor</i>	<i>Ana</i>	<i>Merche</i>	<i>Alberto</i>	
Constructos	Honesto	2	4	2	1	1	Deshonesto
	Generoso	2	4	3	2	1	Avaro
	Simple	4	3	2	3	4	Complejo
	Atrevido	1	2	4	3	4	Prudente
	Motivado	1	2	4	3	4	Desmotivado

**Tabla 2.** Ejemplo de una rejilla donde una persona analiza las características de sus conocidos.

Existen numerosas técnicas para extraer nuevo conocimiento a partir de una rejilla. Por ejemplo, usando distancias entre vectores o aplicando lógica difusa (*fuzzy logic*). Sin embargo, la técnica que se aplica en este artículo es la propuesta por Ford [5], que permite obtener implicaciones parciales (y por tanto relaciones asimétricas) entre constructos de una manera fácil de automatizar.

El proceso de análisis es muy simple: primero se separa la matriz de la rejilla en tantas matrices como valores se permitan a los constructos con un valor asignado de manera que cada celda tendrá un 1 si el elemento tenía ese valor para el constructo o 0

si no. A partir de estas matrices binarias, se comparan los constructos para ciertos valores con la siguiente fórmula, donde  $X$  es el vector de elementos y  $\Psi$  y  $\Pi$  son valores de un constructo como “*extremadamente(1) honesto*” o “*algo(4) complejo*”.

$$\begin{aligned}
 \text{incidence of } \Psi \quad i(\Psi)_i &= \begin{cases} 1 & \text{if } x_i \text{ is a } \Psi \\ 0 & \text{otherwise} \end{cases} \\
 \text{ocurrence of } \Psi \text{ in } i(\Psi) & \quad \text{occ}(i(\Psi)) \\
 \text{degree of Confirmation of "All } \Psi \text{'s are } \Pi \text{"}: & \\
 P(\Pi|\Psi) &= \frac{\text{occ}(i(\Psi) \cap i(\Pi))}{\text{occ}(i(\Psi))}
 \end{aligned}$$

De tal forma que, con este análisis, se obtienen reglas del tipo:

$$(\text{Valor\_Constructo\_1}) \text{ Constructo1 } \mathbf{implica} (\text{Valor\_Constructo2}) \text{ Constructo\_2} \\
 \text{[Grado\_de\_Confirmación]}$$

Que si damos un nombre a los valores de los constructos, por ejemplo *extremadamente* para el mayor valor quedaría:

$$\text{Extremadamente Constructo\_1 } \mathbf{implica} \text{ Extremadamente Constructo\_2 [ ]}$$

Otra opción de análisis es agrupar valores de constructos, en ese caso se crearía una nueva matriz a comparar. En este caso, las reglas quedarían:

$$\text{Al menos Algo Constructo\_1 } \mathbf{implica} \text{ Al menos Algo Constructo\_2 [ ]}$$

Donde se pueden distinguir dos características de las reglas, por un lado la precisión (utilizar un valor del constructo frente a un conjunto de valores) y el grado de verdad (el grado de confirmación) que nos permite hacernos una idea de la validez de la regla.

Hay que señalar que el cálculo del grado de confirmación se realiza de una manera estadística, por lo tanto el número de elementos debería de ser suficientemente amplio para poder dar por válidos esos resultados. Por último, esos resultados sólo serán válidos para la población de donde están sacados los elementos, es decir, si los elementos son todos de un mismo tipo de sistema, los resultados solo serán válidos para ese tipo de sistema.

#### 4 Aplicaciones a la IR Orientada a Objetivos

Como ya se ha comentado, el interés de la TCP es la elicitación de conocimiento que son difíciles de expresar. Aplicando la técnica del *repertory grid* se obtienen reglas que relacionan constructos a partir de elementos. Por tanto, lo primero que debemos hacer es determinar que conceptos vamos a utilizar como constructos y cuales como elementos. Hay que recordar que aquello que queramos relacionar debe ser considerado un constructo, puesto que las reglas que vamos a obtener son relaciones entre constructos. Las siguientes sub-secciones exploran algunas de las posibles aplicaciones con distintos elementos y constructos, y los avances que hemos obtenido en alguna de ellas.

#### 4.1 Utilización de Interesados como Elementos

La idea de utilizar interesados como elementos es la primera a considerar, puesto que la teoría psicológica trata con personas. Así, si realizamos un test a cada interesado y posteriormente se unen en una misma rejilla, se pueden analizar los resultados. Aquí se puede intentar conocer la importancia relativa de cada constructo (objetivo) para determinados agentes (interesados). Así, se obtiene una idea de cómo un sujeto ve la relación de objetivos con agentes del UdeD. Los polos serían, poca afinidad (1) y gran afinidad (5) de un objetivo para un elemento (interesado). Los valores enteros intermedios de los polos identifican variaciones dentro de esos polos.

Con este enfoque se puede descubrir relaciones entre objetivos. Este tipo de análisis desembocaría en reglas del tipo: *Constructo\_1 es afín con el Constructo\_2 (o no lo es o lo es en que grado)*.

Diversas posibilidades de análisis se presentan con las reglas. Por ejemplo: la detección de posibles conflictos (si  $C_1$  es afín a  $C_2$ , y  $C_2$  es afín a  $C_3$ , pero  $C_1$  no es afín de  $C_3$ ) y la detección de constructos si afinidad. En todos estos casos el análisis proporciona una realimentación aclarar las situaciones con los interesados. Con lo cual, se contribuye para que los objetivos (constructos) estén mejor definidos.

#### 4.2 Utilización de Funcionalidad como Elementos

La manera de encontrar correlaciones entre objetivos (*goals* y *softgoals*) es utilizar elementos de funcionalidad como elementos de la *rejilla*. Aquí el grano del elemento puede ser desde tareas hasta aplicaciones, pero hay que tener en cuenta que, para elementos de grano fino, su número disponible será escaso y, por tanto, al utilizar una técnica estadística, los resultados poco fiables. Por lo tanto hay que utilizar elementos de grano grueso, cuya población pueda aumentar. Consecuentemente las mejores elecciones son utilizar componentes en la selección de COTS o utilizar directamente aplicaciones de un dominio que nos pueden proporcionar relaciones entre objetivos para ese dominio.

Por ejemplo, si se consideran constructos a los *softgoals* se pueden utilizar los datos para construir los modelos SIG del *NFR Framework*. Para ello se relacionan los elementos funcionales con los *softgoals* por medio de los 5 valores definidos por el *framework* (satisfecho, parcialmente satisfecho, neutral, parcialmente insatisfecho o insatisfecho) y se obtienen las reglas por el método de análisis de Ford. A partir de estas reglas es posible obtener las correlaciones entre *softgoals* (Tabla 3).

<i>If ++ then ++ and If + then at least +</i>	(++s)	<i>If -- then -- and If - then at least -</i>	(++d)
<i>If ++ then -- and If + then at least -</i>	(-- s)	<i>If -- then ++ and If - then at least +</i>	(-- d)
<i>If at least + then at least +</i>	(+s)	<i>If at least - then at least +</i>	(+d)
<i>If at least + then at least -</i>	(- s)	<i>If at least - then at least -</i>	(- d)

**Tabla 3.** Conversión entre reglas obtenidas por el análisis de la rejilla y correlaciones RNF.

Es importante observar que se obtienen relaciones asimétricas (como las definidas en [6]). En el caso de que tengamos entre dos constructos (*softgoals*) la misma

correlación asimétrica para la parte positiva y negativa es posible utilizar reglas simétricas como se muestra en la tabla 4.

(++s) and (++d)	(++)	(--s) and (--d)	(--)
(+s) and (+d)	(+)	(-s) and (-d)	(-)
(++s) and (--d) or (--s) and (++d)	( )	(+s) and (-d) or (-s) and (+d)	( )

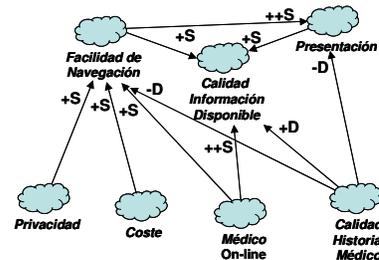
**Table 4.** Conversión de dos correlaciones asimétricas a una simétrica.

Las reglas obtenidas por este tipo de rejilla pueden ser usadas para otras aplicaciones, como por ejemplo:

- Crear un asistente que ayude a relacionar objetivos funcionales con *softgoals*. Los dos tipos de objetivos deben ser relacionados para poder elegir entre alternativas. Este asistente sugerirá relacionar el objetivo funcional con aquellos *softgoals* que estén relacionados con alguna *softgoal* ya evaluada.
- Si se utilizan como elementos las posibles alternativas dadas por el modelo de objetivos funcional, construyendo la rejilla de manera automática a partir de las relaciones entre objetivos, se puede calcular la distancia del sistema deseado (un vector con los valores de los *softgoals*) a las distintas alternativas. Este enfoque permite buscar la mejor alternativa.
- La rejilla también puede ser usada para detectar conflictos. Si existe una regla del tipo *si*  $SG_1 ++$  *implica*  $SG_2 ++$  y otra *si*  $SG_3 ++$  *implica*  $SG_2 --$  nos indica que  $SG_1$  y  $SG_3$  tienen un conflicto en el cumplimiento de  $SG_2$ .
- Las reglas entre objetivos se pueden utilizar para crear un árbol de decisión que a partir de los *softgoal* nos proporcione el conjunto de objetivos funcionales. Para ello, si existen reglas del tipo *si*  $G_1$  *implica*  $SG_2 --$  el objetivo  $G_1$  quedaría descartado, mientras que si son del tipo *si*  $G_1$  *implica*  $SG_2 ++$  la probabilidad de que  $G_1$  esté en el conjunto devuelto aumentará.

En la Fig. 3 se muestra parte de un ejemplo<sup>3</sup> de la aplicación de este método al dominio de las aplicaciones web de salud a partir de los datos en [13]. Se presenta una parte de la rejilla.

	CapMed	PatientSite	Your Health	IQHealth	ePatient	WMRS	98point6
Coste	1	5	5	3	5	1	5
Acceso Público	5	5	5	3	5	5	5
Calidad Historial Médico	5	5	4	5	4	5	5
Demo/ Tour	5	5	1	5	5	1	1
Médico On-Line	1	1	1	1	1	3	1
Evaluación Riesgo Salud	1	1	5	5	5	3	1
Intercambio de Datos	1	1	1	1	5	1	1
Privacidad	1	2	4	4	1	2	2
Facilidad de Navegación	3	4	4	3	4	3	4
Calidad Información Disponible	3	5	4	3	4	3	4
Presentación	3	4	4	3	4	3	4



<sup>3</sup> El ejemplo está disponible en la página web del grupo GIRO<sup>2</sup>: <http://giro.infor.uva.es>

**Fig. 3.** Ejemplo de aplicación de la técnica de la rejilla al dominio de las aplicaciones web de salud. Se muestra parte de la rejilla (el modelo original tiene 27 elementos) y las relaciones obtenidas para los softgoals más importantes.

### 4.3 Aplicación a i\*

Por último, es posible definir un proceso simple para elicitar modelos SD de i\* utilizando una rejilla. Para ello se crean dos rejillas, sobre la primera rejilla se aplica el análisis de Ford sobre los interesados para encontrar los agentes del modelo y en la segunda se buscan los agentes con alguna relación. Los pasos son:

1. Solicitar a los interesados una lista de sus objetivos (funcionales y no funcionales, aunque en este paso no se diferencian para no confundir a los interesados), recursos que necesitan y tareas que pueden llevar a cabo.
2. Reunir los elementos y a partir de ellos:
  - 2.1. Comprobar todos los elementos para encontrar sinónimos y escoger un nombre para cada concepto.
  - 2.2. Separar los objetivos funcionales de los no funcionales.
  - 2.3. Comprobar el tipo de los elementos, principalmente en el caso de las tareas y objetivos funcionales (similares debido a la naturaleza funcional de ambos).
  - 2.4. Crear la primera rejilla con los interesados como constructos y los elementos de i\* refinados en los pasos anteriores como elementos.
3. Obtener relaciones entre interesados a partir de la rejilla siguiendo las siguientes reglas:
  - *Si  $SH_x$  entonces  $SH_y$* : probablemente indica que  $SH_x$  es un rol del agente  $SH_y$ , puesto que indicará que los elementos que utiliza  $SH_x$  son un subconjunto de los que usa  $SH_y$ . Por ejemplo para:
 
$$SH_x: 110000000100$$

$$SH_y: 111100001100$$

$$SH_x \sqcap SH_y = 110000000100 = SH_x \rightarrow \text{Grado de confirmación } 1.$$
 Se obtiene la regla *Si  $SH_x$  entonces  $SH_y$  [1.0]* por tanto  $SH_x$  debe ser una parte de  $SH_y$ , es decir un rol de un agente (probablemente).
  - *Si  $SH_x$  entonces  $SH_y$  y Si  $SH_y$  entonces  $SH_x$* : utilizan los mismos recursos, luego lo más probable es que sean el mismo agente.
  - Obviamente, estos resultados son solo una guía para encontrar agentes, los agentes finales deben ser contrastados con la opinión de los interesados.
4. Con los interesados asignados a roles y agentes, seleccionar un interesado por cada rol y agente encontrado y solicitarle que rellene los datos de una rejilla como el del paso 2.4, pero simplificado a los actores / roles obtenidos (cada interesado rellena el actor / rol que le corresponda).
  - 1 si el interesado quiere alcanzar un objetivo / crea un recurso o quiere hacer una tarea (según sea el tipo de elemento).
  - 2 si el interesado no tiene ninguna relación con el elemento.
  - 3 si el interesado puede ayudar a alcanzar un objetivo / necesita un recurso o es necesario para realizar una tarea (según el tipo de elemento).

5. Analizar la rejilla. El análisis es diferente del realizado por Ford, en este caso simplemente hay que encontrar aquellos agentes (o roles) que para un elemento dado tienen un 1 y un 3. Es decir dependen de un elemento y son necesarios para ese elemento.
6. Crear el modelo SD a partir de las dependencias encontradas en el punto 5. Es importante señalar que la mayor parte de la información pertenecerá al modelo SD, pero aquellos objetivos y tareas que no pertenezcan a ninguna dependencia entre actores probablemente sean parte del modelo SR.

## 5 Conclusiones

En este artículo se han explorado varias posibles aplicaciones de una teoría psicológica, la TCP y de su principal herramienta, el *repertory grid* en la elicitación de información, centrada principalmente en enfoques orientados a objetivos.

Los principales resultados son la obtención de tres enfoques sistemáticos para obtener las relaciones de afinidad entre objetivos, para obtener las relaciones y softgoals de los modelos SIG del *NFR Framework* y las dependencias entre agentes para de los modelos SD de *i\**. Gracias a estas técnicas, esperamos facilitar la tarea de elicitación de estos modelos orientados a objetivos.

Es importante resaltar que poco se ha hecho en relación a la elicitación en modelos orientados a objetivos. La gran mayoría de las heurísticas son basadas en preguntas del tipo porqué, qué o cómo. Nuestra propuesta, a pesar de partir de objetivos, interesados y tareas, que son lo que queremos elicitar, utiliza la teoría de confirmación, basada en los *repertory grids*, para aclarar y descubrir otros objetivos, interesados y tareas a partir de un conjunto original. Los elementos iniciales o simientes se elicitan de la manera tradicional, mas la aplicación de la TCP permite profundar el conocimiento sobre el Universo de Discurso.

El siguiente paso es utilizar las estrategias en un entorno real. En el primer caso no es difícil para hacer una experimentación, que puede ser hecha con ejemplos de laboratorio. Pero el segundo caso (relaciones RNF) implica disponer de un conjunto grande de aplicaciones sobre las que realizar el estudio y, en el caso de *i\**, un grupo de interesados dispuestos a probar la técnica. En cualquier caso hemos empezado el estudio sobre el conjunto de aplicaciones sobre discapacitados que desarrollado por el Departamento de Informática de la Universidad de Valladolid.

Por último, sería interesante dar soporte a estas técnicas por medio de una herramienta. En estos momentos estamos trabajando en un primer prototipo sobre Excel y VBA que, a partir de una tabla donde está almacenado el *repertory grid*, nos devuelva las reglas del análisis propuesto por Ford que superen un cierto grado de confirmación. A partir de estas reglas, el siguiente paso es aplicar cualquiera de las dos técnicas mostradas. Este desarrollo tiene como base nuestra experiencia en el uso de esta infraestructura en la evaluación de requisitos [8].

## Reconocimientos

El primer autor de este artículo disfruta de una beca concedida por la Consejería de Educación de la Junta de Castilla y León y el Fondo Social Europeo.

## Referencias

- [1]. Antón, A.I., and Potts, C. “*The Use of Goals to Surface Requirements for Evolving Systems*”, in proc. of International Conference on Software Engineering (ICSE '98), Kyoto, Japan, April 1998, pp:157-166
- [2]. Chung, L., Nixon, B., Yu, E. and Mylopoulos, J. “*Non-Functional Requirements in Software Engineering*” Kluwer Academic Publishers 2000.
- [3]. Cysneiros, L.M., and Leite, J.C.S.P. “*Non-Functional Requirements: From Elicitation to Conceptual Model*” IEEE Transactions on Software Engineering 30(5). May 2004. pp:328-350.
- [4]. Dardenne, A., van Lamsweerde, A., “*Goal-Directed Requirements Acquisition*”, Science of Computer Programming, vol. 20, 1993, 179-190.
- [5]. Ford, K.M., Petry, F.E., Adams-Webber, J.R., and Chang, P.J. “*An Approach to Knowledge Acquisition Based on the Structure of Personal Construct Systems*”. IEEE Trans. Knowledge and Data Eng., 3(1). 1991 pp:78-88.
- [6]. Giorgini, P., Mylopoulos, J., Nicchiarelli, E., and Sebastián, R. “*Reasoning with goal models*” In Proceedings of the 21st International Conference on Conceptual Modeling (ER 2002), Tampere, Finland, October 2002.
- [7]. Goguen J., Linde C.. “*Techniques for Requirements Elicitation*”. Proc. of Requirements Engineering '93. Jan. 1993. San Diego, CA. Los Alamitos, CA: IEEE Computer Society, 1993.
- [8]. González-Baixauli, B., Leite J.C., and Mylopoulos, B. Visual Variability Analysis with Goal Models. Accepted in RE'2004.
- [9]. Kelly, G.A., *The Psychology of Personal Constructs*. New York: Norton, 1955.
- [10]. Leite, J.C.S.P., Freeman, P.A., “*Requirements Validation Through Viewpoint Resolution*” IEEE Transactions on Software Engineering: 17(12), 1991 pp:1253-1269
- [11]. Leite, J.C.S.P. et al. “*Enhancing a Requirements Baseline with Scenarios*” Requirements Engineering Journal, 2(4). 1997. pp:184-198.
- [12]. Mylopoulos, J., Chung, L., and Yu, “*From Object-Oriented to Goal-Oriented Requirements Analysis*”, Communications of the ACM, 42(1), Jan. 1999 pp:31-37.
- [13]. “*Personal Health Record Sites (PHR) Comparison*”. The Informatics Review. [http://www.informatics-review.com/thoughts/PHR\\_Site\\_Comparison.htm](http://www.informatics-review.com/thoughts/PHR_Site_Comparison.htm).
- [14]. van Lamsweerde, A. “*Goal-Oriented Requirements Engineering: A Guided Tour*”, Proc of the 5 IEEE Int. Symp. on Requirements Engineering, 2001, pp:249-262.
- [15]. Yu, E. “*Towards Modelling and Reasoning Support for Early-Phase Requirements Engineering*”, Proceedings of the 3rd IEEE Int. Symp. on Requirements Engineering (RE'97) Jan. 6-8, 1997, Washington D.C., USA. pp:226-235.
- [16]. Yu, E., Mylopoulos, J., “*Why goal-oriented requirements engineering*”, Proceedings of the Fourth International Workshop on Requirements Engineering: Foundations of Software Quality, Pisa, Italy, June 1998, pp. 15-22.
- [17]. Yu, E., “*Strategic Modelling for Enterprise Integration*”, Proceedings of the 14th World Congress of International Federation of Automatic Control (IFAC'99), Beijing, China, July 5-9, 1999, pp. 127-132.