



Universitat Ramon Llull

TESIS DOCTORAL

Título PROPUESTA DE UN MÉTODO DE VALIDACIÓN DE
ESQUEMAS CONCEPTUALES Y ANÁLISIS
COMPARATIVO DE LA NOCIÓN DE INFORMACIÓN EN
LOS MÉTODOS DE DESARROLLO DE SISTEMAS DE
INFORMACIÓN

Realizada por FELICIANO SESÉ MUNIÁTEGUI

en el Centro ESADE – ESCUELA SUPERIOR DE ADMINISTRACIÓN
Y DIRECCIÓN DE EMPRESAS.

Departamento POLÍTICA DE EMPRESA, RECURSOS HUMANOS Y
SISTEMAS DE INFORMACIÓN

Dirigida por DR. EDUARD BONET I GUINÓ

Índice

Índice	iii
Índice de diagramas	vii
Capítulo 1: Introducción.....	1
El valor estratégico de los sistemas de software.....	1
Riesgo de los proyectos informáticos	2
Delimitación del problema a tratar: Malentender los requerimientos.	4
La comunicación de los requerimientos y el problema de la información.	7
Génesis de este trabajo	13
La construcción de un programa genérico	14
La definición de una notación	15
La construcción de un compilador del lenguaje	17
La comparación con otros métodos de especificación de sistemas.....	17
El enfoque axiomático.....	18
La comparación con los presupuestos metafísicos de otros métodos .	20
Organización de este trabajo	21
Capítulo 2: Proyectos de sistemas de información	25
Marco del problema	27
Tres facetas del problema	28
La naturaleza del problema	31
Formalidad: información desinterpretada. La computadora seca.	31
La precisión y exactitud de la información.	33
El problema de la precisión y exactitud de la información	35
El procedimiento hermenéutico	38
Significado: Información subjetiva. La computadora húmeda.	41
Significación: Información intersubjetiva. El sistema social.	44

*Propuesta de un método de validación de esquemas conceptuales
y análisis comparativo de la noción de información
en los métodos de desarrollo de sistemas de información*

La propuesta de los modelos conceptuales	47
El problema de validación de esquemas conceptuales	52
Una aproximación al problema mediante formularios de empresa	57
El enfoque axiomático	59
Capítulo 3: Especificación de sistemas de información	63
El modelo de datos relacional.....	64
Presentación informal de la notación	67
Especificación de relaciones.....	67
Especificación de formas visuales básicas.....	70
Lista de formas visuales básicas.....	71
Especificación conceptual del sistema de información.....	72
Definiciones.....	73
Composición de forms: forms compuestos.....	74
Campos protegidos.....	78
Conexión de forms abierta.....	80
Operadores aritméticos y sinónimos.....	81
Atributos internos.....	83
Forms internos.....	84
Operadores de agregación.....	84
Campos sugeridos y ligaduras.....	86
Especialización de forms.....	89
Definición formal del lenguaje en notación BNF	91
Definición del vocabulario.....	91
Definición de la sintaxis.....	92
Vocabulario del cálculo.....	100
Gramática del cálculo.....	102
Definición de expresión.....	102
Definición de restricción.....	102
Definición de form.....	102
Precedencias de asociación de símbolos.....	102
Axiomas del cálculo	103
Reglas de inferencia del cálculo.....	104
Reglas Φ	105
Reglas Θ	105
Reglas σ	105
Reglas ρ	105
Reglas Ω	106
Capacidad expresiva	106
Valoración de la notación.....	116

Capítulo 4: La información como representación de la realidad.	119
Construcción del mapa: La dimensión ontológica.....	123
Los antiguos griegos	123
El debate medieval sobre los universales y el triángulo del significado de Ogden-Richards	129
El empirismo británico: Locke, Berkeley, Hume y Russell.....	131
Palabras y objetos. Distinción entre idealismo y nominalismo	135
Construcción del mapa: La dimensión epistemológica	137
Los distintos paradigmas en el área de sistemas de información.	140
Paradigma de Realismo Preceptivo (a priori): Ingeniería ontológica..	140
Paradigma Conceptualista Pragmático: Diseño de ontologías	142
Paradigma Conceptualista Preceptivo	143
Paradigma Realista Pragmático	146
Paradigma Nominalista Preceptivo: Métodos Formales	149
Paradigma Nominalista Pragmático	151
Conclusiones de este capítulo.....	154
Limitaciones y contribuciones del esquema propuesto	156
Capítulo 5: Distintas acepciones del término información	159
Información como soporte.....	162
Información como señal	162
La información como regularidad (pattern).....	164
La información como signo	168
La información como símbolo (referencia)	170
La información como significación (significado objetivo)	170
La información como sentido (significado subjetivo).....	172
La información como sentimiento.....	173
La información como síntoma.....	174
La información como intención comunicativa.....	175
La información como convención.	176
La información como acto de habla	180
La información como acto social.....	183
Conclusiones de este capítulo.....	185
Valoración	185
Limitaciones y contribuciones de este estudio.....	188

Capítulo 6: Conclusiones	191
Contribuciones a la discusión del problema.....	192
Contribuciones del método propuesto	194
Sumario de las principales contribuciones	196
Respecto al lenguaje de especificación propuesto.....	197
Respecto los problemas de especificación de requerimientos.....	197
Respecto al análisis de las nociones de información.....	198
Limitaciones.....	198
Relativas al ámbito de aplicabilidad.....	199
Relativas al lenguaje de especificación	199
Relativas al análisis de las distintas nociones de información.....	199
Futuras líneas de trabajo	200
Bibliografía	203

Índice de diagramas

<i>Diagrama 1.</i>	<i>Distintas formas visuales de los mismos datos.</i>	14
<i>Diagrama 2.</i>	<i>Cambio de la naturaleza de la información al atravesar los tres mundos.</i>	28
<i>Diagrama 3.</i>	<i>Concisión, exactitud y precisión.</i>	37
<i>Diagrama 4.</i>	<i>Ecuaciones de Maxwell.</i>	38
<i>Diagrama 5.</i>	<i>Concisión, exactitud, precisión y contexto.</i>	38
<i>Diagrama 6.</i>	<i>Bucle de elicitación embebido en el bucle de formulación.</i>	42
<i>Diagrama 7.</i>	<i>Flujos de información de sentido opuesto de los dos bucles.</i>	43
<i>Diagrama 8.</i>	<i>Contextualización de los esquemas conceptuales en los sistemas de información.</i>	49
<i>Diagrama 9.</i>	<i>Esquema conceptual expresado en notación Entity-Relationship.</i>	50
<i>Diagrama 10.</i>	<i>Secuencialidad del proceso de diseño de esquemas conceptuales.</i>	53
<i>Diagrama 11.</i>	<i>Triángulo del significado de Ogden-Richards</i>	130
<i>Diagrama 12.</i>	<i>Independencia de las dimensiones Idealismo y Nominalismo</i>	137
<i>Diagrama 13.</i>	<i>Posiciones ontológicas y epistemológicas en Desarrollo de Sistemas de Información.</i>	154
<i>Diagrama 14.</i>	<i>Resumen de las trece acepciones de la noción de información.</i>	188

Capítulo 1: Introducción

El valor estratégico de los sistemas de software

Los sistemas de software, las aplicaciones informáticas, se están convirtiendo en un elemento crítico para la competitividad de las empresas en un creciente número de sectores de actividad económica. Este tipo de afirmaciones pueden ser leídas en publicaciones de gestión de empresa, escritas por reconocidos especialistas del área. Por ejemplo, hay afirmaciones tan contundentes como las de Prahalad: *'The company's software applications are rapidly emerging as its central nervous system'* y *'A company's software is becoming a critical source of competitive advantage and competitive risk.'* (Prahalad and Krishnan, 1999) O las realizadas por Quinn: *'Software is and will be at the core of most innovation during the next several decades.'* (Quinn et al., 1996)

Pero estas aserciones no sólo son defendidas desde el mundo académico, algunos de los empresarios con mayor éxito también coinciden con ellas. Por la autoridad que les otorga el fulgurante y sostenido crecimiento de sus empresas, destacaremos a dos de ellos: Sam Walton y Michael Dell.

El caso del primero, fundador de la empresa Wal-Mart, que en poco más de cuatro décadas ha pasado de establecer su primer supermercado en un barracón de 1000 m² situado en la América rural, a ser la mayor empresa del mundo en facturación, que asciende a más de 300.000 millones de dólares en 2005; con unas tasas de crecimiento increíbles: en una sola década, la de los noventa, aumentó su plantilla en un millón de empleados. Partiendo de la base de 300.000 a finales de los años ochenta ha alcanzado a tener una plantilla de más de 1.300.000 empleados al comienzo de este siglo. Según el propio Walton, una de las principales razones de dicho crecimiento, no le cabe la menor duda, ha sido la

utilización inteligente de las nuevas tecnologías de la información. (Walton and Huey, 1992)

Obviamente, el desarrollo innovador de las nuevas aplicaciones de software no lo explica todo, pero sí parece ser cierto que, aunque no suficiente, es una condición necesaria para explicar el éxito de dicha empresa. (Johnson, 2002)

El segundo caso es el de la empresa Dell, fundada por Michael Dell hace menos de dos décadas, que empezó siendo la iniciativa de un estudiante ensamblando artesanalmente y vendiendo PCs clónicos. Esta empresa facturó 56.000 millones de dólares en 2005; en un sector, el de computadoras, en donde había un buen número de grandes y competitivas empresas. Michael Dell considera que *'Information is the key to any competitive advantage.'* (Dell and Fredman, 1999; p.117)

El ejemplo de estas dos empresas no es tan sorprendente si consideramos que la información constituye el elemento esencial de la coordinación, del control, de la toma de decisiones y del aprendizaje de las empresas, que son las bases no humanas en que se asientan la eficiencia y la eficacia de las organizaciones. Y que los sistemas de software hacen posible, precisamente, que la información sea capturada, procesada, transmitida y distribuida casi instantáneamente, lo que la convierte en ubicua, aumentando radicalmente el espacio de posibilidades de nuevos diseños organizativos, que ahora ya sólo están limitados por el factor humano y la imaginación de los diseñadores de estas nuevas organizaciones. Pero la producción de software, que hace factible muchas innovaciones en la empresa, presenta una serie de nuevos problemas.

Riesgo de los proyectos informáticos

Como señala Prahalad, los sistemas de software no son sólo una fuente de ventajas competitivas, también son el origen de riesgos, que se pueden clasificar en dos grandes grupos, riesgos por pasividad y por actividad. Los riesgos por pasividad se pueden producir al no tomar la iniciativa en las innovaciones o al menos imitar rápidamente al líder innovador. Tal es el caso de Kmart, que fue la competencia directa de Wal-Mart y el modelo de negocio en que se inspiró Sam Walton para establecer su empresa, pero que, como el propio Walton dice en su libro,

tuvo unos directivos renuentes al cambio. Este falta de iniciativa de Kmart en la innovación en tecnología de la información la sintetiza (Nannery, 2002) con un juego de palabras en el título de su artículo 'Use I.T. or lose it'.

Los riesgos por activa (iniciativa), por inhabilidad para llevar a buen término los proyectos de innovación basados en nuevos sistemas de software. Kmart, aunque la innovación que pretendía hacer tenía un retraso de más de una década respecto a la realizada por Wal-Mart, también incurrió en este tipo de riesgos. Coincidió la suspensión de pagos presentada por la empresa con el fracaso de su último proyecto para mejorar la gestión de sus almacenes basado en nuevos sistemas de software, cuyo coste ascendía a 130 millones de dólares (Konicki, 2002).

Hay casos en que el fracaso de un proyecto de software ha llevado directamente a la bancarrota de la empresa. Como el de la empresa Foxmeyer, que a principio de los noventa era en volumen de facturación la cuarta mayorista de productos farmacéuticos de Estados Unidos. La incorrecta ejecución del proyecto de mejora de gestión de almacenes, junto a la mala previsión de las eventualidades, llevó a la empresa directamente a la bancarrota y a ser comprada por un competidor por un importe inferior al que había costado el proyecto (Jesitus, 1997; Scott, ; Osterland, 2000; Scott and Vessey, 2002). El último ejemplo que citaremos, también espectacular, es el del nuevo aeropuerto de Denver, en el que los fallos del software en el sistema informático de control de equipajes impidieron la puesta en marcha del aeropuerto durante meses, lo que unido a unos costes financieros y de funcionamiento de más de un millón de dólares diarios sin poder poner en funcionamiento el aeropuerto y obtener los ingresos correspondientes, llevaron a la quiebra de la empresa (Gibbs, 1994).

Más allá de estas narrativas y anécdotas presentadas, lo alarmante es que la tasa de fracasos en proyectos informáticos es realmente elevada. No abundan las investigaciones sobre estos hechos pues es un ámbito de estudio especialmente opaco a los investigadores; esto es debido a que muchas empresas rehuyen comunicar sus fracasos para evitar la publicidad negativa que supondría el conocimiento público de los mismos. Según un estudio realizado por Standish Group (Keil et al., 1998; Standish Group, 1995), en el año 1995, en Estados Unidos, el coste de los proyectos de software ascendió a unos 250.000 millones de

dólares, de los cuales 59.000 millones fueron debidos al sobrecoste de algunos proyectos y 81.000 millones en proyectos cancelados. Su investigación muestra que el 31,1% de los proyectos son cancelados antes de ser acabados y un 52,7% de los mismos cuestan un 189% más de lo presupuestado, lo que significa que sólo el 16,2% de los proyectos finalizan a tiempo y con el importe presupuestado.

Delimitación del problema a tratar: Malentender los requerimientos.

Esta tasa de fracasos, incomparablemente superior a la de otras ingenierías, que Gibbs califica en el título del artículo citado anteriormente como 'La crisis crónica del software', llevan a preguntarse cuáles son las razones de la misma. Brooks, desde su experiencia como jefe del proyecto IBM OS/360, que llevó a IBM a la supremacía mundial en su sector durante las décadas de los 70 y 80, la imputa a la gran complejidad de la ingeniería del software, que considera notablemente superior a la de las ingenierías tradicionales. El origen de esta complejidad la atribuye al carácter abstracto del software.

A diferencia de las otras ingenierías que diseñan y construyen máquinas materiales, el software es inmaterial, lo que conlleva que, al no estar restringido por las leyes físicas, las posibilidades de diseño aumentan considerablemente, no tienen otro límite que la imaginación humana, lo que en ocasiones puede llevar a que los diseños sean de gran complejidad. Otra fuente de problemas, también debido al carácter abstracto del software, es su invisibilidad, lo que no sólo impide el uso de una de las más poderosas herramientas conceptuales de que dispone el ser humano, la imagen, sino que también restringe severamente la comunicabilidad entre mentes. (Brooks, 1986)

Existe un gran número de estudios analizando las posibles causas de los problemas, los riesgos de los proyectos de software. El origen de estos riesgos tiene multitud de causas (Alter and Sherer, 2004; Sherer and Alter, 2004), sin embargo estudios empíricos basados en las experiencias y percepciones de directivos y jefes de proyectos de sistemas de información destacan tres de ellos como los más importantes: malentender los requerimientos (misunderstanding requirements), la falta de apoyo y dedicación de la alta dirección al proyecto y la no

involucración de los usuarios en el mismo. El orden de importancia de estos tres factores varía de un estudio a otro e incluso depende del país en donde se realice. (Schmidt et al., 2001; Keil et al., 1998; Standish Group, 1995).

Es una verdad necesaria que si se entienden equivocadamente los requerimientos, si se hacen mal los planos del sistema a construir, se construye el sistema equivocado. Lo relevante de estos estudios es que, según los directivos y jefes de proyecto encuestados, este hecho se produce con gran frecuencia. Algunos estudios revelan que el 56% de los errores detectados en los sistemas de software tienen su origen en la especificación deficiente de los requerimientos del sistema a construir (Davis, 1993).

Estas evidencias son consistentes con la intuición de Brooks sobre el hecho de que el carácter abstracto del software dificulta la comunicabilidad entre las personas, lo que explica no sólo que se produzcan malentendidos en la especificación de requerimientos, sino también los otros dos riesgos: la renuencia de directivos y usuarios a participar e involucrarse en proyectos que no son capaces de entender en sus primeras fases.

Un hecho que agrava el problema de los requerimientos es el coste (esfuerzo) necesario para reparar los errores cometidos. Distintos estudios empíricos estiman que si el coste de solventar un error cuando se detecta en el momento de fijar los requerimientos es 1, si es detectado en el momento de instalar el sistema acabado es de 50, y si se descubre cuando el sistema ya está operativo el esfuerzo para repararlo llega a 200 (Davis, 1993).

Podemos resumir lo dicho anteriormente con las palabras de Brooks:

The hardest single part of building a software system is deciding precisely what to build. No other part of the conceptual work is so difficult as establishing the detailed technical requirements, including all the interfaces to people, to machines, and to other software systems. No other part of the work so cripples the resulting system if done wrong. No other part is more difficult to rectify later.

Therefore the most important function that software builders do for their clients is the iterative extraction and refinement of the product requirements. For the truth is, the clients do not know what they want. They usually do not

know what questions must be answered, and they almost never have thought of the problem in the detail that must be specified. (Brooks, 1986; p.199)

Un hecho que agrava la posibilidad de malentendidos en el establecimiento de los requerimientos se origina en su raíz, empieza en la ambigüedad del propio término ‘requerimiento’. Según la definición estándar de la IEEE Standard Glossary of Software Engineering Terminology (IEEE, 1990), un requerimiento es:

1. *A condition or capacity needed by a user to solve a problem or achieve an objective.*
2. *A condition or capability that must be met or possessed by a system or system component to satisfy a contract, standard, specification, or other formally imposed documents.*
3. *A documented representation of a condition or capability as in (1) or (2).*

En esta definición caben muchas maneras de entender lo que es un requerimiento. Raghavan et al. (1994) y McConnell (2000) muestran su ambigüedad utilizando ejemplos que se ajustan a ella:

- 1- Escriba un programa que solicite n números y los muestre en pantalla presentados en orden decreciente.
- 2- Desarrolle el software necesario para que el avión modelo X pueda aterrizar en los principales aeropuertos automáticamente sin intervención del piloto.
- 3- Escriba un programa que reciba el importe de las compras anuales de todos los clientes de la empresa y los muestre en pantalla presentados en orden decreciente.
- 4- Desarrolle un sistema automatizado de gestión de pedidos que permita procesarlos en menos de 12 horas.
- 5- Cree un sistema informático que permita incrementar en un 10% la cuota de mercado de la empresa.

De esta pequeña muestra se pueden destacar tres dimensiones de ambigüedad: por el ámbito de aplicación, por el nivel de concreción y por el nivel de abstracción.

El ámbito de aplicación de los sistemas de software es cada vez mayor. Desde productos como videojuegos o procesadores de textos,

pasando por sistemas de control de máquinas, como motores de coche, o interacción entre máquinas, como el caso del segundo ejemplo de requerimiento, o sistemas que procesan y transmiten información en las empresas y organizaciones, como muestran los otros cuatro ejemplos presentados. Este último ámbito de aplicación del software es normalmente calificado de software para sistemas de información.

En los ejemplos también se puede observar los distintos niveles de concreción. El último, planteado como un objetivo estratégico de empresa puede requerir como ‘condición o capacidad’ que se cumpla el anterior, agilizar la gestión de pedidos; que, a su vez, actúa como objetivo para otra serie de condiciones o capacidades, entre las que podría figurar la tercera, un programa que facilite mediante la información tomar decisiones adecuadas. Que, cambiando el nivel de abstracción y formulándolo de manera desinterpretada, correspondería al primer requerimiento del ejemplo.

La comunicación de los requerimientos y el problema de la información.

Aunque algunos de los errores cabe atribuirlos a causas accidentales, la gran tasa de los mismos lleva pensar que el problema es sistemático, que existe un defecto de origen en la comunicación que se establece entre analistas, usuarios y directivos.

Como respuesta a estos problemas, en los últimos años hemos asistido a una exuberante proliferación de métodos para analizar y establecer los requerimientos de los sistemas que se han de construir. Hace treinta años no existía prácticamente ninguno, mientras que en 1994, según una estimación citada por Avison y Fitzgerald (1995), el número de métodos con nombre de marca estaba alrededor del millar; dando lugar a un nuevo problema, que dichos autores llaman la ‘jungla de las metodologías’.

Frente a la situación de un ingeniero de caminos cuyo trabajo consiste en aplicar una serie de principios científicos asumidos por todos sus compañeros de profesión, un ingeniero de software, o un analista de sistemas de información, se enfrenta a una situación distinta, no existe un conjunto de principios compartidos por todos. Por ello tiene un problema previo, intentar discriminar el método más adecuado entre tan

gran número de propuestas de naturaleza muy diversa y, en algunos casos, contradictoria.

Esta situación parece ser debida a que mientras que el resto de las ingenierías se fundamentan en leyes de la naturaleza, que están explicadas por sólidos principios científicos, la ingeniería del software, aunque en su vertiente de construcción, esto es, en las actividades de programación y de diseño técnico, está sólidamente asentada en la lógica y las matemáticas, en su parte de ingeniería de requerimientos, está basada en una noción, la información, que está lejos de estar bien asentada y fundamentada científicamente. Este problema se agrava cuando los sistemas de software que se construyen son para sistemas de información, esto es, para construir máquinas que permitan la transmisión de información, la comunicación, entre seres humanos.

Cuando la ciencia no da respuestas parece natural acudir a la filosofía. Por ejemplo, si el problema es el de malentender los requerimientos, que son fallos en la transmisión de la información entre directivos, usuarios, analistas y programadores, y si consideramos que la hermenéutica, tal como la define Schleiermacher, es el ‘arte de evitar los malentendidos’, puede parecer natural acudir a esta tradición filosófica para intentar fundamentar los métodos de análisis de requerimientos.

En las dos últimas décadas hemos asistido a la irrupción de la filosofía en la disciplina de sistemas de información. Tiene una presencia cada vez más importante en la disciplina en un intento de fundamentar los distintos métodos y programas de investigación. Esto responde a un hecho, a pesar de proclamar que vivimos en la sociedad de la información o que nos dedicamos a la disciplina de sistemas de información, no sabemos lo que es información.

A modo de ejemplo de esta irrupción de la filosofía en el área de sistemas de información, baste una pequeña muestra. El trabajo basado en Hegel de Chae y Courtney (2000), los fundamentados en Kant de Monod (2004; 2003; 2002) y Mahfouz and Paradice (2000), en la fenomenología de Heidegger de Winograd y Flores (1986), Boland Jr. y Day (1982) y Boland Jr. (1985); en la hermenéutica de Gadamer, en (Boland Jr., 1991) y (Klein and Myers, 1999; Myers, 2004), en la ontología de Mario Bunge, en (Rosemann et al., 2004 ; Wand and Weber, 1988), en el realismo crítico de Bhaskar, en (Mingers, 2004; Mingers, 2005). Por no citar los numerosos trabajos basados en la filosofía de los

actos de habla (speech acts) de los filósofos Austin y Searle, o en la teoría de la estructuración de Giddens, o de la acción comunicativa de Habermas. O los basados en los deconstruccionistas franceses Paul de Man y Jacques Derrida, como el trabajo de Beath y Orlikowski (1994).

Este creciente interés se ha visto reflejado en los principales congresos internacionales (como ICIS, AMCIS, ECIS, PACIS), que han establecido sus correspondientes sesiones ('tracks') sobre 'filosofía y sistemas de información', en los que los investigadores hacen público y discuten sus trabajos.

Recurrir a la filosofía para intentar salir de la perplejidad que puede producir la contemplación de la jungla de las metodologías de desarrollo de sistemas de información puede agravar la divergencia de teorías, alejándonos todavía más del desarrollo acumulativo propio de las ciencias que Kant ya propugnaba para la metafísica '*...it has not yet had the good fortune to attain to the sure scientific method. ...We find, too, that those who are engaged in metaphysical pursuits are far from being able to agree among themselves*' (Kant, 1781a; p.14). Convergencia que Benbasat y Zmud defienden para nuestra disciplina: '*it is imperative that the IS research community produce cumulative, theory-based, context-rich bodies of research*' (Benbasat and Zmud, 1999; p.9).

A juzgar por la evolución posterior de la filosofía, el empeño de Kant no tuvo fortuna, tal como muestra el filósofo contemporáneo Richard Rorty: '*La historia de la filosofía está demarcada por los filósofos que han pretendido, desautorizando a los filósofos precedentes, asentarla en bases firmes y seguras para transformarla en una ciencia. Por ejemplo, el método de las 'ideas claras y seguras' de Descartes, el 'transcendental' de Kant, el de 'reducción' de Husserl, el del primer Wittgenstein de disolver los problemas filosóficos anteriores estudiando su forma lógica y mostrando que carecían de sentido, o el del segundo Wittgenstein, que rebatió sus tesis anteriores analizando el lenguaje.*' (Rorty, 1967; p.47)

De hecho coincidimos con lo que sugiere el título de la mesa redonda propuesta por Lyytinen (2003) en la Americas Conference on Information Systems de 2003 'Information Systems and Philosophy: The Hopeless Search for Ultimate Foundations.' Sin embargo, pensamos que el estudio de la filosofía a lo largo de la historia aunque no solucione los problemas actuales al menos puede ayudar a aclararlos y delimitarlos. Tal como dice el filósofo Daniel Dennett (1987; p.15) '*La filosofía no produce a menudo resultados firmes y dignos de confianza como lo hace la ciencia en sus mejores*

momentos. Puede no obstante, producir nuevas maneras de ver las cosas, de pensar en ellas, de formular las preguntas, y de ver qué es lo importante y por qué.

Utilizada con este propósito la filosofía es extraordinariamente rica en propuestas. Aunque la noción de información tradicionalmente ha sido considerada poco importante por los filósofos (Coffa, 1991), la noción de conocimiento es central en la disciplina. Podemos sacar partido de ésta si tenemos en cuenta que hay un considerable solape entre las nociones de información y conocimiento. Los conceptos, ya se les llame ideas, universales o términos generales, se utilizan para ‘transportar’ información y también son el ‘repositorio’ de los conocimientos. Además, en el pasado siglo, la filosofía analítica situó el significado de las palabras en el núcleo del trabajo filosófico. Lo cual, si tenemos en cuenta que *‘Where metaphysics had sought the essence of things, analytical philosophy...settled for the meanings of words.’* (Quine, 1990; p.55), establece un continuo con la filosofía de los siglos anteriores.

Además, permite explicitar distintos presupuestos, normalmente no explicitados por los autores de sistemas de información, que sin embargo han sido largamente debatidos en filosofía durante toda su historia.

Un estudio basado en problemas metafísicos puede parecer a primera vista un mero divertimento teórico, simple especulación, algo completamente alejado de la realidad práctica y del día a día en la construcción de sistemas de información. Nada más lejos de la realidad, a nuestro parecer. El propósito de este trabajo es eminentemente práctico. Conocer cuáles son los presupuestos metafísicos de las distintas metodologías de sistemas de información y de cuál es la naturaleza del problema al que la vamos a aplicar puede ayudar a decidir cuál es la más idónea. Utilizar una metodología en un proyecto, o realizar una investigación, basada en un paradigma inadecuado al tipo de problemas que se pretende resolver puede llevar a malgastar una gran cantidad de recursos. Tal como lo expresa el premio Nobel de física Murray Gell-Mann (1994) con una frase que los matemáticos de principios del siglo XX atribuyeron a David Hilbert y que ha sido repetida multitud de veces: No hay nada más práctico que una buena teoría.

Creemos que una visión de conjunto de las distintas concepciones acerca del término ‘información’ que subyace en los diversos paradigmas y programas de investigación en el área de sistemas de información es relevante en esta época de necesaria especialización de la investigación.

La especialización conlleva al indeseable efecto de la miopía, que los investigadores concentrados en un determinado dominio de problemas para el que es adecuado determinado paradigma tiendan a considerar como apropiados otros grupos de problemas, cuyos fenómenos son de naturaleza distinta (como reza el proverbio inglés: *“If the only tool you have is a hammer, you will see every problem as a nail.”*) Esto es mucho más frecuente en el área de sistemas de información que en las ciencias naturales. En ciencias naturales el ámbito de aplicabilidad de las distintas teorías es bastante claro, por ejemplo, es evidente la escasa utilidad de la teoría electromagnética de Maxwell en el ámbito de la predicción del tiempo atmosférico.

En el área de sistemas de información, debido a la indefinición y gran ambigüedad de su término básico ‘información’, no sólo es muy grande la posibilidad de malentendidos (*‘misunderstandings’*) sino también de falacias (presentar argumentos que parecen plausibles mediante inferencias falsas), o concepciones falsas (*‘misconceptions’*), e incluso de mendacidades, entendiendo éstas en el sentido que le da Gadamer (1986; p.179) de ‘pérdida de sentido de la verdad y de la verdad en general’.

Sin embargo, reconociendo las desventajas de la especialización, su contrario, un estudio que abarque distintas áreas de especialización, también conlleva otros efectos no menos indeseables. De la miopía se puede caer en la hipermetropía, en la pérdida de rigor y precisión. Esta pérdida de rigor tiene su origen en distintas causas. La primera es el amplio abanico de filósofos consultados para realizar este estudio y la diversidad de campos de investigación que aún girando entorno a la disciplina de análisis y diseño de sistemas de información pertenecen a distintos ámbitos de investigación, como son: la ingeniería ontológica, los métodos formales de especificación de sistemas, la ingeniería de requerimientos, los métodos de modelado conceptual, la teoría de normalización de bases de datos y los métodos de análisis orientados a objetos.

En este trabajo, esta posible pérdida de rigor, debida al intento de abarcar diversos campos, se ha intentado paliar especializando el estudio en una sola noción, la de información. Y, más concretamente, a la noción de información como representación de los hechos y cosas que ocurren en el mundo. La información como representación transmitida con conceptos, ya se le llame, conceptos, ideas, universales o términos

generales, tiene una larga historia de debates en filosofía. Desde el enfrentamiento entre Aristóteles y Platón, y a su vez éstos con los sofistas; pasando por lo que se conoce como la ‘batalla de los universales’ en la Edad Media y por las disputas entre los empiristas ingleses, y las de éstos con los racionalistas continentales. Llegando hasta nuestros días, con las distintas posturas de los filósofos analíticos y del lenguaje, como Quine, Putnam, Austin o Searle. O como los filósofos actuales de orientación más metafísica, como Barry Smith, Mario Bunge o Roy Bhaskar, que aunque no tienen la notoriedad de los anteriores, han sido empleados por distintos autores del área de sistemas de información para fundamentar sus posiciones.

Lo sorprendente es que en bastantes casos dentro de una misma corriente filosófica encontramos las distintas posiciones enfrentadas. Por ejemplo, entre los empiristas británicos está la posición conceptualista de Locke, la nominalista de Berkeley y Hume, y la realista platónica de Bertrand Russell. O en el ámbito de la fenomenología, está la postura platónica de Husserl frente a la más nominalista de Gadamer.

Dentro del campo de los sistemas de información este estudio filosófico sería obviamente superfluo si no fuera porque el mismo problema, el mismo abanico de posiciones, se presenta actualmente en la disciplina. Distintos investigadores del área de sistemas de información, por ejemplo Tim Berners-Lee, Nicola Guarino, Ron Stamper o los creadores del método de análisis y diseño de sistemas UML, Grady Booch, James Rumbaugh y Ivar Jacobson, basan la viabilidad de sus respectivos proyectos en premisas discrepantes sobre las características de los conceptos, con el agravante de que generalmente no las explicitan ni las argumentan. Estos supuestos presentan la misma diversidad y divergencia que los observados recurrentemente en filosofía a lo largo de su historia, con la diferencia que en ésta sí que se han debatido explícitamente aunque desgraciadamente no se haya llegado a ningún tipo de consenso.

Berners-Lee, el inventor de la World Wide Web, propuso una extensión de la misma, la Web Semántica, que supondría un avance espectacular sobre la primera. Su propuesta se basa en dar a la información que alberga la web un ‘*well-defined meaning*’ de modo que los ordenadores puedan actuar de un modo que sería equivalente a entender el significado de las palabras que albergan en sí mismos (Berners-Lee et

al., 2001). Esta postura está próxima, por lo menos, al conceptualismo de Locke.

Una posición más extrema la mantiene alguno de los miembros del programa de investigación denominado Ingeniería Ontológica (hace algunos años hubiera sido considerado un término altamente improbable la unión conceptual de la parte más pragmática de la ciencia, como es la ingeniería, con la rama más metafísica de la metafísica, como es la ontología). Esta nueva disciplina ha producido en los últimos años multitud de artículos en revistas de prestigio y congresos especializados. Se aproxima a un enfoque platónico, pretende obtener *'the definitive and exhaustive classification of entities in all spheres of being. It can thus be conceived as a kind of generalized chemistry.'* (Smith and Welty, 2001) Obviamente, si alcanzasen sus objetivos solucionarían el problema de la Web Semántica además de un gran número de problemas del área de sistemas de información. Por la otra parte, las posiciones de los autores del método UML son más atemperadas y se asemejan a la de Aristóteles, mientras que la de Stamper está muy próxima al nominalismo.

Génesis de este trabajo

Este estudio se inició en un ámbito completamente ajeno a consideraciones filosóficas. Empezó como un problema de programación al observar que en las aplicaciones informáticas de empresa, un buen número de programas son similares. Los programas que se han de hacer para mantener al día el archivo de productos de la empresa, son estructuralmente idénticos a los que se hacen para realizar la misma función con el fichero de clientes, el de empleados, o el de proveedores.

En el diagrama 1 representamos una pantalla y un archivo de ordenador para mostrar las acciones que realiza un ordenador cuando ejecuta un programa que permite hacer el mantenimiento de un archivo de empresa, por ejemplo, de productos.

El funcionamiento básico de un programa para realizar las altas de productos consiste en permitir que el usuario rellene los campos correspondientes de la pantalla y que cuando éste pulse la tecla 'Alta' coja el texto de cada campo de la pantalla y lo guarde en el archivo de disco en el campo correspondiente. Si en vez de dar de alta los productos, lo

que queremos es dar de alta los clientes, lo único que cambia es, por una parte, los textos que aparecen en la pantalla y por la otra, el nombre del archivo de disco en donde se almacenarán los datos, así como el de los distintos campos en que se almacenará cada atributo.

Observado que todos los programas de este tipo tienen una estructura idéntica, se construyó un único programa que, después de solicitar el archivo que se deseaba actualizar, leía las características del mismo y generaba la pantalla correspondiente para poder realizar la actualización.

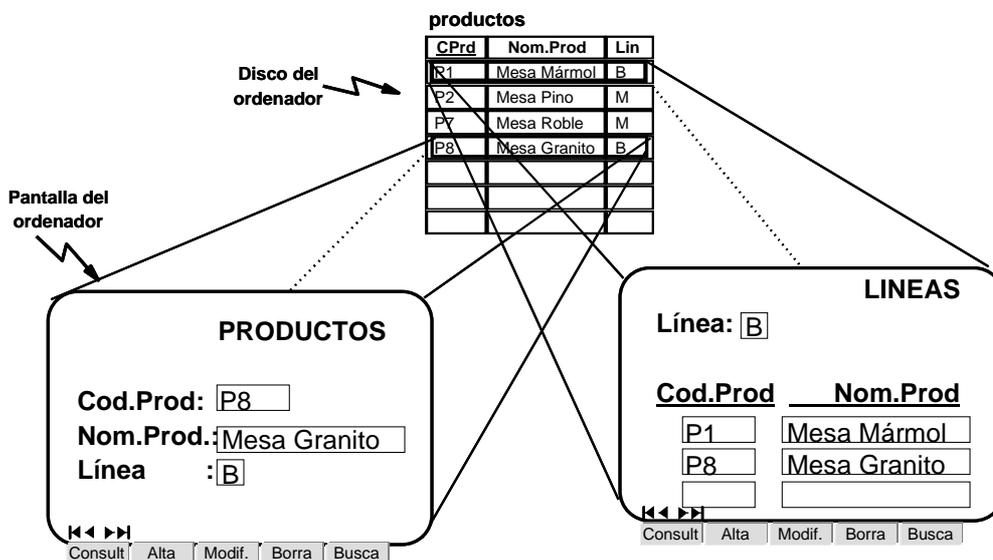


Diagrama 1. Distintas formas visuales de los mismos datos.

La construcción de un programa genérico

El siguiente paso fue ampliar el generador de programas para que pudiera tratar formas visuales (maneras de presentar los datos en pantalla) distintas. Por ejemplo, la que se presenta en el diagrama 1 a la derecha.

Vemos en el archivo de productos que cada producto tiene un código, que lo identifica, un nombre y está clasificado con el atributo 'Lin' que lo asigna a un tipo de producto. Una 'B' significa que se asigna a la línea 'Blanca' (por ejemplo, muebles de cocina) y una 'M', a la línea 'Marrón' (muebles de comedor). En este caso, en que para cada valor del atributo

‘Lin’ pueden haber varios productos, la forma visual adecuada es la que se muestra en la pantalla ‘LINEAS’.

Entre los valores de los atributos de un archivo sólo existen tres tipos de correspondencia, 1:1, 1:n y n:m. En nuestro ejemplo, el tipo 1:1 significa que a cada código de producto le corresponde un único valor de línea de producto y viceversa sólo hay un producto de cada línea. El 1:n, indica que, en el caso del ejemplo, cada producto se asigna a una sola línea, pero una línea puede tener varios productos. El tipo n:m se produce cuando un producto puede ser asignado a distintas líneas. Aunque este tipo es representable en forma de matriz, es poco frecuente verlo en los documentos que se utilizan para hacer transacciones empresariales, tanto con el exterior de la empresa, por ejemplo las facturas, como en las operaciones internas, por ejemplo un formulario para comunicar los gastos de viaje. Cuando se presentan en pantalla los datos, este tipo es reducible a dos presentaciones de la forma visual 1:n. Para el caso n:m, en el ejemplo del diagrama 1, la forma visual de la derecha se mantendría y la de la izquierda debería modificarse para permitir adjudicar varias líneas a un mismo producto. Pero en este caso la pantalla sería más compleja, habría un campo, el nombre del producto, que tendría una correspondencia 1:1 con el código de producto, y otro, la línea, con una correspondencia 1:n.

En este punto se observó que las formas visuales complejas, como las que tienen los documentos que se utilizan para realizar transacciones de empresa, se pueden reducir a la superposición de las dos formas visuales básicas, la 1:1 y la 1:n.

La definición de una notación

Mientras se ampliaba el programa para que contemplase estos casos, movidos por consideraciones de simetría y de simplicidad, se ideó una notación para expresar la estructura de las pantallas a partir de la especificación de los archivos. Expresando lo anterior mediante una notación funcional, escribiremos $\mathbf{a} \rightarrow \mathbf{b}$ para indicar que a cada valor de \mathbf{a} le corresponde uno sólo de \mathbf{b} , y $\mathbf{a} \rightarrow \gg \mathbf{b}$ para representar el caso de que a cada valor de \mathbf{a} le pueden corresponder varios de \mathbf{b} . Con esta notación, la correspondencia 1:1 entre \mathbf{a} y \mathbf{b} se expresa: $\mathbf{a} \rightarrow \mathbf{b}$ y $\mathbf{b} \rightarrow \mathbf{a}$, la 1:n como $\mathbf{a} \rightarrow \gg \mathbf{b}$ (esto indica que entre los dos atributos \mathbf{a} y \mathbf{b} existe una

correspondencia que no es función matemática). Se entiende que si $\mathbf{a} \rightarrow \mathbf{b}$ (esto es, si \mathbf{a} y \mathbf{b} están relacionados) y **no** $\mathbf{b} \rightarrow \mathbf{a}$ entonces $\mathbf{b} \rightarrow \mathbf{a}$.

Con esta notación se pueden expresar las dos pantallas del diagrama 1. Dado el archivo productos(CPrd, NomProd, Lin), las pantallas se expresarían:

productos {CPrd \rightarrow {NomProd, Lin}}

lineas { Lin \rightarrow {CPrd, NomProd}}

La segunda pantalla es deducible de la primera. En efecto, la primera pantalla indica que los tres atributos están relacionados y establece una única restricción (que a cada valor de CPrd le corresponde uno sólo de NomProd y de Lin). En ausencia de cualquier otra restricción se infiere que a cada valor de Lin le corresponden varios de CPrd y NomProd.

Aunque no es necesario, para expresar la correspondencias y las restricciones entre atributos, utilizamos, con alguna pequeña variante, la notación del método relacional (Codd, 1970). La razón de emplear esta notación, es, en primer lugar, porque es más compacta, y en segundo lugar, porque en la actualidad es la más utilizada tanto en el mundo académico como en el de los productos comerciales de sistemas de gestión de bases de datos.

Según esta notación el archivo de productos se expresa como productos(CPrd, NomProd, Lin) y las restricciones, las funciones matemáticas entre atributos, se expresan indicando los atributos que forman una 'clave'. Se dice que un atributo (o un conjunto de atributos) de un archivo es clave si sus valores no pueden estar repetidos en el archivo. El concepto de clave implica el de función y, tal como se demuestra en el método relacional, es posible expresar (capturar) todas las funciones entre atributos utilizando únicamente las claves. Este procedimiento de diseño de los archivos para que capturen todas las funciones entre atributos se conoce como normalización. El conjunto de archivos normalizados (una vez normalizados se les llama relaciones) se conoce como el esquema conceptual.

La especificación del sistema de nuestro ejemplo quedaría:

ESQUEMA [productos(CPrd, NomProd, Lin) : K(CPrd)]

FORMS [

productos {CPrd → {NomProd, Lin}}

lineas { Lin →> {CPrd, NomProd}}

]

En esta especificación entre los dos primeros corchetes se expresa el esquema conceptual, el conjunto de archivos normalizados (relaciones) que constituyen la base de datos de la aplicación informática que queremos construir. La K(CPrd) indica que el atributo CPrd es la clave (Key) de la relación.

Entre los dos segundos corchetes se expresa los formularios de empresa que queremos utilizar para consultar y mantener los archivos. La palabra FORMS es una abreviación de formularios y coincide con el nombre en inglés que expresa el mismo concepto.

La construcción de un compilador del lenguaje

Utilizando el programa Yacc de construcción de compiladores (Kernighan and Pike, 1987; Levine et al., 1995), se construyó un compilador que crea los archivos y las pantallas especificados, y que traduce la especificación a código ejecutable por el ordenador. De tal modo que para cada formulario se pueden realizar las acciones de consulta, alta, modificación, borrado y búsqueda de registros, además de poder ir al siguiente, anterior, primero o último registro del archivo.

El sistema que se desarrolló tiene demasiadas limitaciones como para ser utilizado en la práctica, en la construcción automática de programas de empresa. Sin embargo, creemos que tiene utilidad práctica como herramienta de especificación formal de sistemas de información y como herramienta de validación de esquemas conceptuales. En este trabajo se defiende la notación propuesta a nivel de prueba de concepto, como una vía a investigar que permita atenuar el grave problema de los malentendidos en la especificación de los requerimientos de sistemas de información.

La comparación con otros métodos de especificación de sistemas

En este punto, se revisó la bibliografía científica para poder comparar esta notación con otras propuestas. Considerando que el formulario de empresa es el inicio y el fin de una gran parte de los sistemas de información para la empresa, el número de artículos que utilizan este

enfoque al problema es muy pequeño. En ninguno de estos artículos hemos encontrado una propuesta de notación para especificar los sistemas.

La notación que proponemos presenta a nuestro juicio dos ventajas. Puede reducir los malentendidos que se producen entre analistas y programadores y también los que ocurren entre analistas y usuarios. El mejor modo de reducir los malentendidos del primer caso es mediante los métodos de especificación formal. Esta es un área de investigación que generalmente se basa en la utilización de la lógica matemática para la especificación de sistemas. Pero esta aproximación al problema tiene un defecto, que las especificaciones son poco concisas y difícilmente entendibles por los usuarios, que son los que tienen que validarlas antes de que el sistema se empiece a construir. Esto es, no tiene viabilidad para reducir los malentendidos entre usuarios y analistas (Loucopoulos and Karakostas, 1995). El método que aquí se propone es una especificación formal del sistema de información a construir, y al mismo tiempo, dado que las fórmulas se pueden visualizar como formularios de empresa, los usuarios pueden comprenderlas más fácilmente y por tanto validarlas.

Otra vía de investigación que tiene una gran actividad desde hace varias décadas es la conocida como Modelado conceptual de sistemas de información. El propósito de esta área de investigación es el desarrollo de notaciones y métodos para la obtención de esquemas conceptuales correctos, sobre los que se puedan construir las aplicaciones del sistema de información. Tiene la ventaja, frente a otros métodos, de que su notación es concisa, pero su desventaja es que sólo modela la mitad del mundo, los datos, pero no las actividades de actualización de los mismos (Winslett, 2004; Coad and Yourdon, 1991). Esto hace que las especificaciones no sean completamente formales. Además, en la mitad del mundo que modelan, el esquema conceptual, soslayan el problema de la validación del mismo por parte del usuario (Shanks et al., 2003).

El enfoque axiomático

Aunque el método que se propone en este trabajo es heredero, es una continuación, de los métodos de modelado conceptual, la diferencia entre ambos puede explicarse en términos de axiomatización. Las pantallas diseñadas (los forms) para actualizar los archivos son las distintas maneras de presentar los datos de éstos. Si estos forms se

expresan como fórmulas, y si los forms deben ser ‘obtenibles’ de los archivos, también expresados como fórmulas, entonces podemos reformular el problema a la manera de un cálculo lógico. Las fórmulas que especifican los archivos son los axiomas, y las fórmulas que especifican los forms son los teoremas.

Hay distintas formas de axiomatización de teorías, dos de ellas son la euclídea y la hilbertiana (Stegmüller, 1973). La axiomatización euclídea, o como también la llama el autor, ingenua, consiste en establecer el conjunto de axiomas, que son las verdades evidentes. La axiomatización hilbertiana, lo que hoy se conoce como cálculo lógico, es un sistema de símbolos desinterpretado sobre los que se construye un lenguaje formal, una sintaxis, que establece las fórmulas bien formadas. De entre las fórmulas bien formadas se escoge un subconjunto que representa los axiomas (sin ninguna referencia a verdades evidentes), y se establece un conjunto de reglas de transformación mecánica (esto es, sin tener en cuenta su contenido significativo), mediante los que se pueden transformar unas fórmulas en otras. Las fórmulas que se obtienen a partir de los axiomas utilizando las reglas de transformación (de inferencia) son llamadas teoremas.

Las teorías científicas matematizadas, como las de la física, siguen un procedimiento análogo al de la axiomatización hilbertiana. En ciencia no se validan los axiomas sino los teoremas que se pueden obtener de los primeros. Por ejemplo, cuando Newton conjeturó la fórmula de gravitación universal, junto a las tres leyes de la dinámica, éstas estaban muy lejos de ser verdades evidentes. La ‘verdad’ de dichas leyes se comprobó por medio de deducciones matemáticas (mecánicas) a partir de ellas, verificando que las leyes especiales que se obtenían a partir de ellas se correspondían, predecían, una gran variedad de fenómenos físicos. En realidad, en ciencia no se validan las leyes generales (los axiomas). Los axiomas son conjeturas que normalmente no se comprueban directamente, y nunca pueden llegar a ser validados con absoluta certeza, sino que se falsan. Cuantas más veces se ha fracasado en el intento de falsarlos en distintos ámbitos de aplicación más ‘válida’ será considerada la teoría.

De la misma manera, en este trabajo se defiende que, excepto en los casos de esquemas conceptuales triviales, lo que los usuarios deben validar son los teoremas, los formularios que se obtienen del esquema

conceptual. En este trabajo se argumenta que la validación directa de los esquemas conceptuales por parte del usuario presenta muchos inconvenientes. Pero el decidir cuál es la mejor forma de validar (o falsar) los esquemas conceptuales depende de cuáles sean nuestras presuposiciones metafísicas sobre las nociones de ‘verdad’ y de ‘concepto’.

La comparación con los presupuestos metafísicos de otros métodos

Al revisar otros métodos alternativos para compararlos con nuestra propuesta, tuvimos que enfrentarnos con serias dificultades de comprensión de los distintos métodos. El primer paso que ayudó a superar esta dificultad fue al observar que las distintas presuposiciones metafísicas que los autores de los métodos hacen sobre la noción de concepto se asemejan a lo que los historiadores de la filosofía conocen como ‘la batalla de los universales’. Un debate sobre la naturaleza de los conceptos que tuvo lugar entre los filósofos de la Edad Media.

Una idea de las dificultades de comprensión y del tipo de malentendidos que se producen nos la ofrece el diccionario Merriam-Webster cuando entre las diferentes acepciones de la palabra ‘proposition’ nos presenta la segunda acepción, la filosófica, con dos posibles significados. En el primer significado define proposición como ‘*2a: an expression in language or signs of something that can be believed, doubted, or denied or is either true or false*’ y con el segundo significado afirma que una proposición es ‘*2b: the objective meaning of a proposition*’. Aparte de la circularidad de la definición es imposible entenderla sin tener en cuenta que hay autores que asumen la existencia de significados independientes de la cognición humana.

Estos problemas de anfibología contaminan fuertemente las traducciones, agravando el problema de los malentendidos, y por tanto aumentando la dificultad de comprensión de los textos de los distintos filósofos. Por ejemplo, al traducir a Aristóteles al inglés, un traductor pone ‘statement-making sentence’ (Aristotle, 350a B.C.) donde otro escribe ‘proposition’ (Aristotle, 350e B.C.). Otro ejemplo lo encontramos en la traducción del término alemán ‘gedanke’, que tal como lo utiliza Frege es traducido como ‘thought’, pensamiento, (Frege, 1918), pero que Carnap prefiere traducirlo como ‘proposition’ (Carnap, 1956; p.124).

Mientras se revisaban distintos filósofos se vio que la noción de información no se acaba en la noción de concepto, hay otras acepciones complementarias que también son relevantes y críticas en el análisis y diseño de sistemas de información.

Por lo menos desde Aristóteles sabemos que toda oración con significado transmite información, pero no toda oración es una declaración, sólo lo son aquellas que son susceptibles de ser verdaderas o falsas (Aristotle, 350a B.C.; p.26). Por ejemplo, una orden puede ser correcta o incorrecta, apropiada o no, pero no es ni verdadera ni falsa. Frege, aunque básicamente se centra en las proposiciones, también advierte de que hay oraciones que transmiten información aunque no tienen contenido proposicional, y otras, como las peticiones, que transmiten más información que su contenido proposicional. Austin y Searle han desarrollado estas ideas en su teoría de los actos de habla (speech acts). Mientras que para Aristóteles la unidad de transmisión de significado es el concepto, y para Frege, la oración, para Austin y Searle lo es el acto de habla.

Esta teoría refuerza la defensa de la utilización de los formularios como objeto de validación de los sistemas de información. Un formulario, ya sea un pedido, una factura o un parte de producción, es un acto de habla, la unidad de transmisión de significado. Si esto es así, el intentar validar los significados a nivel de concepto o de proposición, que es como se propone en el área de modelado conceptual, puede dar origen a malentendidos.

Además, como presentaremos en este trabajo, hay distintos tipos de actos de habla, entre ellos están los enunciados y las peticiones. Una diferencia curiosa entre ambos es que se produce la inversión referencial. Mientras que un enunciado es una representación de la realidad, asume que las palabras se ajustan a la realidad, en una petición la correspondencia es a la inversa, es la expresión de un deseo, de que la realidad se ajuste a las palabras.

Organización de este trabajo

En este trabajo no se contesta una pregunta de investigación. Se propone un método de validación de esquemas conceptuales. Para juzgar el mérito o demérito de la propuesta cabe tener en cuenta lo que se ha

debatido recientemente en la comunidad de sistemas de información. Recientemente algunos autores reclaman que para juzgar los trabajos presentados para su publicación se tenga en cuenta que dentro de la disciplina hay dos tipos de trabajos, unos intentan explicar cómo se comporta el ‘mundo’ afectado por las tecnologías de la información, éstos son la mayoría en las revistas, mientras que otros trabajos proponen soluciones a los problemas detectados por los primeros. Al primer grupo le llaman ‘behavioral science’ y al segundo ‘design science’ (Hevner et al., 2004), que coincide con lo que Simon (1996) llamó ‘ciencias de lo artificial’.

Para juzgar este último tipo de trabajos, primero debe especificarse el problema y mostrar que no ha sido propuesta ninguna solución satisfactoria. Esto se hace en el capítulo 2, en el que además se intenta precisar el significado de algunos de los términos utilizados.

En el capítulo 3 se presenta la notación del método que proponemos. Esto lleva a las preguntas que se deben contestar en las ciencias del diseño, ‘*What utility does the new artifact provide?*’ y ‘*What demonstrates that utility?*’ (Hevner et al., 2004). O, tal como lo expresa Orlikowski: ‘*A considerable portion of IT research centers on the design, deployment, and use of artifacts that represent tangible solutions to real-world problems. ...As in engineering, the practical question, ‘What works?’ drives much of IT research.*’ (Orlikowski and Barley, 2001).

Pensamos que la pregunta de ‘*What works?*’ no tiene una contestación absoluta. La respuesta es relativa a otras propuestas existentes. La argumentación debe ser ‘mejor que’ o ‘peor que’ otra propuesta. Pero la oportunidad y viabilidad de una propuesta depende de la naturaleza que se atribuya a la información.

En el estado del conocimiento actual, la naturaleza de la información, la pregunta de qué es información, no es reducible a otras nociones más básicas, por ello la consideramos como una noción primitiva, al igual que hacen todos los métodos de análisis y diseño de sistemas de información que hemos revisado. Sin embargo, al revisar los distintos métodos para poder compararlos con el que proponemos, se observó que atribuían, normalmente de manera tácita, características diversas y discrepantes a la información. Esto nos llevó a revisar la noción de información.

En el capítulo 4 se estudia la noción de información como representación de la realidad, presentando algunas de las diferentes concepciones que han sido propuestas en filosofía. Esto permite distinguir y posicionar, siquiera aproximadamente, los distintos paradigmas que son asumidos, muchas veces implícitamente, en los diversos métodos propuestos.

El problema de qué características tienen los conceptos no ha sido debatido explícitamente en el campo de los sistemas de información. En este sentido, la explicitación de los distintos significados del término 'concepto' pretende ser una, aunque pequeña, aportación científica. Aunque, el considerarlo una aportación pasa por asumir que, como afirma Quine, *'The less a science has advanced, the more its terminology tends to rest on an uncritical assumptions of mutual understanding'* (Quine, 1966; p.77) Es una propuesta de abrir un debate que pueda llevar a una deseable convergencia en el significado de los términos entre las distintas áreas de investigación y producir ciencia acumulativa.

En el capítulo 5 se presenta un resumen de algunas de las distintas acepciones con que se utiliza el término información. Desde cómo se entiende en física y matemáticas hasta cómo la consideran algunos filósofos del lenguaje y de la sociología.

Por último, en el capítulo 6 se presentan las conclusiones del trabajo presentado y se indican las insuficiencias del mismo, o dicho de forma positiva, las posibles líneas de trabajo que abre.

Capítulo 2: Proyectos de sistemas de información

El problema de ambigüedad y de diferentes niveles de abstracción que hemos observado en la definición del término requerimiento también lo encontramos en la definición de ‘sistema de información’. Distintas comunidades entienden de manera diferente el término.

Hay comunidades que restringen su denotación a la máquina de software que se ha de construir, tal es el caso del Departamento de Defensa de Estados Unidos –posiblemente el mayor comprador de aplicaciones informáticas del mundo:

An information system (IS) consists of mission-specific applications, data, and technical infrastructure architecture consisting of support applications, application platforms, and the external environment including devices such as terminals, printers, and communications networks. (Defense Information Systems Agency. Center for Standards, 1996)

Otras acepciones lo consideran como un aspecto de la organización:

Aspect of an organisation, that provides, uses and distributes information. It is thus an aspect of a human system, possibly containing computer systems. (Euromethod Project, 1996)

O, directamente como una organización, esto es, como un sistema social que utiliza tecnología y emplea un lenguaje parcialmente formalizado

An information system consists of a collection of people, processes, data, models, technology and partly formalized language, forming a cohesive structure which serves some organizational purpose or function. (Davis and Olson, 1985)

*Propuesta de un método de validación de esquemas conceptuales
y análisis comparativo de la noción de información
en los métodos de desarrollo de sistemas de información*

Otros autores lo consideran a un nivel más conceptual, como una representación de la realidad: '*An information system is an artificial representation of a real-world system as perceived by humans.*' (Wand and Weber, 1988)

O como un lenguaje formalizado: '*Information systems are "social systems only technically implemented." They are formal linguistic systems for communication between people which support their actions.*' (Goldkuhl and Lyytinen, 1982)

O una definición más desarrollada de este último enfoque:

El sistema de información es un lenguaje de comunicación de la organización, construido conscientemente para representar, de manera fiable y objetiva, rápida y económica, ciertos aspectos de su actividad, pasada o futura. Las frases y las palabras de este lenguaje son los datos cuyo significado viene de las reglas con que los elaboran los hombres o las máquinas. Los mecanismos de representación propios de este tipo de lenguajes obtienen su eficacia en la repetición de los actos de las organizaciones. (Peaucelle, 1981)

En el proyecto FRISCO (Verrijn-Stuart, 2001) agrupan las distintas concepciones de sistema de información en tres grandes grupos, un sistema de información puede ser entendido:

- *As a technical system, implemented with computer and telecommunications technology.*
- *As a social system, such as an organization in connection with its information needs.*
- *As a conceptual system (i.e. an abstraction of either of the above).*

Como señalan Iivari y Hirschheim (1996), estas tres interpretaciones no son contradictorias, se refieren a tres facetas de una misma realidad:

An information system is interpreted here as a computer-supported system which provides a set of people (users) with information on specified topics of interest in a certain organizational context. According to this interpretation, one can distinguish three major aspects in the modeling of information systems: organizational context and users (host organization), topics of interest to the users (Universe of Discourse), and computers (technology).

Y podemos considerarlos como tres sistemas indisociables: El sistema social, el sistema conceptual y el sistema de software. Para que un sistema social funcione como un grupo es necesario que sea establecido

un lenguaje común y unas categorías conceptuales que les permitan entenderse mutuamente. Tal como afirma Schein (1992), si los miembros de un grupo no se pueden comunicar y entenderse mutuamente el grupo es imposible por definición. Esta es una condición necesaria para la coordinación.

Cuando se construye software para sistemas de información se construye una máquina que sirva a estas necesidades de comunicación. De aquí surgen los tres niveles de requerimientos, los requerimientos de empresa, los requerimientos de información y los requerimientos de software. Para delimitarlos los situaremos en el contexto del conjunto de actividades que se producen durante la construcción sistemas de software para sistemas de información.

Marco del problema

La actividad de análisis, diseño y construcción de sistemas de información involucra básicamente a tres tipos de agentes: los programadores que escriben los programas, los analistas que especifican los programas que se han de hacer y los usuarios que explicitan las necesidades que tienen. Dado que generalmente no todos los usuarios participan en el establecimiento de los requerimientos llamaremos 'prescriptores' a aquellos que prescriben, determinan o deciden, los requerimientos del sistema a construir.

El tradicional ciclo de vida de sistemas de información puede ser reducido a una secuencia de tres grupos de actividades básicas: la primera fase, consistente en entender el problema (en el Diagrama 2, ①: 'Elicitar'); la segunda, ②: especificar la máquina de software que se ha de construir, y la última, ③: construir la máquina.

En la fase 1 de elicitación, el analista trata de descubrir las necesidades exactas del prescriptor. Esta actividad es considerada la más difícil de llevar a cabo puesto que, como dice Brooks, muy frecuentemente el usuario no sabe lo que quiere e incluso cuando lo sabe no es capaz de expresarlo de manera precisa (Brooks, 1986). Tal como lo resumió un ingeniero de software: *'Writing the code isn't the problem, understanding the problem is the problem'* (Curtis et al., 1988).

En la fase 2, una vez el problema es entendido, el analista tiene que traducirlo a la notación que utilice para especificar el sistema. Aquí el

dilema está en utilizar una notación formal, de modo que el programador no pueda malentender, pero que el prescriptor normalmente no entiende, o hacer uso de una notación informal asequible, comprensible, para el prescriptor pero que el programador puede malinterpretar.

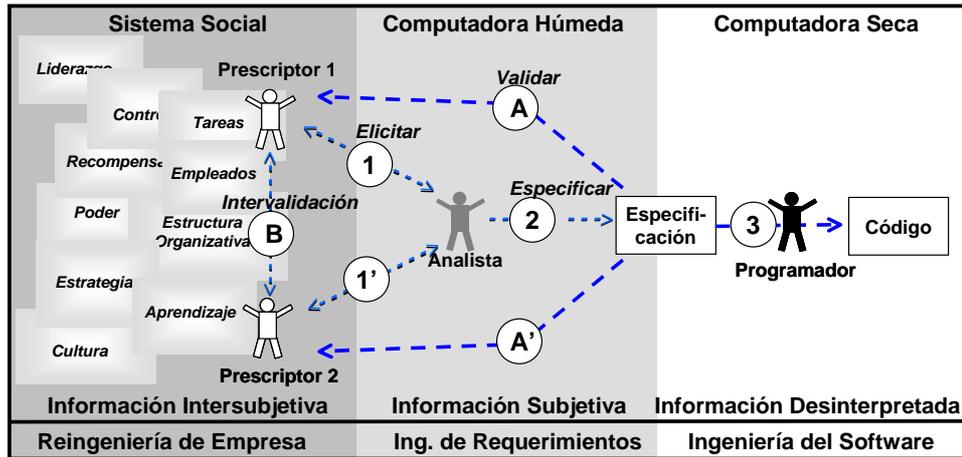


Diagrama 2. Cambio de la naturaleza de la información al atravesar los tres mundos.

La fase 3, la programación, pertenece enteramente al mundo de la ingeniería del software, en el que no entraremos pues en este trabajo pretendemos centrarnos en el ámbito del análisis, o ingeniería, de los requerimientos. Además de porque, según los estudios citados anteriormente, no es una de las principales causas del fracaso de los proyectos.

Tres facetas del problema

La información transmitida desde el prescriptor hasta el programador cambia de naturaleza conforme atraviesa los tres mundos representados en el Diagrama 2, el sistema social, la 'computadora húmeda' y la 'computadora seca'. Tomamos prestadas las metáforas 'húmeda' (wet) y 'seca' (dry) de los artículos de Goguen (Luqi and Goguen, 1997; Goguen, 1996 ; Goguen, 1994). El término 'computadora seca' quiere indicar que el mundo de los ordenadores es formal, no dependiente del contexto. Las computadoras son máquinas, procesadores sintácticos, a diferencia de los cerebros humanos (computadoras húmedas), que son procesadores semánticos. Los humanos necesitamos entender,

aprehender el significado de las palabras. Aquí la información depende del contexto y está situada socialmente. Goguen sostiene que la reconciliación de estos dos mundos es la esencia de la ingeniería de requerimientos.

Siguiendo a Hirschheim et al. (1995), extendemos esta imagen a un tercer mundo, el de empresa: el sistema social, en donde los significados deben ser compartidos para poder comunicarse y conseguir así la correcta coordinación de las tareas compartidas. En este mundo, que se puede calificar de hermenéutico, la información es intersubjetiva. Aquí surgen los conflictos más difíciles de tratar por el analista, los conflictos semánticos que se producen entre los prescriptores. Además, en este ámbito, el compartir significados es una necesidad pero no el objetivo final.

Diseñar un sistema de software para empresa es rediseñar, en mayor o menor medida, la empresa. El objetivo es diseñar una organización mejor. En un sentido laxo del término, y aunque excesivamente reduccionista¹ llamaremos a este dominio de problemas ‘reingeniería de empresa’ (Hammer, 1990; Hammer and Champy, 1993). En este ámbito deben tenerse en cuenta un gran número de factores de empresa que en muchos casos son difíciles de objetivar y explicitar: nuevas posibilidades de procesos, tareas, estructuras organizativas, sistemas de control y recompensa, entre otros; que a su vez están condicionados o posibilitados, además de por las nuevas prestaciones de las tecnologías de la información, por el factor humano, la cultura de la empresa, las relaciones de poder, el tipo de liderazgo y la estrategia del negocio. El problema no es tanto adecuar el nuevo sistema de software a la empresa, cuanto conseguir una organización que aproveche las posibilidades de las nuevas tecnologías de la información y un sistema de software que sirva fielmente a los nuevos diseños.

¹ A todas luces, calificar de ‘computadora social’ o, como se hace a continuación, hablar de ‘reingeniería’ de un sistema social es emplear unas metáforas mecanicistas, deshumanizadas, para denotar una realidad que es mucho más compleja y rica en facetas. Aun a riesgo de ofender la sensibilidad y la inteligencia del lector, mantenemos los términos pues sirven a nuestro propósito de resaltar la faceta que se pretende estudiar.

Si aceptamos, como es generalmente asumido, que la ingeniería de requerimientos es el proceso mediante el cual se procura entender las necesidades exactas de los usuarios para traducirlas a una especificación precisa y libre de ambigüedades que posteriormente será utilizada para la construcción del sistema de software (IEEE ,1990; Loucopoulos and Karakostas, 1995), entonces podemos concluir que la esencia de la misma cuando se construyen sistemas de software para sistemas de información de empresa es la reconciliación de los tres mundos citados.

El problema no sólo es que se producen fallos en la comunicación de requerimientos, malentendidos, entre analista y prescriptores sino que es difícil que se comprendan mutuamente. Un alto directivo participante en un estudio realizado en 10 empresas multinacionales manifestó que los analistas de sistemas de información no entendían lo que se les requería, no entendían el negocio, tendían a concentrarse en el aspecto técnico de los problemas y vivían en su propio mundo lleno de jerga técnica (Kettinger and Choong, 2002). Esta desavenencia de percepciones está descrita en (Scharer, 1981) y es presentada a un nivel más esencial por Schein desde el aspecto de la colisión entre culturas profesionales, maneras de entender el mundo que en ocasiones provocan que se atribuyan distintos significados a las mismas palabras (Schein, 1997 ; Schein, 1996)

Si las especificaciones del nuevo sistema de software no son comprensibles para los directivos y usuarios, es difícil conseguir el compromiso de los primeros y la involucración en el proyecto de los segundos, que como se ha indicado anteriormente son las otras dos principales causas de fracaso de los proyectos. Comprometerse en algo sin comprenderlo requiere hacer un acto de fe, virtud poco adecuada a este tipo de actividades.

Otro momento en el que se pueden producir malentendidos aparece cuando el programador, para escribir el código, interpreta las especificaciones del sistema de software a construir. La solución apunta a conseguir que las especificaciones sean formales. Pero esto acarrea el problema de que entonces se hacen más difícilmente comprensibles por los prescriptores.

Algunos autores aducen que es imposible conseguir especificaciones que sean simultáneamente formales y comprensibles por los usuarios. Por ejemplo, podemos leer: *'Achieving all of the preceding attributes in an SRS*

(Software Requirements Specification) is impossible. For example, as we attempt to eliminate inconsistency and ambiguity (usually by reducing the natural language in the SRS), the SRS becomes less understandable to customers who are not computer specialists' (Davis, 1993).

O, afirmado más categóricamente: *'...there is another belief —this one accepted by most— that specifications must be readable by the intended customers/users of the software product and that formal methods do not fulfill that need'* (Glass, 1996).

O, más explícitamente: *'The need for a formal specification, however, contradicts the need for specifications which are comprehensible by the users'* (Loucopoulos and Karakostas, 1995).

El propósito de este trabajo es mostrar que la presunta aporía, afirmada por estos últimos autores, no es tal. Pero para ello, antes es necesario definir con más precisión y menos ambigüedad los términos 'formal' y 'comprensible'.

La naturaleza del problema

Formalidad: información desinterpretada. La computadora seca.

Es una ironía que en el campo de los métodos formales, cuyo propósito es conseguir la precisión y el rigor en las especificaciones, el término 'formal', que en el ámbito de la lógica matemática tiene un significado preciso, se use con significados distintos y, por tanto, carezca de dichos atributos. Según Glass (1996) la definición de método formal es muy imprecisa: algunos consideran método formal a 'cualquier técnica rigurosa' para desarrollar software. Otros, un método basado en técnicas matemáticas para describir un sistema (Bowen and Hinchey, 1995). O también, los métodos basados en matemáticas elementales que pueden ser utilizados para producir documentación precisa y sin ambigüedades (Woodcock and Davies, 1996). En una reunión de trabajo para estudiar la especificación formal de las interfaces de usuario, a la que acudieron grupos de matemáticos, ingenieros de software y especialistas en diseño de interfaces, rápidamente quedó patente que el término 'método formal' tenía significados diferentes para los distintos grupos (Rouff, 1996).

La definición de 'especificación formal' que propone la IEEE (1990) tampoco está libre de imprecisión y ambigüedad:

*Propuesta de un método de validación de esquemas conceptuales
y análisis comparativo de la noción de información
en los métodos de desarrollo de sistemas de información*

1. *A specification written and approved in accordance with established standards.*
2. *A specification written in a formal notation, often for use in proof of correctness.*

La primera acepción es ampliamente utilizada en círculos profesionales, mientras que la segunda está más circunscrita a la comunidad de métodos formales (Bowen and Hinchey, 1995).

Estos autores dan una idea de la divergencia de significados:

What is clear is that despite 25 years of use, few people understand exactly what formal methods are or how they are applied. Many nonformalists seem to believe that formal methods are merely an academic exercise—a form of mental masturbation that has no relation to real-world problems.

Sin embargo, existe una definición que, aunque más restrictiva, es totalmente precisa y carece de ambigüedad. Es la que propuso Turing como respuesta al problema planteado por Hilbert de encontrar un criterio objetivo para decidir si una demostración matemática era válida o no. Turing inventó la máquina que lleva su nombre, diseñó un ordenador ideal. Y estableció que una especificación y una demostración son formales si son ejecutables por un ordenador (Chaitin, 1982)

En el campo de la lógica matemática se asume que los símbolos de un sistema formal no tienen significado. Si cuando se realiza la demostración de un teorema se persigue la certeza de su corrección, entonces se debe aplicar mecánicamente —mediante una máquina de Turing— un conjunto de reglas de inferencia desinterpretadas a un conjunto de símbolos también desinterpretados (Hamilton, 1988; Quine, 1981).

Nagel presenta brevemente el significado de cálculo formal: *'...the complete formalization of a deductive system. This involves draining the expressions occurring within the system of all meaning: they are to be regarded simply as empty signs. How this signs are combined and manipulated is to be set forth in a set of precisely stated rules. The purpose of this procedure is to construct a system of signs (called a 'calculus') which conceals nothing and which has in it only that which we explicitly put into it. The postulates and theorems of a completely formalized system are 'strings' (or finitely long sequences) of meaningless marks, constructed according to*

rules for combining the elementary signs of the system into larger wholes.' (Nagel and Newman, 1958; p.26)

Utilizaremos el término 'formal' en este restrictivo pero preciso sentido. Estas estrechas acepciones de 'método formal' y 'especificación formal' son especialmente adecuadas a nuestro problema pues los programadores no pueden rehuir la formalización. Deben acabar escribiendo objetos formales: programas de ordenador. Por ello, cuanto antes se formalice menos posibilidades de malentendidos, de errores de interpretación, se producirán durante el proceso de ingeniería de software (Wing, 1990).

La precisión y exactitud de la información.

Los problemas de comunicación entre analista y programador pueden ser atenuados en gran medida mediante la formalización. Los términos del sistema de información a construir (cliente, producto, etc.) no tienen otro significado para el programador que las acciones que debe ejecutar la máquina respondiendo a las distintas peticiones que le demandan los usuarios. Los términos que utiliza el sistema actúan como etiquetas sin más referencia que el comportamiento interno de la máquina.

Sin embargo, los términos que representará el sistema de información deben tener pleno significado para los usuarios que lo utilicen, por esto la solución de desinterpretación no puede aplicarse. Y tampoco puede hacerse en el momento de la comunicación entre prescriptor y analista, pues si los prescriptores fueran capaces de expresar sus requerimientos formalmente, la figura del analista no sería necesaria. Para poder especificar los requerimientos, el analista debe entender lo que le pide el prescriptor y reflejar, mediante los términos que este último utiliza, los requerimientos del sistema a construir.

Existen dos mitos, asumidos por prominentes autores, que presentan una vía de solución a estos problemas de comunicación: el mito del 'museo' y el de la 'precisión lingüística'. Si los términos generales se correspondiesen a conceptos perfectamente delimitados o delimitables, no se producirían problemas de comunicación ni horizontales, entre prescriptor y analista, ni verticales, entre prescriptores.

El mito del museo, calificado por algunos filósofos analíticos como semántica ingenua, consiste en considerar los conceptos, los significados de los términos generales, como los objetos exhibidos en un museo, en

el que los objetos son los significados y las etiquetas que los acompañan son las palabras. Esto supondría que cambiar de lengua es cambiar simplemente las etiquetas (Quine, 1969; Putnam, 1988). El lingüista Saussure (1915; p.116) argumenta: *'If words stood for pre-existing concepts, they would all have exact equivalents in meaning from one language to the next; but this is not true.'*

El mito de la precisión lingüística, que no presupone los conceptos, considera no problemático establecer el significado de los términos con gran precisión y concisión (Hirschheim et al., 1995). Aunque en algunos ámbitos donde se aplica la ingeniería de software, como el de control de sistemas físicos, esto muchas veces es posible, en el mundo de la gestión de empresa el significado de los términos es frecuentemente ambiguo y difuso.

Un ejemplo de este último mito lo encontramos en el área de ingeniería de requerimientos, en donde podemos leer que, cuando el analista especifica los requerimientos, los términos primitivos deben explicarse con claridad y precisión (Zave and Jackson, 1997) (el sentido 'primitivo' en este contexto es de 'no definible a partir de otros'). Implícitamente, dan la impresión de que la precisión y la claridad son únicamente un problema que depende de la capacidad y del trabajo del analista. Intentar dar definiciones, o explicaciones, de los términos primitivos no sólo es una tarea difícil y consume mucho tiempo al tener que dominar conceptualmente el dominio de aplicación, muchas veces tiene posibilidades de éxito dudosas.

En el ámbito de los métodos de análisis orientados a objetos también aparece la pretensión de la precisión en la definición de los objetos y de las clases de objetos (que corresponden a los términos primitivos). Podemos leer: *'We define an object as a concept, abstraction, or thing with crisp boundaries and meaning for the problem at hand.'* (Rumbaugh et al., 1991), y *'we define an object as anything with a crisply defined boundary.'* (Booch, 1994), con la recomendación *'...make sure that each class is crisply defined...'* (Booch et al., 1999)

En el campo de modelos conceptuales, de los modelos de datos semánticos, algunos autores parecen asumir que es posible alcanzar la precisión y exactitud de los términos utilizados:

Accurate specifications and validation of information requirements is critical to the development of organizational information systems. Semantic data models were developed to provide a precise and unambiguous representation of organizational information requirements. They serve as a communication vehicle between analysts and users. (Kim and March, 1995; p.103)

Los conceptos o clases parecen ser entes perfectamente definidos que se corresponden con una realidad. El dilema es que frecuentemente al intentar aumentar la precisión de una definición perdemos irremediablemente su exactitud.

El problema de la precisión y exactitud de la información

Aunque los diccionarios muchas veces consideran los términos 'precisión' y 'exactitud' como sinónimos, emplearemos la distinción propuesta por Russell (1985), que es la que generalmente es entendida en ciencia. Exactitud (accuracy) es el grado de correspondencia de una especificación con la realidad. Precisión es el grado de detalle en que queda delimitada la especificación. Ilustremos el dilema con un ejemplo físico.

Si decimos que la Tierra es una esfera de radio $6,3 \times 10^3$ Km, es una afirmación cierta, es una proposición exacta pero su precisión es de 100 Km. Si intentamos aumentar la precisión a los 10 Km. y decimos que el radio es de $6,37 \times 10^3$ Km, ya no es cierta, no se corresponde con la realidad. A este nivel de precisión la Tierra no tiene simetría esférica, es un elipsoide de revolución. Para ser exactos a este nivel deberíamos dar la longitud de los dos semiejes, el polar y el ecuatorial, que difieren en 21 Km. Con una precisión de 1 Km, nuestro planeta ya no tiene simetrías, no tiene suficientes regularidades. La descripción exacta del objeto requiere muchos más datos. Cuando pretendemos llegar a la precisión de 1 metro, descubrimos que en breves intervalos de tiempo cambia su forma.

Un corolario que se puede extraer de este ejemplo es que no podemos describir un objeto con más precisión y exactitud de la que permitan las regularidades que muestra. Este hecho, que con objetos físicos visualizables es evidente, no parece ser tan claro cuando se trata con 'objetos abstractos', con conceptos o términos generales. Por ejemplo, el término 'montaña' tiene dos tipos de indeterminaciones (de vaguedad):

dónde empieza y acaban las montañas concretas, y cuándo una protuberancia en el terreno es o no una montaña. La ambigüedad está en que puede significar distintas cosas, por ejemplo 'estar en la cima de una montaña' y 'tener una montaña de problemas' (Quine, 1960).

La terminología de empresa es más difícil de tratar que la del anterior ejemplo. A fin de cuentas, el término 'montaña' se refiere a una realidad física, sin embargo, muchos de los términos empleados en la empresa, como 'cliente', 'producto' o 'coste de un producto', son más abstractos, se refieren a construcciones sociales, reflejan una realidad social, en mayor o menor medida indeterminada.

En dos manuales de marketing que hemos revisado (Kotler, 1974) (Santesmases Mestre, 1996) no aparece ninguna definición ni aclaración de lo que es un cliente, lo que no deja de ser sorprendente cuando es un concepto central en dicha disciplina.

Podríamos arriesgar nosotros una definición: Cliente es una persona física o jurídica a la que vendemos o alquilamos los productos o servicios que produce nuestra compañía. Pero esto no elimina la indeterminación ni la ambigüedad. Si vendemos una máquina a una empresa, nuestro interlocutor no puede ser la persona jurídica, que es un ente abstracto. Entonces nos deberemos preguntar quién es el cliente: ¿el que la paga? ¿el que la utiliza? ¿el que la prescribe? ¿el que asesora? ¿el que decide la compra? ¿el que la autoriza? ¿el que firma el contrato? Además, deberíamos preguntarnos si el que decide es una persona o se toman decisiones colegiadas.

Pero la indeterminación no está sólo en quién es el representante de nuestra empresa cliente, también está en la delimitación de la empresa en sí. La delimitación jurídica muchas veces no es adecuada para usos no jurídicos. Por ejemplo, una empresa multinacional puede tener diferentes personalidades jurídicas en distintos países y sin embargo tener la gestión fuertemente centralizada, actuar como una única empresa. O, por el contrario, una empresa con una única personalidad jurídica puede estar fuertemente divisionalizada con un gran nivel descentralización, y actuar cada división como empresas diferentes. Además, el grado de propiedad de las empresas subsidiarias puede variar desde la propiedad total hasta tener una pequeña participación. La pregunta es cuándo dos entidades deben ser consideradas una misma entidad. La respuesta no es única, dependiendo del propósito con que la hagamos (por ejemplo, calcular el

riesgo financiero de una operación, o consolidar los balances, o gestionar las relaciones comerciales, etc.) la respuesta puede ser distinta. Además, estas estructuras de reificación son dinámicas, cambiantes en el tiempo. (Madnick, 2001)

En realidad, el término cliente expresa no un concepto sino un cúmulo (cluster) de ellos. Pero, además, este cúmulo puede cambiar de una empresa cliente a otra, y variar en el tiempo, y adoptar configuraciones diversas para los diferentes departamentos y las distintas divisiones sectoriales de nuestras empresas clientes.

Como hemos visto en el ejemplo de la descripción de nuestro planeta, el problema no es que no se pueda alcanzar una precisión elevada, la dificultad está en que cuando aumenta la precisión, según las regularidades que muestre el objeto, la cantidad de texto, de datos necesarios para describirlo, puede crecer desmesuradamente. Podemos resumir esto en el Diagrama 3, que relaciona la concisión de una especificación con su exactitud y precisión.

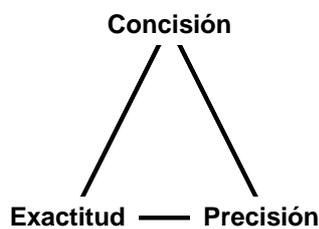


Diagrama 3. Concisión, exactitud y precisión.

Pero la concisión de una especificación o descripción no sólo depende de la naturaleza del objeto, también es función del contexto. En el ámbito lingüístico, Quine muestra que para resolver el problema de ambigüedad e imprecisión de los términos se necesita lo que llama 'masa semántica'. Frecuentemente, un término aislado no tiene

suficiente 'masa semántica' para ser preciso y unívoco. Pone el ejemplo de la palabra 'bore' que en inglés tiene un gran número de significados distintos. A veces una frase tampoco basta para eliminar la ambigüedad. 'Our mothers bore us' tiene dos significados, 'nuestras madres nos aburren' y 'nuestras madres nos dieron a luz'. A veces es necesario todo un párrafo o un libro entero para entender cabalmente una palabra (Quine, 1953). Otras veces incluso es necesario comprender toda una cultura para captar el significado de una palabra, tal como ocurre con la palabra 'balan', perteneciente a una lengua aborigen australiana, que aproximadamente significa 'mujeres, fuego y objetos peligrosos' (Lakoff, 1987).

$$\begin{aligned}\nabla \cdot \mathbf{D} &= \rho \\ \nabla \cdot \mathbf{B} &= 0 \\ \nabla \times \mathbf{E} &= -\frac{\partial \mathbf{B}}{\partial t} \\ \nabla \times \mathbf{H} &= \mathbf{J} + \frac{\partial \mathbf{D}}{\partial t}\end{aligned}$$

Diagrama 4. Ecuaciones de Maxwell.

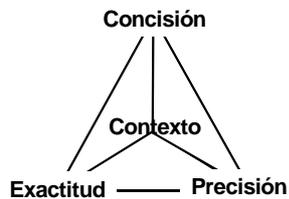


Diagrama 5. Concisión, exactitud, precisión y contexto.

Pero no hace falta que las culturas sean remotas en el espacio o en el tiempo para que surja la dificultad de entendimiento. Convivimos con 'culturas remotas'. 'Los neutrinos no tienen masa', o las ecuaciones de Maxwell del electromagnetismo (véase Diagrama 4), que serían proposiciones perfectamente incomprensibles hace quinientos años, para nosotros sólo tienen un significado muy vago.

Para entender cabalmente lo que afirman los físicos se requiere una cantidad ingente de 'masa semántica'. En todos estos casos el significado es relativo a la cultura, incluidas las culturas profesionales. Como resumen, ampliaremos el esquema anterior con otra variable, el contexto; lo que nos da un tetraedro, resultando la concisión de una especificación la resultante de la exactitud y la precisión de la misma, y

del contexto compartido por los dos comunicantes (Diagrama 5).

En el mundo de la empresa, uno de los problemas es la necesaria colaboración de diferentes culturas profesionales. Cuanto mayor es la distancia cultural entre ellas, mayor es la cantidad de información de contexto que se ha de transmitir para entenderse. Pero Quine sólo señala el problema, no ofrece ninguna solución.

El procedimiento hermenéutico

La hermenéutica es una disciplina que se ocupa de este tipo de problemas. En el campo de los sistemas de información ha sido propuesta para el análisis de sistemas de información (Goldkuhl and Lyytinen, 1982; Flores and Winograd, 1989). También para el estudio del uso de los sistemas de información (Boland Jr. 1991) y como procedimiento de investigación de los fenómenos relacionados con los sistemas de información (Lee, 1994; Walsham, 1995; Klein and Myers,

1999) —este último artículo fue considerado el mejor del año de la revista MIS Quarterly.

La hermenéutica nació como disciplina para intentar resolver el problema de interpretación correcta de los textos sagrados, distantes culturalmente en el tiempo. Tal como la define Schleiermacher, la hermenéutica es el ‘arte de evitar los malentendidos’. Definición que la hace especialmente prometedora para el tipo de problemas que nos ocupa.

El ámbito de aplicación de la hermenéutica, que empezó en la interpretación de los textos antiguos, fue ampliado por Schleiermacher al dominio de la lingüística: *‘Hermeneutics rests on the fact on the non-understanding of discourse: taken in its most general sense, including misunderstanding in the mother tongue and in everyday life’* (Schleiermacher, 1828; p.227). Y al de las ciencias sociales, contraponiéndolas a las naturales, por Dilthey, que formula de manera concisa lo que ya apuntó el anterior autor, el principio fundamental de la hermenéutica, el círculo hermenéutico: *‘A whole should be understood on the basis of the particular and the particular on the basis of the whole. This contradiction generates the procedure of the hermeneut. He operates with hypotheses’* (Dilthey, 1868; p.231)

Este principio es una formulación concisa y más fecunda del problema de la ‘masa semántica’ formulado por Quine. En efecto, para entender el significado de una oración es necesario entender previamente el significado de cada una de las palabras que la forman, pero muchas palabras tienen distintas acepciones y se pueden entender a diferentes niveles de precisión, por lo que para concretar su significado es necesario entender previamente el sentido de la de la oración de la que forman parte. Oración que a su vez es una parte de un texto, en donde se vuelve a repetir el círculo hermenéutico. A su vez, el texto es una parte de una cultura y ésta de una historia, que forman círculos de comprensión cada vez más amplios.

En el caso de que se compartan lengua, cultura y conocimientos, la convergencia entre lo que uno quiere decir y lo que el otro entiende puede ser rápida, pero la coincidencia exacta nunca se puede dar pues esto requeriría tener las mismas historias personales. Como afirma Dilthey *‘. all understanding always remains partial and can never be completed. Individuum est ineffabile’* (Dilthey, 1860; p.249). Por eso el procedimiento hermenéutico consiste en el diálogo en principio ilimitado en el tiempo.

Gadamer amplió el ámbito de aplicación de la hermenéutica a la filosofía. En su propuesta de analizar los problemas filosóficos desde la comprensión del lenguaje comparte con la filosofía analítica —tal como la plantearon Bolzano, Frege y Russell— y con la filosofía del lenguaje una premisa: que muchos problemas filosóficos son problemas de lenguaje y mediante el análisis de éste, mediante el esclarecimiento del lenguaje, es posible esclarecer el pensamiento. Como dicen estos últimos filósofos, muchos problemas filosóficos no se resuelven sino que, al esclarecer el significado de los términos empleados, se disuelven².

Pero, tal como la configuró Schleiermacher, y Gadamer reitera, la hermenéutica no es una ciencia, es una arte. A diferencia de la filosofía analítica, que parte de una tradición científica, Gadamer demarca claramente la hermenéutica de la ciencia:

El fenómeno de la comprensión y de la correcta interpretación de lo comprendido no es sólo un problema específico de la metodología de las ciencias del espíritu. ...de este modo ya desde su origen histórico el problema de la hermenéutica va más allá de las fronteras impuestas por el concepto de método de la ciencia moderna. Comprender e interpretar textos no es sólo una instancia científica, sino que pertenece con toda evidencia a la experiencia humana del mundo. En su origen el problema hermenéutico no es un problema metódico. No se interesa por un método de la comprensión que permita someter los textos, igual que cualquier otro objeto de la experiencia, al conocimiento científico. Ni siquiera se ocupa básicamente de constituir un conocimiento seguro y acorde con el ideal metódico de la ciencia.’ (Gadamer, 1960; p.23)

Y delimita claramente ambos dominios, el de la verdad y la certeza, el de la comprensión subjetiva del mundo y la comprensión científica:

² Es sorprendente que Quine y Gadamer, a pesar de ser dos de los filósofos más importantes del siglo XX y de haber vivido durante la práctica totalidad de dicho siglo, por lo que es imposible que no tuvieran noticia uno del otro, se ignoraron mutuamente.

Baso esta afirmación en la revisión que he efectuado en buena parte de la obra del primero y en los dos volúmenes del libro *Verdad y Método* del segundo (Gadamer, 1997 ; Gadamer, 1998), que es considerado como su obra central.

Andrie Bowie, en su introducción al libro de Schleiermacher (1998), también señala que parte de las conclusiones de los dos filósofos, a pesar de pertenecer a tradiciones filosóficas de origen muy dispar, son coincidentes.

'El ideal de conocimiento perfilado por el concepto de método consiste en recorrer una vía de conocimiento tan reflexivamente que siempre sea posible repetirla. 'Methodos' significa "camino para ir en busca de algo". Lo metódico es poder recorrer de nuevo el camino andado, y tal es el modo de proceder de la ciencia. Pero eso supone necesariamente una restricción en las pretensiones de alcanzar la verdad. Si la verdad (veritas) supone la verificabilidad --en una u otra forma--, el criterio que mide el conocimiento no es ya su verdad, sino su certeza. Por eso el auténtico ethos de la ciencia moderna es, desde que Descartes formulara la clásica regla de certeza, que ella sólo admite como satisfaciendo las condiciones de verdad lo que satisface el ideal de certeza.' (Gadamer, 1986; p.54)

Si entendemos la hermenéutica tal como la concibe Gadamer, aunque sea útil para plantear el problema, no lo es como procedimiento de solución del mismo. Precisamente, no podemos renunciar al ideal de certeza, a la verdad verificada. El problema de los proyectos de sistemas de información es encontrar un procedimiento sistemático, que se aproxime tanto como sea posible a un método, que permita a los analistas 'recorrer siempre el mismo camino', para intentar soslayar los factores contingentes y circunstanciales de su trabajo, atenuando así la gran tasa de fracasos que se producen. No podemos renunciar al método científico a pesar de movernos en un ámbito de problemas en el que la subjetividad humana —que el analista comprenda al prescriptor y que este último comprenda las especificaciones del primero— aparece como factor condicionante y determinante.

Significado: Información subjetiva. La computadora húmeda.

La demarcación entre método y arte de Gadamer se apoya en el significado del término 'comprensión'. Tal como explica Dilthey, la palabra 'comprensión' es bivalente. No denota lo mismo decir que entendemos el comportamiento de otra persona o que entendemos el comportamiento de un electrón. Comprendemos a otra persona cuando mediante la imaginación somos capaces de ponernos en su lugar, de identificarnos con ella. Sin embargo, este movimiento mental no funciona con un electrón. Decimos que hemos comprendido su comportamiento cuando somos capaces de explicar, basándonos en regularidades previamente observadas —en leyes físicas—, su comportamiento mecánico. Siguiendo a Dilthey, emplearemos el término 'explicación' para esta segunda acepción de 'comprender'.

*Propuesta de un método de validación de esquemas conceptuales
y análisis comparativo de la noción de información
en los métodos de desarrollo de sistemas de información*

Cuando el analista comprende lo que le dice el prescriptor no lo hace ‘científicamente’. Empleando el ejemplo que propone Giddens, su actitud se parece mucho a la adoptada cuando se lee una novela, que no requiere reflexión, la novela absorbe al lector de manera prerreflexiva (Giddens, 1976).

El acto de comprender es enteramente subjetivo, el que comprende algo es una persona, un sujeto. Si el problema es comprender lo que dice otro, la única vía es el diálogo. En el Diagrama 6, a la izquierda, está el gráfico que muestra el diálogo entre el prescriptor y el analista presentado anteriormente en el Diagrama 2. A la derecha de la figura reformulamos este mismo diálogo, que aparece en forma de dos círculos concéntricos. En el interno, de elicitación, el diálogo es puramente informal, difícilmente reducible a un método que asegure la perfecta comprensión. Para intentar reducir los malentendidos, la ingeniería de requerimientos propone, como ya hemos citado, un segundo diálogo, el de formulación; que se produce cuando el analista vislumbra que ha entendido al prescriptor. En ese momento cambia la forma de diálogo y pasa al bucle exterior. El analista especifica su propuesta de máquina, que el prescriptor interpreta y sitúa en el contexto de su organización para validarla o refutarla. En este último caso pasan de nuevo al bucle interno.

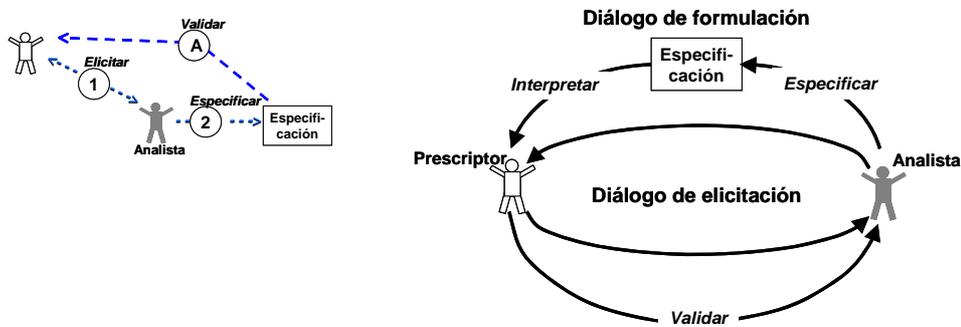


Diagrama 6. Bucle de elicitación embebido en el bucle de formulación.

En el Diagrama 7 se presentan los dos diálogos separados para resaltar la dirección del flujo de información, que es una manera de decir quién es el que debe comprender. En el primero, el de elicitación, la transmisión de información es básicamente desde el prescriptor hacia el analista. Esta comprensión debe ser exacta (se debe corresponder con lo que quiere decir el prescriptor) y al nivel de precisión requerido por el mismo. En el segundo diálogo, la información sigue el camino inverso al

anterior, ahora el que debe comprender es el prescriptor. Para ello debe interpretar la especificación propuesta por el analista.

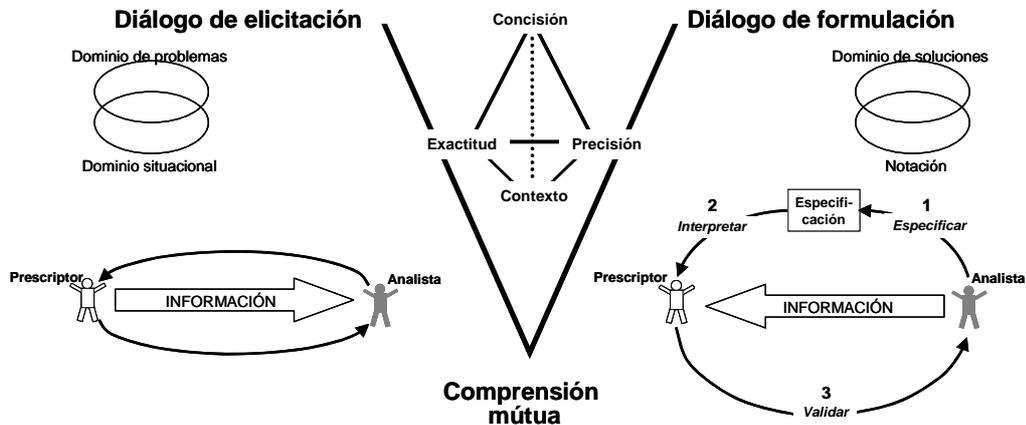


Diagrama 7. Flujos de información de sentido opuesto de los dos bucles.

La comunicación correcta se alcanza cuando se consigue la exactitud y la precisión. Por ello nos quedan las otras dos componentes, la concisión y el contexto. Cuanto mayor sea el contexto compartido, los conocimientos comunes, mayor será la concisión de la comunicación. Una medida de la concisión nos la da la cantidad de información, de ‘texto’, que se ha de transmitir, que se traduce en tiempo necesario para comunicarse correctamente, sin malentendidos. El problema en los proyectos de sistemas de información es que los dos términos nos vienen dados de antemano, el tiempo de comunicación y el contexto compartido por el analista y el prescriptor ya están prefijados cuando se inicia el proyecto.

La situación de partida es que el prescriptor conoce el dominio de problemas de su actividad (por ejemplo, los conocimientos de contabilidad) y también el dominio de situaciones en que aplica el primer dominio en su empresa concreta. Dominios que el analista desconoce.

Por la otra parte, el analista, sabedor de las posibilidades de las nuevas tecnologías de la información, conoce el dominio de soluciones de los problemas, y las notaciones para expresarlas, que el prescriptor ignora.

Deben dialogar dos subjetividades, separadas por distintas culturas profesionales y por distintas situaciones y contextos. Esto requiere para la perfecta comprensión una gran cantidad de transmisión de

*Propuesta de un método de validación de esquemas conceptuales
y análisis comparativo de la noción de información
en los métodos de desarrollo de sistemas de información*

información. Pero esta debe hacerse en una situación en la que el tiempo está limitado.

Aún en las situaciones ideales, en que el analista ya conoce el vocabulario del dominio de conocimientos del prescriptor, esto es, los requerimientos de información, y puede reflejarlos en una especificación formal, queda una parte de la información que no es fácil de codificar y explicitar, la relativa a los requerimientos de empresa, del sistema social. La especificación del nuevo sistema debe poder ser contextualizada en la situación concreta de la empresa analizada, para poder evaluar las posibilidades de rediseño de los procesos de empresa. Esta información es en muchos casos tácita y difícilmente explicitable. Por ello es imprescindible la validación por parte del prescriptor. Que, por otra parte, no hace más que cerrar el círculo de diálogo que propone la hermenéutica.

Por esto, si no queremos renunciar a la formalidad de las especificaciones, es necesario que estas sean comprensibles por los prescriptores, esto es, que puedan situarse en el contexto concreto de los usuarios que van a utilizar la máquina especificada.

Significación³: Información intersubjetiva. El sistema social.

Pero el problema no se acaba cuando el prescriptor y el analista se han comunicado correctamente. Los sistemas de información frecuentemente afectan a distintos colectivos de una misma organización e involucran a varios prescriptores. Para que el sistema de información que se quiere construir sea correcto se debe evitar que se produzcan malentendidos entre los diferentes prescriptores.

3

En el diccionario de la Real Academia Española, el término ‘significación’ tiene distintas acepciones, una de ellas es ‘sentido de una palabra o frase’. El problema es que el la palabra ‘sentido’ tiene todavía más acepciones que el término que pretende definir. Una de las que más se acerca al significado con el que lo queremos emplear cae en el problema de la circularidad y define ‘sentido’ como ‘significación cabal de una proposición o frase’ (DRAE).

Emplearemos el término ‘significación’ en la acepción con que utiliza Giddens el término ‘signification’, que es una de las acepciones que recoge el diccionario Webster para denotar ‘the meaning that a term regularly conveys or is intended to convey’(Merriam-Webster, 2000). Esto es, el significado que normalmente expresa un término, el que es compartido por una colectividad, el significado intersubjetivo.

Aun suponiendo que se pudiera solucionar el problema de los malentendidos en la transmisión de requerimientos que en el Diagrama 2 aparece como un movimiento horizontal, entre prescriptor y analista y entre este último y el programador, no se acabarían los posibles malentendidos. También se producen entre los distintos colectivos de la organización (por brevedad les llamaremos ‘departamentos’) representados por sus respectivos prescriptores.

El problema que aparece es que los términos que utilizan los distintos departamentos en ocasiones divergen en significado. Se descubre que no existen ‘regularly conveyed meanings’ a lo largo de la empresa, que hay términos que tienen significado pero no significación. Queda patente lo que defiende (Searle, 1969; p.117), que el significado de un término o de una oración puede cambiar cuando se cambia el contexto; por lo que es falaz hablar de significación, o significado literal, que es como lo llama este autor.

Desde luego, el problema de entendimiento entre distintos departamentos de una organización no es nuevo. De hecho las organizaciones han funcionado muchos años antes del advenimiento de las nuevas tecnologías, y los problemas de malentendidos no han sido graves. Lo que es nuevo es que la nueva tecnología de la información, que proporciona un soporte electrónico a la información —a diferencia de la anterior, cuyo soporte es el papel— permite que una vez se ha introducido una información en el sistema en un lugar de la empresa, ésta sea accesible desde cualquier otro punto de la misma sin importar su ubicación geográfica.

Pero aparece un nuevo problema, las personas que actuaban de interfaz entre los distintos departamentos, no sólo realizaban la tarea rutinaria de transcripción de datos de unos formularios a otros, sino que también realizaban una labor muy importante de interpretación de los mismos. Con la ventaja frente a la nueva situación de que cuando observaban una incoherencia en los datos podían establecer un diálogo de elicitación con su interlocutor. Las nuevas tecnologías al eliminar la rutina de transcripción de datos también eliminan la de interpretación.

Imaginemos que en el departamento de ventas registran el pedido de un cliente. Con tecnologías antiguas, en ventas deben hacer una copia del pedido y enviarla a almacén para que sirvan las mercancías solicitadas. En el almacén un empleado deberá transcribir la información del pedido

a otro documento, el albarán. Esto supone que en almacén tienen que volver a teclear los mismos datos. Si se utilizan las nuevas tecnologías éste es un trabajo que es innecesario, pues ralentiza la transmisión de la información y además puede producir errores de transcripción.

En la práctica real de las empresas, un simple pedido puede desencadenar una red más amplia de conversaciones entre distintos departamentos. Por ejemplo, para confirmar al cliente el pedido que ha realizado —para decirle si se le puede servir y comunicarle el plazo de entrega—, primero se debe consultar al departamento de finanzas el riesgo financiero del cliente; a almacén, si están disponibles los productos solicitados; en caso negativo, se preguntará a fábrica el plazo de fabricación; que, a su vez, si no dispone de las materias primas, deberá consultar al departamento de compras por el plazo de entrega de las mismas, quienes deberán consultar a los proveedores. Esto sólo para comunicar al cliente las condiciones de entrega. En el caso de que el cliente las acepte, vuelven a producirse las mismas conversaciones pero cambiando las consultas por órdenes.

Aprovechando las posibilidades de la tecnología actual se puede concebir un sistema que evite volver a teclear los datos en los distintos lugares en que se producen las conversaciones. Pero el trabajo del empleado que actúa de interfaz con otro departamento no se reduce a la función transcripción de datos, también realiza un trabajo de interpretación y traducción de los términos que le llegan de un departamento para adecuarlos a la terminología de su propio departamento. Esto es así pues los distintos departamentos —finanzas, marketing, fabricación— están imbuidos de distintas culturas profesionales y tienen distintos propósitos, lo que les lleva a utilizar distinta terminología, cada departamento tiene su propio esquema conceptual. Una condición necesaria para poder crear sistemas de software que permitan transmitir la información sin transcripciones es integrar los distintos esquemas conceptuales.

Pero esto conlleva una tarea difícil, primero detectar los conflictos semánticos y, después, el entendimiento y la negociación del significado de los términos entre los distintos departamentos. Esta actividad la hemos llamado intervalidación (B, en el Diagrama 2). Que establecer términos con significado público, con significación, no es una tarea fácil podemos comprobarlo en el artículo de Alter, cuyo título habla por sí

solo: *'Same Words, Different Meanings: Are Basic IS/IT Concepts our Self-Imposed Tower of Babel?'* (Alter, 2000).

Un hecho que agrava el problema de la descontextualización es que una manera humana de alcanzar acuerdos es disminuyendo la precisión y aumentando la ambigüedad de los términos (Astley and Zammuto, 1992). Un ejemplo de este fenómeno lo podemos observar si revisamos las definiciones de requerimiento y de especificación formal propuestas por la IEEE que han sido presentadas en las páginas 6 y 31 de este trabajo.

Delimitar un concepto, definir con precisión su significado, requeriría disponer de algún instrumento para determinar o medir significados. Carecemos de tal instrumento, o mejor dicho, el único que tenemos es nuestro cerebro, lo que da al problema un carácter irreduciblemente subjetivo. Esto implica que hemos de aceptar la dificultad de conseguir significados precisos. Pero no quiere decir que estos sean completamente subjetivos. Si el significado fuera exclusivamente subjetivo, personal, si las palabras no tuvieran, aunque imprecisa, una significación la comunicación humana sería imposible. Por tanto nos movemos, como afirma (Schleiermacher, 1828; p.233), en un punto intermedio: *'The elements of the language cannot be completely indeterminate, but also not completely determinate.'*

La propuesta de los modelos conceptuales

La utilización de un vocabulario coherente y compartido en toda la organización, esto es, el uso de un vocabulario con significación, es una condición necesaria para establecer una comunicación rápida y fluida en los procesos de control y coordinación de una empresa (Goodhue and Wybo, 1992). Durante años la disciplina de modelado conceptual ha atacado este problema de establecer la significación de los términos utilizados en una empresa. El propósito del modelado conceptual es el establecimiento de un esquema conceptual único para toda la empresa. En las tres últimas décadas hemos asistido a una gran proliferación de notaciones y de metodologías para realizar esta tarea. Lamentablemente, a pesar de los esfuerzos de distintos organismos de estandarización ANSI⁴ (ANSI/X3/SPARC, 1975), ISO⁵ (Griethuysen, 1982), NIST⁶

⁴ ANSI: American National Standards Institute

(NIST, 1993), IFIP⁷ (Verrijn-Stuart ,2001; Falkenberg et al., 1998), IEEE⁸ (IEEE, 1999) no existe consistencia terminológica ni siquiera en el nombre de lo que se produce como resultado del modelado conceptual. El nombre quizás más utilizado es el de ‘esquema conceptual’ que fue propuesto en (ANSI/X3/SPARC, 1975), pero también es referido como ‘modelo de empresa’ (business model) y ‘modelo de datos’ (data model) (Verrijn-Stuart ,2001) o como ‘modelo conceptual’ (Shanks et al., 2003), y también como ‘modelo de datos semántico’ (Wand et al., 1999) u ‘ontología’ (Guarino and Welty, 2002) o ‘diagrama de clases’ (Booch et al., 1999)

Esta situación no deja de ser una ironía si tenemos en cuenta que precisamente estos grupos de investigadores persiguen el desarrollo de métodos que permitan la definición consistente y compartida del significado de términos. Tal como se explica en (IEEE ,1999): *'The conceptual schema provides a single integrated definition of the concepts relevant to an enterprise, unbiased toward any particular application. The primary objective of this conceptual schema is to provide a consistent definition of the meanings and interrelationship of concepts. This definition can then be used to integrate, share, and manage the integrity of the concepts.'*

En el informe ISO (Griethuysen ,1982) se define el esquema conceptual como *'A consistent collection of sentences expressing the necessary propositions that hold for a universe of discourse.'* En donde se entiende por ‘universo de discurso’: *'The collection of all objects (entities) that ever have been, are, or ever will be in a selected portion of a real world or postulated world of interest that is being described.'*

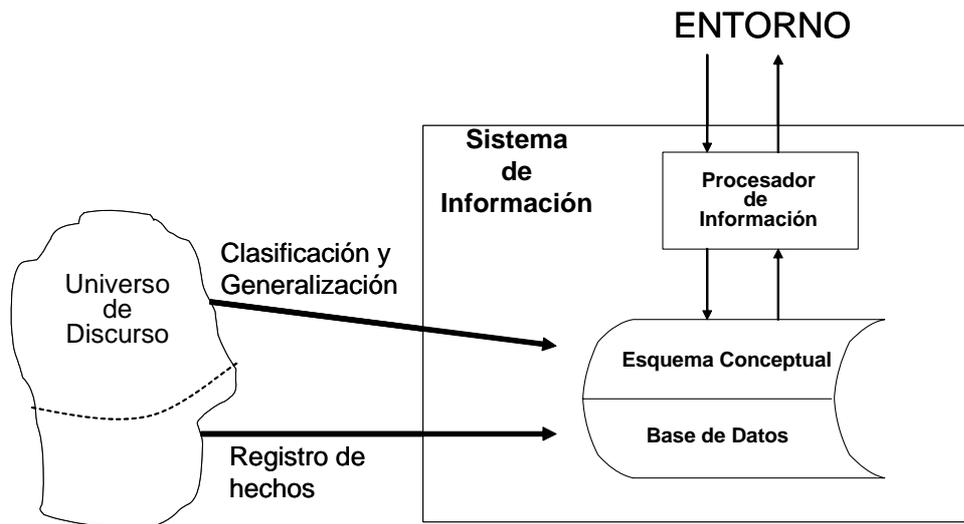
⁵ ISO: International Organization for Standardization

⁶ NIST: US National Institute of Standards and Technology

⁷ IFIP: International Federation for Information Processing

⁸ IEEE: Institute of Electrical and Electronics Engineers. Software Engineering Standards Committee of the IEEE Computer Society

En él se presenta el esquema propuesto por el informe ISO tal como lo reelaboraron (Hirschheim et al., 1995; p.179) en donde se contextualiza el esquema conceptual.



Fuente: (Hirschheim et al., 1995)

Diagrama 8. Contextualización de los esquemas conceptuales en los sistemas de información.

Para elaborar el esquema conceptual del sistema de información de una determinada empresa, se empieza delimitando el 'Universo de Discurso', la parte del mundo real que se quiere modelar. En esta porción de mundo en la que hay distintos entes, individuos y objetos, los empleados de la empresa utilizan diferentes categorías conceptuales mediante las cuales los clasifican. Por ejemplo, supongamos una hipotética empresa de la cual queremos obtener el esquema conceptual. De entre los distintos individuos que tienen alguna relación con la misma, a la empresa le interesa registrar los hechos relacionados con estos individuos. Mediante la operación mental de abstracción, se forman clases con los diferentes individuos utilizando algún criterio. A un grupo de ellos, que tiene una relación de compra de objetos, se les denomina clientes, otros son designados como proveedores, etc. De cada clase formada se abstraen las características comunes a todos los individuos que la componen y que se consideran relevantes para el negocio. Por

ejemplo, para los clientes se consideran relevantes su NIF, nombre, dirección y provincia. Si además se decide que la provincia de residencia del cliente debe ser una de las pertenecientes a una lista preestablecida, por ejemplo las provincias de un determinado país, de las cuales interesa saber su código y nombre, se habrá establecido el esquema conceptual de este pequeño ejemplo, que será:

Clientes(código, nombre, dirección, código de provincia)

Provincias(código, nombre)

En la notación Entity-Relationship (Chen, 1976; Chen, 1977), que es el origen de la mayor parte de las notaciones que se utilizan actualmente, el esquema conceptual del ejemplo anterior se expresa con dos entidades, representadas por rectángulos y una interrelación, que en esta notación se simboliza mediante un rombo. En el Diagrama 9 se presenta el esquema conceptual en esta notación.



Diagrama 9. Esquema conceptual expresado en notación Entity-Relationship.

Una vez establecido el esquema conceptual, cuando se produzca un hecho en la porción del mundo real que hemos modelado, éste se reflejará en un dato que se clasificará según las categorías establecidas en el esquema conceptual y será almacenado en lo que en el diagrama 8 se llama la 'Base de Datos'. Los protocolos y procedimientos para realizar esta tarea forman los que en el diagrama se denomina 'Procesador de Información'. Básicamente, las acciones que se realizan mediante este procesador de información son el registro, la modificación, la consulta y el borrado de los datos.

De las definiciones de esquema conceptual propuestas por la IEEE e ISO que hemos visto en la página anterior se puede inferir lo que muchos autores en esta área asumen implícitamente: que entre el mundo real y el esquema conceptual existe una correspondencia que es objetiva,

independiente de la percepción de las personas. Esta postura es calificada como ‘realismo ingenuo’ (naive realism) por Hirschheim et al. (1995) Otros autores matizan esta definición considerando que la representación de la realidad que supone un esquema conceptual no refleja la realidad sino la percepción que un grupo de personas tiene de la realidad: ‘*A conceptual model is a representation (typically graphical) constructed by IS professionals of someone's or some group's perception of a real-world domain.*’ (Shanks et al., 2003)

Obtener el esquema conceptual de una pequeña aplicación como la que hemos visto en el ejemplo no presenta demasiados problemas. En estos casos normalmente el vocabulario del esquema es el de una única persona; por lo que el contexto es reducido y por tanto no requiere mucho esfuerzo su comunicación. Las dificultades crecen conforme el esquema abarca a distintos departamentos de una empresa, que utilizan los ‘dialectos’ profesionales propios de su función. Aumentan también con el número de personas que interviene en su definición, así como con el número de entidades y atributos que lo forman y con el grado en que éstos estén interrelacionados (Scharer, 1981).

La tarea de obtener un único esquema conceptual a partir de los esquemas parciales que se obtienen en las distintas secciones y departamentos de la organización es conocido como ‘integración de esquemas conceptuales’. A pesar de ser un área de investigación que se remonta por lo menos hasta 1979 (El-Masri and Wiederhold, 1979) y de ser un área muy prolífica en publicaciones, no parece que hayan habido grandes avances. Tal como reflejan Doan et al. (2004) en su introducción al número especial sobre Integración Semántica, de la revista ACM SIGMOD RECORD:

Semantic heterogeneity is one of the key challenges in integrating and sharing data across disparate sources, data exchange and migration, data warehousing, model management, the Semantic Web and peer-to-peer databases.

...Despite its pervasiveness and the substantial amount of work in this area, schema matching today remains a very difficult problem. (Doan et al., 2004; p.11)

Para conseguir la integración conceptual se debe crear un contexto único para todos los conceptos.

*Propuesta de un método de validación de esquemas conceptuales
y análisis comparativo de la noción de información
en los métodos de desarrollo de sistemas de información*

Understanding data about the organisation requires a context for correct interpretation. This context is normally provided by what is usually called business model, data model or conceptual schema. Within these, rules for interpretation are given by structural implications (part of the semantics is the placement in a conceptual structure) and by general textual descriptions of different conceptual components.(Verrijn-Stuart, 2001; p.16)

Los modelos conceptuales atacan este problema mediante la integración de esquemas conceptuales, situando los términos (los conceptos) en un contexto único para toda la organización. Pero esta contextualización, que es necesaria si queremos alcanzar la significación, presenta el grave problema de que, a su vez, la descontextualiza de los dominios de problemas y situaciones que es donde tienen el significado más preciso y exacto, donde en verdad se pueden validar.

El problema de validación de esquemas conceptuales.

El esquema conceptual constituye el núcleo de la especificación cuando se construye software para sistemas de información. Por esto parece lógico asegurarse de que es correcto antes de empezar a construir la máquina de software. Sin embargo la validación de esquemas conceptuales ha sido una tarea poco investigada y muy descuidada en la mayor parte de los métodos propuestos. Por ejemplo, en el diagrama 10 vemos una muestra de cómo se concibe frecuentemente el proceso de diseño de esquemas conceptuales (Navathe, 1992). En este diagrama cabe destacar la ausencia de alguna actividad de validación de los mismos.

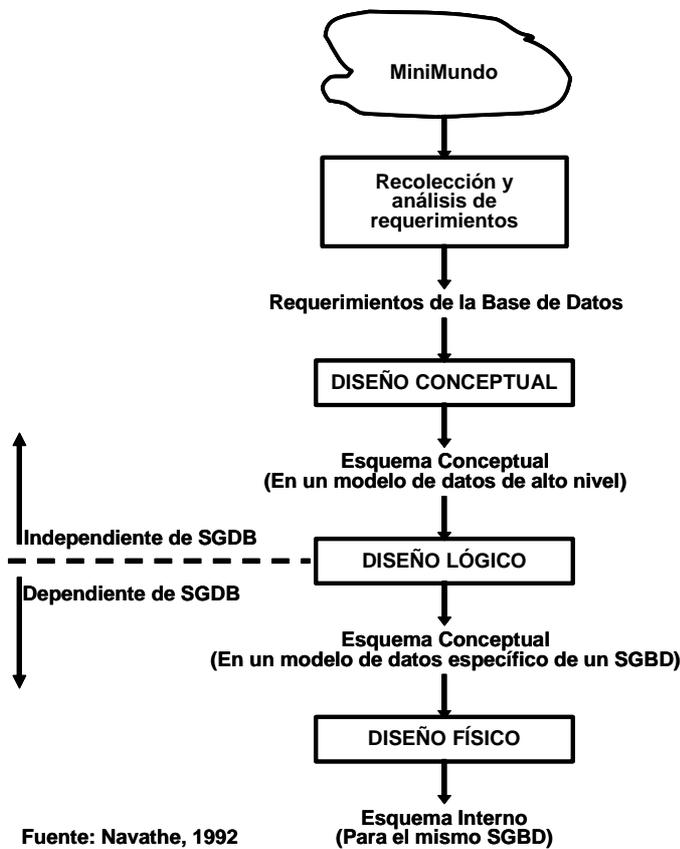


Diagrama 10. Secuencialidad del proceso de diseño de esquemas conceptuales.

Otro ejemplo más claro es el de Booch, Jacobson y Rumbaugh, los autores del método UML, uno de los métodos de mayor prestigio en la actualidad. De la lectura de sus libros se puede inferir que dichos autores presuponen que el significado de los términos es objetivo y tiene la suficiente precisión como para ser ‘capturado’ por los analistas. No es extraño que partiendo de esta postura dediquen tan poca atención a la validación de los requerimientos de información por parte de los usuarios. Por ejemplo, Booch y Rumbaugh ni siquiera mencionan el problema de la validación en sus respectivos libros (Booch, 1994; Rumbaugh et al., 1991). Sólo Jacobson le dedica alguna atención mediante la utilización de su técnica de ‘casos de uso’ (use cases) (Jacobson et al., 1992; pp.313,465). Técnica que posteriormente se incorporó en el método UML (Booch et al., 1999). Sin embargo esta técnica no ataca explícitamente el problema de la validación de los

requerimientos de información, esto es, la verificación de la coherencia de los significados atribuidos a los distintos términos del esquema conceptual por los diferentes usuarios. Es más, en el libro ‘Use Case Modeling’, dedicado íntegramente a estudiar y elaborar dicha técnica, sus autores desaconsejan explícitamente la utilización de la misma para la validación del esquema conceptual, o, como le llaman en dicho libro, modelo de dominio (domain model) (Bittner and Spence, 2003; pp.21-22). Estos autores desaconsejan terminantemente el empleo de dicha técnica para las aplicaciones que denominan ‘*CRUD (Create, Retrieve, Update, Delete behavior)*.’ Esto es, para las aplicaciones, para el software de empresa que hace las tareas de mantenimiento de archivos, que consisten en dar de alta, consultar, modificar o borrar registros de archivos como los de clientes, productos, pedidos, albaranes, facturas, recibos, etc.

Esta omisión es especialmente problemática cuando el método UML proclama que ‘...adequately addresses all scales of architectural complexity, across all domains.’ (UML, 2003; p.1-5) y que ‘...it has proven to be of value in building systems in all sorts of problem domains and encompassing all degrees of size and complexity.’ (Booch et al., 1999; p.11) Esto es especialmente grave si tenemos en cuenta que una gran parte del modelado que se hace en sistemas de información es precisamente del tipo CRUD. El mismo Booch, al referirse a las aplicaciones usuales en empresa afirma:

Applications such as this perform very little computational work. Instead, large volumes of data must be stored, retrieved, and moved about. Most of our architectural work therefore will involve decisions about declarative knowledge (what entities exist, what they mean, and where they are located) rather than procedural knowledge (how things happen). (Booch, 1994; p.381)

Otros autores se atreven a dar una estimación cuantitativa: al menos el 95% de los procedimientos en el software para sistemas de información es de mantenimiento de registros, algorítmicamente muy simple. (Coad and Yourdon, 1991; p.148)

Y que, como afirma Jacobson, no es una tarea trivial el diseñar un esquema conceptual correcto sin tener en cuenta el uso que se hará de él:

In our experience it is hard to define exactly which objects are essential in a particular system without going through, in detail, how the system is to be used. ... We also think that it is extremely hard to tell which attribute relations and objects should have without knowing how the object will be used.

Thus it is natural to model the attributes and the relations after (or while) the operations are defined. (Jacobson et al., 1992; p.483)

Postura que coincide con la de distintas tradiciones filosóficas que reconocen que el significado de las palabras está en su uso. Desde la filosofía pragmática de Peirce: *'...there are symbols, or general signs, which have become associated with their meanings by usage. ... The word lives in the minds of those who use it.'* (Peirce, 1894; p.5) y también, desde la aproximación hermenéutica, Dilthey: *'Linguistic usage is the basis of interpretation,...'* (Dilthey, 1900; p.81), y Wittgenstein: *'For a large class of cases --though not for all-- in which we employ the word "meaning" it can be defined thus: the meaning of a word is its use in the language.'* (Wittgenstein, 1953; p.20)

Giddens, desde el ámbito de la sociología, también afirma: *'Meaning in communicative acts, as it is produced by lay actors, cannot be grasped simply in terms of a lexicon, any more than it can be transcribed within frameworks of formal logic that pay no attention to context-dependence.'* (Giddens, 1976; p.25)

Si aceptamos que el significado de los términos viene condicionado por su utilización concluiremos que la validación de los requerimientos de información es crítica en el desarrollo de sistemas de información para empresa. Sin embargo es un problema que ha sido poco estudiado (Choobineh and Lo, 2004). Scheer y Hars (1992) reflexionando sobre la problemática de la validación de los esquemas conceptuales afirman que los métodos no están maduros y que se requiere más investigación en dicha área. Una década después, (Shanks et al., 2003; p.86) constatan que *'Unfortunately, little is known about how to validate conceptual models effectively and efficiently.'*

Los pocos autores que tratan el problema (Connolly and Begg, 2005; Shanks et al., 2003; Kim and March, 1995; Loucopoulos and Karakostas, 1995; Bubenko, 1977) en general ofrecen pocas guías de cómo debe llevarse a cabo. Las dos maneras de validación del esquema conceptual más citadas son la revisión del mismo por parte de los usuarios y la descripción de las transacciones.

La primera manera de validar consiste en que los usuarios comprendan el significado del esquema e identifiquen las discrepancias entre éste y la realidad tal como la perciben. Tal como lo defienden algunos autores:

*Propuesta de un método de validación de esquemas conceptuales
y análisis comparativo de la noción de información
en los métodos de desarrollo de sistemas de información*

Semantic data models were developed to provide a precise and unambiguous representation of organizational information requirements. They serve as a communication vehicle between analysts and users. (Kim and March, 1995; p.103)

Sin embargo, este procedimiento presenta algunos inconvenientes. El primero es que el usuario debe familiarizarse con la notación con la que está especificado el esquema conceptual. Lo cual, aunque se utilicen notaciones gráficas como la del método Entity-Relationship visto anteriormente, no siempre es fácil, sobre todo si se quiere conseguir que el usuario entienda, con el nivel adecuado de precisión, el alcance de cada símbolo utilizado. Otro inconveniente más grave es que los usuarios no perciben su universo de discurso al nivel de agregación en que se presenta en los esquemas conceptuales. Los usuarios utilizan el sistema de información, interaccionan con el esquema conceptual, no al nivel de conceptos básicos y generales para toda la empresa sino en el de transacciones de empresa (Smith and Smith, 1977). Por ejemplo, una transacción de venta, es percibida por el usuario como un todo, como una factura. Sin embargo, una factura normalmente produce varias entidades e interrelaciones en el esquema conceptual.

Otro inconveniente de este procedimiento de validación es que el esquema conceptual constituye sólo la mitad de la especificación necesaria para construir el sistema de información. Especifica solo los datos pero no las acciones o funciones que se realizan sobre los mismos. Según las palabras del propio creador del método Entity-Relationship Peter Chen, *I suggest that in the future, we can take the ER concept and add something onto it so that it becomes a unified modeling technique. Then you will have one technique that can be applied to both structural and functional modeling.* (Winslett, 2004; p.115)

Yourdon, aun defendiendo los métodos de modelado conceptual, abunda en la misma crítica al señalar que el esquema conceptual —el modelo de la información, como lo llama él— únicamente modela la mitad del mundo real. (Coad and Yourdon, 1991; p.27)

Frente a la posibilidad de completarlo con otra técnica que enfatice las operaciones sobre los datos, como el Análisis Estructurado con su diagrama de flujo de datos (DFD), del cual dicho autor es uno de los creadores, afirma que, aunque es una idea académicamente atractiva, en los libros que la aplican el acoplamiento entre los dos métodos es muy

débil, y cita como ejemplo su propio libro (Yourdon, 1989), enfatizando además que en la práctica de cada día este acoplamiento entre los dos métodos es inexistente. (Coad and Yourdon, 1991; pp.24-25)

La segunda manera de validar el esquema conceptual propuesta por Connolly y Begg es el 'test de las transacciones'. El objetivo de esta técnica es la comprobación de que el esquema conceptual soporta las transacciones que los usuarios harán sobre él. Estos autores entienden por transacciones de los usuarios exactamente lo que hemos calificado anteriormente como operaciones de mantenimiento de archivos, las acciones de tipo CRUD.

The objective of this step is to check the model to ensure that the model supports the required transactions. Using the model, we attempt to perform the operations manually. If we can resolve all transactions in this way, we have checked that the conceptual data model supports the required transactions. (Connolly and Begg, 2005; p.456)

Parece obvio que antes de desarrollar el software se compruebe que el esquema conceptual es capaz de soportar las transacciones que se construirán sobre él. No obstante, sin negar la obviedad, la propuesta de estos autores presenta tres limitaciones. La primera es que presentan las transacciones de manera descriptiva, no sistemática. La segunda, posiblemente por causa de la primera, no proponen ninguna manera de realizar la tarea. Y, la tercera, que asumen que la validación la realiza el analista, por lo cual más que una validación es una verificación. Entendiendo estos dos términos en el sentido que propone Boehm (1984) 'Validation: Am I building the right product? Verification. Am I building the product right?' Asumiéndolo en este sentido, quien debe validar, declarar válido el sistema, el producto que se le ofrece, es el usuario.

Una aproximación al problema mediante formularios de empresa.

En este trabajo se propone explorar la utilización de los formularios de empresa (documentos preimpresos como las facturas, albaranes, pedidos, recibos, tickets de venta, partes de producción, etc) como medio de validación del esquema conceptual, así como de especificación del sistema de información.

*Propuesta de un método de validación de esquemas conceptuales
y análisis comparativo de la noción de información
en los métodos de desarrollo de sistemas de información*

Los formularios son el instrumento básico que emplean las empresas y organizaciones para coordinar y controlar las operaciones diarias, tanto las internas de la empresa como las externas con sus proveedores, clientes, instituciones gubernamentales y otras entidades. (Choobineh et al., 1992). Este hecho de que los formularios sean utilizados intensivamente en las empresas hace que los usuarios estén muy familiarizados con ellos por lo que los convierte en un vehículo adecuado para que comuniquen los requerimientos del sistema y también para que lo validen.

A pesar de las ventajas que parece que presentan los formularios, la investigación publicada sobre este enfoque es muy escasa. En las búsquedas hacia atrás, las referencias que se encuentran en los artículos, y hacia delante, realizadas en las bases de datos digitales de ACM, IEEE e ISI Web of Knowledge se han encontrado los siguientes artículos: (Hammer et al., 1977), (Rowe and Shoens, 1979), (Hammer and Berkowitz, 1980), (Rowe and Shoens, 1982), (Tsichritzis, 1982), (Rowe and Shoens, 1983), (Yao et al., 1984), (Rowe, 1985), (Mannino et al., 1986), (Tseng and Mannino, 1988), (Choobineh et al., 1988), (Embley, 1989), (Choobineh et al., 1992), (Mfourga, 1997), (Kim and Park, 1997), (Adam and Gangopadhyay, 1997), (Wu et al., 2004), (Choobineh and Lo, 2004).(Hammer and Berkowitz ,1980)

En ninguno de estos artículos se presenta una notación para la especificación de los formularios con el propósito de la validación del esquema conceptual. Exceptuando los trabajos de Choobineh y Mannino, que se basan en los formularios para la elicitación de requerimientos que permitan especificar el esquema conceptual, y en el artículo de Wu, que propone la utilización del formulario para el diseño de sistemas de apoyo a la toma de decisiones (DSS: Decision Support Systems), en el resto de los artículos se proponen sistemas de desarrollo de aplicaciones informáticas sin presentar ninguna notación para expresar los formularios.

En el próximo capítulo se presenta una notación que permite expresar los formularios de empresa mediante fórmulas. Fórmulas que son deducidas del esquema conceptual mediante un conjunto de reglas de inferencia que se proponen. Esto dota a la notación de una estructura similar a la de un cálculo lógico en el que el esquema conceptual es el

conjunto de axiomas y los formularios, las fórmulas que los expresan, son los teoremas deducibles de los axiomas.

El enfoque axiomático

El enfoque axiomático ya está implícito en todas las técnicas de modelado conceptual, en la de análisis y diseño orientado a objetos y en el de las ontologías. Basta con que recordemos la definición que se propone en ISO de que el esquema conceptual es ‘una colección coherente de oraciones que expresan las proposiciones necesarias que son válidas en un universo de discurso’ y la comparemos con lo que se entiende por axioma ‘proposición tan clara y evidente que se admite sin necesidad de demostración’ (DRAE), esto es: proposición que se asume como necesariamente válida.

Esto puede explicar el éxito de estas aproximaciones al problema de especificación de sistemas de información. La axiomatización es un modo muy útil de sistematización, de reducción de la complejidad cognitiva, de los sistemas a analizar. Las ventajas del método axiomático en ciencia las expone Mario Bunge en su libro Filosofía de la física. Aunque la cita es demasiado larga, la presentamos con las propias palabras del autor:

“Antes de Euclides (fl. 300 a. de C.) hubo opiniones y doctrinas, pero hasta donde alcanza la prueba documental existente apenas existió teoría en el sentido moderno, con la sola excepción de la teoría de las proposiciones de Eudoxo. Esto es, había enunciados más o menos laxamente conectados y no sistemas hipotético-deductivos, esto es, sistemas basados en suposiciones iniciales explícitamente enunciadas (llamadas también axiomas o postulados). Ciertamente, la noción de prueba se inventó un par de siglos antes (probablemente en la escuela pitagórica) y la necesidad de establecer algunas suposiciones a fin de probar algo fue reconocida junto con el concepto de consecuencia lógica. Pero las pruebas estaban tan aisladas como las propias premisas: ninguna de las dos era sistémica. Fue el reino feliz del solucionador de problemas más que el de constructor de teorías: los problemas se abordaban uno tras otro y se resolvían siendo cualesquiera premisas que pareciesen prometedoras, sin cuidarse mucho de su circularidad o coherencia, menos aún de su homogeneidad (la pertenencia a un conjunto de items mutuamente relacionados).”

*Propuesta de un método de validación de esquemas conceptuales
y análisis comparativo de la noción de información
en los métodos de desarrollo de sistemas de información*

El sistema de geometría que Euclides expuso en sus Elementos no consistió sólo en un apilamiento de recetas de cálculo (como casi toda la matemática sumeria y egipcia). El sistema de Euclides fue incluso más que una vasta colección de piezas de conocimiento matemático: fue probablemente la primera teoría plenamente elaborada inventada por la humanidad. Empequeñecer a Euclides diciendo que “sólo” codificó el conocimiento matemático de sus días es revelar una pobre comprensión de la naturaleza e importancia de las teorías. Es más, Euclides formuló su teoría geométrica del modo más completo y coherente posible en su tiempo —a saber, del modo axiomático que él mismo introdujo. La lección de Euclides en la metodología de la construcción de teorías fue entonces ésta: Si nos interesa la sistematicidad y el rigor, intentemos el método axiomático.” (Bunge, 1973; p.157)

Basta ojear el libro de Euclides, Los Elementos (Euclid, 300 B.C.), para poder apreciar no sólo el método axiomático sino también el rigor científico de su exposición. Intentar alabar a Euclides diciendo que aplicó el método científico igual que se hace en la actualidad es minusvalorar al autor, es tergiversar la realidad. Son los científicos actuales los que hacen ciencia tal como la hizo Euclides, tal como nos enseñó. En cuanto al método científico somos griegos.

La brillantez y la apariencia de verdad apodíctica de la geometría confundió a muchos filósofos (los racionalistas) que asumieron que se podría descubrir la realidad del mundo, o al menos de una parte de él, utilizando únicamente la razón, mediante el método deductivo propio de las matemáticas. La geometría es uno de los ejemplos que utilizó Kant para mostrar la existencia de los juicios sintéticos a priori, esto es, verdades que reflejan el mundo real (sintéticos) pero que se descubren utilizando únicamente la razón (a priori). Incluso un filósofo tan antirracionalista como Hume, que distinguió entre las verdades que son meramente una ‘relación entre ideas’, ciertas únicamente debido a los significados atribuidos a las palabras (lo que Kant llamó juicios analíticos) y las ‘cuestiones de hecho’ que son datos de la realidad (los juicios sintéticos de Kant), y que consideraba las matemáticas como meras ‘relaciones de ideas’, presenta indicios de perplejidad cuando considera la geometría.

Siglos después, Einstein también se pregunta: *‘How can it be that mathematics, being after all a product of human thought which is independent of experience, is so admirably appropriate to the objects of reality? Is human reason,*

then, without experience, merely by taking thought, able to fathom the properties of real things.' (Einstein, 1922; p.28) Y concluye: *'In my opinion the answer to this question is, briefly, this: --As far as the laws of mathematics refer to reality, they are not certain; and as far as they are certain, they do not refer to reality.'* (Ibid.; p.28)

La geometría como ciencia que 'descubre' la realidad del mundo mediante el uso de la razón (sintética a priori) fue refutada por Hilbert a principios del siglo pasado. Hilbert, mediante la desinterpretación de la geometría, demostró que los axiomas de Euclides aunque en apariencia son claros y evidentes no por ello son necesariamente ciertos. En realidad existen muchas geometrías posibles, por ejemplo la geometría que utiliza la teoría general de la relatividad no es euclidiana. El problema de utilizar una u otra depende del ámbito de problemas que queramos estudiar.

Einstein nos presenta esta interpretación moderna, hilbertiana, de la geometría, en la que las líneas rectas, los puntos, etc. no conllevan ningún conocimiento o intuición de la realidad, solo presupone los axiomas *'which are to be taken in a purely formal sense, i.e. as void of all content of intuition or experience. These axioms are free creations of the human mind. All other propositions of geometry are logical inferences from the axioms (which are to be taken in the nominalistic sense only).* (Ibid.; p.30)

Sin embargo, aun despojado de su carácter metafísico, el método axiomático presenta grandes ventajas. *'...The progress achieved by axiomatics consists in its having neatly separated the logical-formal from its objective or intuitive content; according to axiomatics the logical-formal alone forms the subject-matter of mathematics, which is not concerned with the intuitive or other content associated with the logical-formal.'* (Ibid.; p.30)

Visto desde la perspectiva axiomática, los enfoques de modelado conceptual, así como los de análisis orientado a objetos y los ontológicos, son de axiomatización ingenua, euclídea. Esto es patente si tenemos en cuenta que o bien omiten la fase de validación del esquema conceptual, o intentan validarlo directamente.

Lo que se presenta en el próximo capítulo es una propuesta axiomática, de modo que el esquema conceptual sea validado a través de los formularios de empresa.

La axiomatización moderna se expresa mediante cálculos lógicos. Un cálculo lógico está formado por un vocabulario, una sintaxis que permite construir fórmulas, un subconjunto de fórmulas elegido que son llamados axiomas y un conjunto de reglas de deducción, que permite transformar fórmulas. Las fórmulas obtenidas mediante la transformación de los axiomas se llaman teoremas.

En el próximo capítulo se presenta una notación formal que expresa formularios de empresa. En nuestro caso los axiomas son el esquema conceptual y las fórmulas que expresan los formularios son los teoremas.

Capítulo 3: Especificación de sistemas de información

En este capítulo se presentan las características de la notación que proponemos para representar mediante fórmulas los formularios de empresa. Está basada en el modelo de datos relacional (Codd, 1970; Codd, 1982) y puede ser considerada como una ampliación del mismo que subsana una limitación que éste comparte con los métodos de modelado de datos. Esta limitación es, tal como señalan Coad y Yourdon (1991) y el creador del modelo de datos Entity-Relationship Peter Chen (Winslett, 2004), que dichos métodos sólo modelan ‘la mitad del mundo’, los datos, pero no las operaciones de actualización de los mismos tal como las realizan los usuarios de empresa.

La notación que proponemos está construida como una extensión del modelo de datos relacional, por la que aprovecha las ventajas del mismo y de los modelos de datos semánticos. Esto es, no sustituye el ámbito de problemas que resuelven dichos métodos sino que los complementa permitiendo una validación de los esquemas conceptuales más adecuada.

En este capítulo, en primer lugar se presenta brevemente el modelo de datos relacional, definiendo la notación que emplearemos para expresar el esquema conceptual. A continuación, se hace una presentación informal de la notación que proponemos para expresar los formularios de empresa a partir del esquema conceptual. Posteriormente, se presenta la notación a la manera de un cálculo formal. Después, para mostrar la capacidad expresiva de la notación propuesta, se realiza un ejemplo completo de análisis y diseño conceptual en el caso de una aplicación sencilla de gestión de pedidos de una hipotética empresa. Por último, se hace una valoración del método que proponemos.

La notación que proponemos permite especificar formalmente el sistema de información que se quiere construir, y también validar que el esquema conceptual de datos que se ha definido es completo, esto es, que a partir de él se pueden definir todas las transacciones de empresa tal y como los usuarios requieren.

El modelo de datos relacional

El modelo de datos relacional (Codd, 1970) es el más conocido de todos los modelos de datos que han sido propuestos en las últimas décadas. La casi totalidad de los sistemas de gestión de bases de datos que son utilizados actualmente en las empresas y organizaciones están basados en dicho modelo. Ha sido presentado en un gran número de libros y artículos explicándolo. Por este motivo, en esta sección se hace una presentación breve e informal en la que se resaltan las características que son relevantes para definir la notación que proponemos. Se puede encontrar una especificación formal del modelo de datos relacional en (Date, 1982).

Un modelo de datos tiene tres componentes: un conjunto de tipos de estructuras de datos, un conjunto de operadores que permiten transformar los datos, y un conjunto de reglas de integridad que especifican las transformaciones válidas de los datos. (Codd, 1981) (Date, 1983; p.181)

El modelo de datos relacional es una manera de representar el conjunto de datos —el conjunto de archivos— que utiliza una organización. El elemento básico en este modelo es la relación, que sustituye a los archivos o ficheros de los sistemas convencionales. Según dicho modelo, una relación R es un subconjunto del producto cartesiano de varios conjuntos D_1, D_2, \dots, D_n .

$$R \subset D_1 \times D_2 \times \dots \times D_n$$

Las relaciones se especifican presentándolas en forma de tabla. Por ejemplo, la relación de clientes de una determinada empresa puede ser

Clientes

codCliente	Apellido	codProvincia
1	Pérez	S

2	Puig	B
3	Coll	B
4	Aguirre	Ma

En este ejemplo la relación se llama Clientes. El dominio D_1 sería el conjunto de los números naturales; el D_2 , el conjunto de los apellidos; y el D_3 , el conjunto de códigos de provincia de las antiguas matrículas de coche. Los dominios D_1 y D_3 no presentan problemas de definición, sin embargo con el D_2 , el de apellidos, dado que en la práctica es imposible definir este conjunto, normalmente se recurre a establecer que dicho dominio está formado por el conjunto de los textos alfabéticos de menos de un número determinado de caracteres.

En la primera fila de la tabla se especifican los nombres de las columnas, también llamados *atributos* de la relación. Los atributos representan el uso que se hace de un dominio en determinada relación. Su nombre no tiene que coincidir con el del dominio; además, distintos atributos pueden corresponder a un mismo dominio.

Para especificar una relación se deben declarar los atributos que la componen, las claves que tiene y los dominios de los atributos. Esta especificación se llama el esquema de la relación.

Se define como clave de una relación el conjunto de atributos de la misma que tiene dos propiedades, la unicidad y la minimalidad. La propiedad de unicidad establece que no puede haber dos filas de la relación cuyos atributos que forman la clave tengan el mismo valor. La propiedad de minimalidad establece que no se puede suprimir ningún atributo de la clave sin destruir la propiedad de unicidad. Una relación puede tener varias claves. De entre todas ellas se elige una, a la que se le llama 'clave primaria'. El resto de claves se llaman 'claves alternativas'.

Además de proponer una manera de estructurar los datos, el modelo relacional establece dos reglas de integridad que deben cumplir los datos: la integridad de entidades y la integridad referencial. La regla de integridad de entidades establece que los valores de un atributo que forma parte de una clave primaria no pueden ser nulos (se dice que un valor es nulo cuando no existe).

Para establecer la segunda regla de integridad se ha de definir una nueva noción, la de ‘clave externa’. Se dice que un conjunto de atributos de una relación R es una clave externa de otra relación S , si en esta última es una clave primaria. Con esta definición se puede expresar la regla de integridad referencial: los valores de una clave externa deben existir en la relación donde son clave primaria o ser nulos.

En su modelo, Codd, además de la estructura y las reglas de integridad, define dos familias de operadores que permiten transformar las tablas, el álgebra relacional y el cálculo relacional. No se presentan aquí pues no son relevantes para nuestro propósito por dos razones. La primera es debida a que la finalidad del álgebra y cálculo relacional es permitir interrogar las bases de datos (esto es, son lenguajes de interrogación de la base de datos), mientras que nuestro propósito es caracterizar fórmulas que permitan delimitar las transacciones correctas (que no violen las reglas de integridad) para actualizar las bases de datos. En el próximo capítulo se explica con algo más de precisión esta diferencia haciendo referencia a la distinción que se hace en un cálculo lógico entre fórmulas satisfacibles y fórmulas válidas. La segunda razón es debida a que los operandos del álgebra y cálculo relacional son relaciones (que tienen estructura de tabla) al igual que los resultados de las operaciones, mientras que a nosotros nos interesa caracterizar formularios que tienen una estructura más compleja que la de tabla.

Codd, además de definir un modelo de datos, prescribió un criterio de decisión que permite determinar lo que es un buen diseño de una base de la base de datos. Además, propuso un procedimiento que permite obtener los buenos diseños. Este procedimiento está basado en la noción de dependencia funcional. Dados dos conjuntos de atributos A y B de una relación, se dice que B depende funcionalmente de A si existe una función matemática que hace corresponder los valores de A en B . Esto es, B depende funcionalmente de A si no puede darse el caso que el mismo valor de A tenga dos imágenes.

Este procedimiento, que llamó de ‘normalización’ de las relaciones, consiste en transformar las relaciones de tal modo que capturen todas las dependencias funcionales mediante las claves de las relaciones. Estas transformaciones de las relaciones se consiguen sin pérdida de información. Lo que quiere decir que a partir de las relaciones normalizadas se pueden obtener las relaciones originales. Una ventaja de

la normalización es que consigue eliminar todos los datos redundantes, lo que supone una gran ventaja en el diseño conceptual de bases de datos (Codd, 1971).

A continuación se presenta la notación que emplearemos para especificar las relaciones, los atributos, sus dominios y las claves (primarias, alternativas y externas).

Presentación informal de la notación

Especificación de relaciones

En la notación que proponemos, una relación (abusando del lenguaje llamaremos relación al esquema de la relación) se especifica indicando su nombre, los atributos que la constituyen, los dominios de los mismos y las claves de la relación.

Especificación de atributos

Se indican escribiendo el nombre de la relación y a continuación, entre paréntesis, los nombres de los atributos separados por comas.

Por ejemplo, para especificar una relación de clientes que tenga como atributos el código, el nombre del cliente y el código de la provincia en que reside pondremos

cli(cod,nom,cprv)

Las características de la relación las pondremos a continuación separadas por el símbolo ':' (dos puntos).

Las características son las claves y los dominios. Los separaremos entre ellos por el símbolo ';' (punto y coma).

Especificación de claves

Las claves se indican poniendo una 'K' y entre paréntesis la lista de claves, siendo la primera de ellas la clave primaria y las otras las claves alternativas.

En el ejemplo de clientes pondríamos

cli(cod,nom,cprv) : K(cod)

Si se diese la circunstancia que en nuestra empresa no debiese haber más de un cliente en cada provincia, entonces cprv sería clave alternativa y lo indicaríamos de la siguiente manera

cli(cod,nom,cprv) : K(cod, cprv)

En el caso de que una clave esté formada por más de un atributo se escriben éstos entre paréntesis y separados por comas, por ejemplo en una relación facturas la clave primaria puede ser el número de factura y, con la suposición de que nuestra empresa nunca haga más de una factura por día al mismo cliente, también formarán una clave los dos atributos código de cliente y fecha de factura. Esto se expresará escribiendo

fac(num,ccli,fecha,dto) : K(num, (ccli,fecha))

En el caso de una relación de componentes que tenga los atributos cpac, cmat, Q, que indica las unidades Q de la materia prima cmat que hay en el producto cpac, y en la que la clave primaria está formada por los dos atributos cpac y cmat escribiremos

compo(cpac,cmat,Q) : K(cpac,cmat)

Especificación de dominios

La especificación de dominios se establece escribiendo D o DOMS y a continuación entre paréntesis las características de cada dominio.

Dominios alfanuméricos

Si los dominios son caracteres alfanuméricos lo especificaremos poniendo un número y a continuación la letra A, el número indicará el número de caracteres que puede tener como máximo dicho atributo. Por ejemplo en la relación clientes pondremos

cli(cod,nom,cprv) : K(cod) ; D(4 A, 20 A, 2 A)

con esto indicamos que el código de cliente está formado por cuatro caracteres alfanuméricos, que el nombre de cliente tiene como máximo 20 caracteres alfanuméricos y que el código de provincia está formado por 2 caracteres alfanuméricos.

Dominios numéricos

Para declarar un atributo con dominio numérico (lo haremos cuando queramos hacer operaciones aritméticas con dicho atributo) se escribe:

- un número que indica el tamaño del campo

- una N para indicar numérico
- otro número inmediatamente después de la N para indicar el número de decimales.

por ejemplo para un atributo precio pondríamos 6 N0, para uno de descuento (en tanto por uno) 4 N2, con lo que indicaríamos un campo con 4 espacios, numérico y con dos decimales x,xx ej. 0,20

En la relación componentes se podría poner, por ejemplo,

compo(cpac,cmat,Q) : D(4 A, 4 A, 2N0) ; K(cpac,cmat)

Como se ve, se puede poner antes la D y luego la K o al revés.

Claves externas

Para especificar que un atributo de una relación es clave externa, esto es, que sus valores deben existir en la clave primaria de otra relación, en lugar de escribir la longitud del atributo y su naturaleza (ej. 4 A) se escribe el símbolo \subset y a continuación el nombre del atributo en el que debe comprobar la existencia de los valores.

Por ejemplo, en el esquema

prv(cod,nom) : K(cod); D(2 A, 10 A)

cli(cod,nom,cprv) : K(cod); D(4 A, 12 A, 2 A)

que representa la relación de provincias, código y nombre, y la relación de clientes. cprv.cli es clave externa pues los únicos códigos de provincia permitidos serán los que existan en la relación prv; en este caso en lugar de poner el dominio de cprv.cli 2 A, deberemos poner \subset cod.prv. El símbolo \subset debe leerse como 'contenido en'. En el ejemplo leeremos que el dominio de cprv.cli está 'contenido en' cod.prv. Si se declara esta restricción, no podrá haber valores de cprv.cli que no existan en cod.prv. Esto quiere decir que se podrá dar de alta una provincia aunque no exista clientes en la misma, pero no se podrá dar de alta a un cliente con un valor de provincia que no exista en la relación provincias.

El esquema correcto será

prv(cod,nom) : K(cod); D(2 A, 10 A)

cli(cod,nom,cprv) : K(cod); D(4 A, 12 A, \subset cod.prv)

Imaginemos ahora que en nuestra empresa exista la norma de que no puede haber ninguna provincia sin cliente. Esto podría tener sentido si,

por ejemplo, nuestros clientes fueran distribuidores y nuestra empresa quisiera tener por lo menos un distribuidor en cada provincia; esto significaría que, además de que todos los valores de $\text{cprv.cli} \subset \text{cod.prv}$, se debería cumplir

$\text{cod.prv} \subset \text{cprv.cli}$, lo que expresaremos poniendo

$\text{cprv.cli} = \text{cod.prv}$. A este tipo de clave externa le llamaremos ‘clave externa por igual’.

El esquema, con esta nueva norma de la empresa quedaría

$\text{prv}(\text{cod}, \text{nom}) \quad : \text{K}(\text{cod}); \text{D}(2 \text{ A}, 10 \text{ A})$
 $\text{cli}(\text{cod}, \text{nom}, \text{cprv}) \quad : \text{K}(\text{cod}); \text{D}(4 \text{ A}, 12 \text{ A}, = \text{cod.prv})$

Especificación de formas visuales básicas

Supongamos la relación

$\text{cli}(\text{cod}, \text{nom}, \text{cprv}) : \text{K}(\text{cod}); \text{D}(4 \text{ A}, 12 \text{ A}, 2 \text{ A})$

en donde cod es el código de cliente y clave primaria; nom , el nombre del mismo y cprv , el código de la provincia en que reside

De esta relación se pueden deducir dos tipos de forms diferentes por su forma :

1FV : Primera Forma Visual

$\text{cliente} \{ \text{cod.cli} \rightarrow \{ \text{nom.cli}, \text{cprv.cli} \} \}$

que genera la pantalla

CLIENTE	
cod.cli	:
nom.cli	:
cprv.cli	: ..

2FV : Segunda Forma Visual

$\text{prvcli} \{ \text{cprv.cli} \rightarrow \{ \text{cod.cli}, \text{nom.cli} \} \}$

genera la pantalla

PRVCLI	
cod.prv	: ..
<hr/>	
<u>cod.cli</u>	<u>nom.cli</u>
....
....
....

Esta pantalla permite hacer el mantenimiento de clientes agrupados por la provincia a la que pertenecen.

Se llama ‘identificador’ del form al conjunto de atributos que quedan a la izquierda del símbolo de función ‘→’ o ‘→>’. Cuando el identificador del form coincide con una clave de la relación de la que se deduce, entonces se deduce un form en primera forma visual. Cuando el identificador no coincide con una clave se deduce un form en segunda forma visual.

Lista de formas visuales básicas

Añadiendo el símbolo →> a la izquierda de un form se consigue una lista del mismo.

Por ejemplo `clientes { →> { cod.cli → { nom.cli, cprv.cli } } }`

genera el documento

CLIENTES		
<u>cod.cli</u>	<u>nom.cli</u>	<u>cprv.cli</u>
....
....
....
....

Si se quiere repetir un form previamente definido, para abreviar la escritura se puede utilizar el símbolo ‘^’ seguido de nombre del form :

`clientes { →> ^cliente }`

En general, siempre que para definir un form debamos utilizar otro form previamente definido podremos utilizar este método que aumenta la legibilidad de los forms complejos.

También se puede definir un nuevo form como repetición de la segunda forma visual :

lprvcli { →> ^prvcli }

LPRVCLI		
<u>cod.prv</u>		
..	<u>cod.cli</u>	<u>nom.cli</u>

..	<u>cod.cli</u>	<u>nom.cli</u>

..	<u>cod.cli</u>	<u>nom.cli</u>

Especificación conceptual del sistema de información

Un modelo de un S.I. se define especificando el esquema conceptual de datos y el esquema conceptual de acciones del S.I. El primer esquema está formado por el conjunto de relaciones normalizadas, es segundo es la especificación de los forms, esto es, la especificación de los programas mediante los cuales podemos consultar y actualizar los datos.

La sintaxis es la siguiente :

<Nombre del S.I.> ESQUEMA [<declaración de relaciones>]

FORMS [<declaración de forms>]

Con el ejemplo de la relación de clientes, la especificación del modelo es:

empresa

ESQUEMA [

cli(cod,nom,cprv) : K(cod); D(4 A, 20 A, 2 A)

]

FORMS [

cliente { cod.cli \rightarrow { nom.cli, cprv.cli } }

clientes { $\rightarrow\rightarrow$ ^cliente }

prvcli { cprv.cli $\rightarrow\rightarrow$ { cod.cli, nom.cli } }

]

Definiciones.

Definición: Form base

Dada la relación $r(X,Y,Z,W)$ en donde X,Y,Z,W son conjuntos de atributos (no necesariamente un único atributo).

Entonces:

a) $\{ X \rightarrow Y \}$ es un form base si:

- X es clave (no necesariamente primaria) e Y es cualquier conjunto de atributos (uno o más) de la relación que no están en X .

Y puede ser el conjunto vacío (ningún atributo) y en este caso poner $\{ X \}$ es equivalente a poner $\{ X \rightarrow \}$

b) $\{ Z \rightarrow\rightarrow W \}$ es un form base si:

- Z no es clave y W son atributos (uno o más) que no están en Z .

W puede ser el conjunto vacío, en cuyo caso se escribe $\{ Z \rightarrow\rightarrow \}$, o simplemente $\{ Z \}$.

Ejemplos: Dada una relación $r(a,b,c,d,e)$ en donde a es la clave primaria y c y (d,e) son claves alternativas

$r(a,b,c,d,e) : K(a, c, (d,e))$

presentamos a continuación algunos de los forms base que se pueden deducir, según la definición anterior, de la relación.

$\{ a.r \rightarrow b.r \}, \{ a.r \rightarrow c.r \}, \{ a.r \rightarrow \{ b.r, c.r \} \}, \{ c.r \rightarrow a.r \}, \{ c.r \rightarrow d.r \},$
 $\{ c.r \}, \{ c.r \rightarrow \{ b.r, c.r, d.r \} \}, \{ \{ d.r, e.r \} \rightarrow c.r \}, \{ b.r \twoheadrightarrow a.r \}, \{ b.r$
 $\twoheadrightarrow \{ a.r, b.r, c.r \} \}, \{ b.r \}$

Definición: Identificador de form

Los atributos que están a la izquierda del símbolo de dependencia de un form base, funcional \rightarrow o multifuncional \twoheadrightarrow , se llaman ‘identificadores’ de form, los que quedan a la derecha son los ‘identificados’.

Definición: Form íntegro

Un form base que contenga la clave primaria es un form íntegro.

Para que un form se pueda utilizar para actualizar la base de datos es necesario (pero como veremos más adelante no suficiente) que el form sea íntegro.

Definición: Form completo

Un form base que contenga todos los atributos de la relación de la que se deduce es un form completo.

Composición de forms: forms compuestos.

Supongamos un esquema formado por la relación cli y una nueva relación prv que contiene los códigos y nombres de las provincias

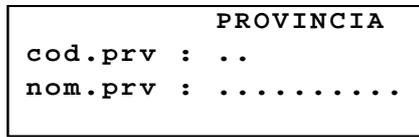
$prv(cod,nom) \quad : K(cod); D(2A, 10A)$
 $cli(cod,nom,cprv) \quad : K(cod); D(4A, 12A, \subset cod.prv)$

Como hemos visto, a partir de estas relaciones podemos definir los forms:

cliente $\{ cod.cli \rightarrow \{ nom.cli, cprv.cli \} \}$ que especifica la pantalla

CLIENTE	
<code>cod.cli</code>	<code>:</code>
<code>nom.cli</code>	<code>:</code>
<code>cprv.cli</code>	<code>: ..</code>

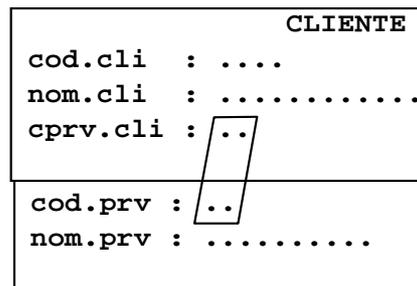
y **provincia** $\{ cod.prv \rightarrow nom.prv \}$ que genera



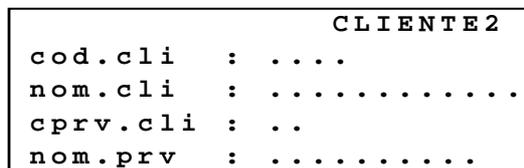
Para producir un documento que contenga la información del form cliente y del form provincia conjuntamente, podemos fusionar los dos documentos mediante el operador conexión que se representa por el símbolo '='.

```
cliente2 { cod.cli
    → { nom.cli, { cprv.cli = { cod.prv → nom.prv } } }
}
```

que genera el documento



En el que los atributos cprv.cli y cod.prv se superponen, apareciendo en el documento únicamente el primero



Estos dos atributos se llaman 'conectores', el que está a la izquierda del símbolo '=' es el 'conector principal' y el de la derecha, el 'conector subordinado'. En el ejemplo, cprv.cli y cod.prv respectivamente.

Para evitar el excesivo uso de llaves ({ }) se establecen precedencias de los símbolos. De mayor precedencia a menor, son:

1. '{', '}'
2. '^'
3. '→', '→>'
4. '='
5. ','

A igualdad de precedencia, ésta se establece de izquierda a derecha.

Aplicando estas reglas de precedencia, queda el form

cliente2 { cod.cli → { nom.cli, cprv.cli = cod.prv → nom.prv } }

Cuando se establece una conexión entre dos forms, se conectan los identificadores de los forms (los atributos que quedan a la izquierda del símbolo de función, '→' o '→>').

En este ejemplo anterior se ha hecho una conexión de un atributo **cprv.cli** con un form **cod.prv → nom.prv** (un único atributo también es un form). Es este caso **cprv.cli** se lee como el form **{cprv.cli →> }**.

En el caso **a.r → b.r = x.s →> y.s**, dadas las precedencias establecidas anteriormente, se está conectando el form **{a.r → b.r}** con **{x.s →> y.s}**. Esto es, se superpone el identificador del primero, que es **a.r**, con el del segundo, **x.s**. Si quisiéramos conectar el atributo **b.r** con el segundo form deberíamos utilizar las llaves para modificar la prioridad de asociación:

{a.r → {b.r = x.s →> y.s } }

Veamos más ejemplos de forms que pueden ser deducidos del mismo esquema de provincias y clientes.

PRVCLI	
cod.prv	: ..
<u>cod.cli</u>	<u>nom.cli</u>
....
....
....

PROVINCIA	
cod.prv	: ..
nom.prv	:

provincia { cod.prv → nom.prv }

prvcli { cprv.cli →> { cod.cli, nom.cli } }

De los dos documentos podemos definir

$\text{prvcli2} \{ \wedge \text{prvcli} = \wedge \text{provincia} \}$ o lo que es lo mismo

$\text{prvcli2} \{ \text{cprv.cli} \rightarrow \{ \text{cod.cli}, \text{nom.cli} \} = \text{cod.prv} \rightarrow \text{nom.prv} \}$

que representan el documento

PRVCLI2	
cprv.cli	: ..
<hr/>	
cod.cli	nom.cli
....
....
....
nom.prv	:

Otro form distinto en forma pero no en contenido es

$\text{prvcli3} \{ \wedge \text{provincia} = \wedge \text{prvcli} \}$ que expandiéndolo queda

$\text{prvcli3} \{ \text{cod.prv} \rightarrow \text{nom.prv} = \text{cprv.cli} \rightarrow \{ \text{cod.cli}, \text{nom.cli} \} \}$

que representan el documento

PRVCLI3	
cod.prv	: ..
nom.prv	:
<hr/>	
cod.cli	nom.cli
....
....
....

otra manera de definir el mismo form es

$\text{prvcli4} \{ \{ \text{cprv.cli} = \wedge \text{provincia} \} \rightarrow \{ \text{cod.cli}, \text{nom.cli} \} \}$

que representa el documento

PRVCLI4	
cprv.cli	: ..
nom.prv	:
cod.cli	nom.cli
....
....
....

La estructura de esta fórmula es del tipo $\{ a = b \} \rightarrow c$, la interpretación es la siguiente. La subfórmula b está conectada a 'a' mediante el operador $=$, lo que quiere decir que es un form base. Si lo excluimos nos queda $\{ a \} \rightarrow c$, que, suprimiendo las llaves ahora innecesarias, produce $a \rightarrow c$. Por lo tanto, el form inicial es un form compuesto de dos forms base: el $\{ a \rightarrow c \}$ y el $\{ b \}$.

En el caso $\{ a = b \} \rightarrow \{ c = d \}$ tenemos tres forms base el $\{ a \rightarrow c \}$, el $\{ b \}$ y el $\{ c \}$.

Campos protegidos

Si se quiere impedir que un campo de un form pueda ser modificado desde la pantalla, se debe declarar como protegido poniendo a continuación del campo el carácter '.

Por ejemplo, si quisiéramos definir un form igual que el de cliente pero desde el cual los usuarios no pudiesen actualizar el nombre de los clientes declararíamos

```
cliente2 { cod.cli → { nom.cli', cprv.cli } }
```

en el cual, en vez de poner **nom.cli**, hemos escrito **nom.cli'**.

Cuando se ejecute este form, los campos protegidos se mostrarán en pantalla pero no se podrá teclear en ellos. En los gráficos de las pantallas los representaremos con el símbolo '.

CLIENTE	
cod.cli	:
nom.cli	: '.....'
cprv.cli	: ..

Si en el form *prvcli* protegemos todos los atributos que corresponden a la relación *prv* tendremos

```
prvcli { { cprv.cli = cod.prv' → nom.prv' }
        →> { cod.cli, nom.cli }
      }
```

que también se puede expresar como

```
prvcli { { cprv.cli = { cod.prv → nom.prv }' }
        →> { cod.cli, nom.cli }
      }
```

o

```
prvcli { { cprv.cli = ^prv' }
        →> { cod.cli, nom.cli }
      }
```

en estos tres forms sólo se actualizarán datos de la relación *cli* pero no de la *prv*.

La precedencia del símbolo ‘ queda entre ‘^’ y ‘→’, ‘→>’

{, }

^

’

→, →>

=

,

Cuando todos los atributos de una relación que aparecen en un form están protegidos, sólo se puede consultar datos de esta relación pero no actualizarlos. Lógicamente, en este caso, en el modelo deberá haber algún form que permita actualizar los datos de dicha relación.

Por ejemplo :

clientes

ESQUEMA [

```

prv(cod,nom)      : K( cod ); D( 2 A, 10 A)
cli(cod,nom,cprv) : K( cod ); D( 4 A, 12 A, ⊆cod.prv );

```

]

FORMS [

```

provin { cod.prv → nom.prv }
lprovin { →> ^provin }
cliente { cod.cli → { nom.cli, cprv.cli = ^provin' } }
lcliente { →> ^cliente }
prvcli { { cprv.cli = ^provin' }
        →> { cod.cli, nom.cli }
        }

```

]

Conexión de forms abierta.

Una alternativa al símbolo = para conectar forms es el símbolo =?.

Cuando ponemos el form prvcli

```

prvcli { { cprv.cli = ^provin' } →> { cod.cli, nom.cli } }

```

Estamos indicando que queremos que se verifique que el código de provincia que tecleemos en pantalla cprv.cli debe estar en la relación de provincias. En el caso de que esto no ocurra, no se podrá realizar actualizaciones de los datos. En otras palabras, desde este form solo podremos entrar clientes con provincias que ya hayan sido dadas de alta en la relación prv. O, dicho de otro modo, se exigirá el cumplimiento de la regla de integridad referencial que establece el modelo de datos relacional.

Si por alguna razón interesase evitar esta restricción, se utilizaría la conexión exterior =? en lugar de la conexión normal.

```

prvcli { { cprv.cli =? ^provin' } →> { cod.cli, nom.cli } }

```

Según establece el modelo de datos relacional, este form no es correcto pues no garantiza la integridad referencial de los datos. Sin embargo, hay casos en que el uso del conector abierto es necesario. Por

ejemplo si quisiéramos dar de alta simultáneamente las provincias y los clientes el form debería ser

```
prvcli2 { { cprv.cli = ^provin } →> { cod.cli, nom.cli } }
```

En este caso, el conector = no permite dar de alta una nueva provincia pues antes de dar de alta los datos intentaría consultar el valor de cprv.cli en el form provin, pero este valor todavía no existe por lo que no permitiría realizar la transacción. Por ello, el form correcto es

```
prvcli2 { { cprv.cli =? ^provin } →> { cod.cli, nom.cli } }
```

Obsérvese que aunque pongamos la conexión exterior la integridad referencial se conserva en este caso, pues no permite entrar un valor de cprv.cli que no exista en cod.prv, dado que los dos se dan de alta simultáneamente.

El símbolo ‘=?’ tiene la misma precedencia que el ‘=’.

Operadores aritméticos y sinónimos.

Un atributo de form puede ser una expresión aritmética de otros atributos que sean numéricos.

En una expresión aritmética se pueden utilizar las cuatro operaciones: +, -, * y / . Se pueden poner paréntesis para variar las precedencias de operación. Por ejemplo :

Algo

ESQUEMA [

```
    num(n1,n2) : D( 3 N0, 3 N0 ); K ( (n1,n2) )
```

]

FORMS [

```
    tasa { n1.numms
```

```
        →> { n2.numms, (n1.numms + n2.numms) * n1.numms }
```

```
    }
```

]

Cuando el ordenador genera la pantalla del form el rótulo que pone encima de la expresión aritmética es 'expresión', si queremos que ponga otro rótulo deberemos declarar un sinónimo de la expresión poniendo antes de la expresión el nombre que queramos seguido del símbolo ‘::’.

```
tasa { n1.numms
```

```
  ->> { n2.numms, indice :: (n1.numms + n2.numms) * n1.numms }
}
```

El atributo de form ‘índice’ se puede utilizar en el form para especificar otras expresiones, por ejemplo podríamos poner

```
tasa { n1.numms
```

```
  ->> { n2.numms,
        indice :: (n1.numms + n2.numms) * n1.numms,
        norma  :: indice / n2.numms }
}
```

o incluso podríamos haber hecho

```
tasa { n1.numms
```

```
  ->> { a :: n2.numms,
        indice :: (n1.numms + a) * n1.numms,
        norma  :: indice / a }
}
```

Este form generará el formato de pantalla

TASA		
n1.numms	:
<hr/>		
a	índice	norma
...
...
...

Obviamente, todas las expresiones aritméticas son campos protegidos. Por ello, en el documento se representan por ' en lugar de “.”

La precedencia del símbolo ‘::’ y de los símbolos aritméticos es

{, }

(,)

^

‘)’
 ‘:’
 ‘*’, ‘/’
 ‘+’, ‘-’
 ‘→’, ‘→>’
 ‘=’
 ‘,’

Los paréntesis, además de ser utilizados a la manera aritmética usual, cumplen también la misma función que las llaves. La diferencia es que los paréntesis convierten a los forms en ocultos (no visibles) en la pantalla del ordenador. Lo que da lugar a los atributos internos y a los forms internos.

Atributos internos

Si queremos que una expresión se calcule pero no aparezca en pantalla la declararemos como campo interno poniéndola entre paréntesis

Por ejemplo, si en el form anterior queremos que calcule el índice pero que no lo muestre por pantalla pondríamos

tasa { n1.num

```

    →> { a :: n2.num,
        ( indice :: (n1.num + a) * n1.num ),
        norma :: indice / a }
}
    
```

El formato de pantalla generado en este caso es

TASA	
n1.num :	
a	norma
...
...
...

Forms internos

Un form se declara interno poniéndolo entre paréntesis en lugar de entre llaves.

prod (cod.prd → nom.prd) en vez de **prod { cod.prd → nom.prd }**

La diferencia es que el segundo genera una pantalla de ordenador, desde la que podremos hacer actualizaciones, y el primero no. Los forms se declaran internos cuando se quiere incluirlos utilizando el símbolo '^' en otro form y no deseamos que genere documento.

Por ejemplo, si tenemos el siguiente esquema

prv(cod,nom,ext,hab) : K(cod); D(2 A, 10 A, 6 N0, 7 N0)
cli(cod,nom,cprv) : K(cod); D(4 A, 12 A, ⊂ cod.prv)

en donde ext significa extensión de la provincia y hab su número de habitantes; deberemos tener por lo menos dos forms para actualizarlas

prov { cod.prv → { nom.prv, ext.prv, hab.prv } }
cli { cod.cli → { nom.cli, cprv.cli = cod.prv } }

si quisiéramos que en el form cli apareciese además el nombre de la provincia deberíamos poner

cli { cod.cli → { nom.cli, cprv.cli = cod.prv → nom.prv' } }

una manera alternativa sería utilizando un form interno

prov { cod.prv → { nom.prv, ext.prv, hab.prv } }
provi (cod.prv → nom.prv)
cli { cod.cli → { nom.cli, cprv.cli = ^provi' } }

Operadores de agregación.

Si en un documento hay un campo numérico que se repite podemos declarar la suma de todos los valores, por ejemplo en

tasa { n1.numms →> n2.numms }

Campos sugeridos y ligaduras.

Supongamos el siguiente esquema de una empresa

prod(cod,nom,precio)

reci(num,cprod,cant)

que responde a la situación de que la empresa tiene una lista (relación) de productos y una relación de recibos que indican las ventas de los productos, num.reci es el número de recibo, cprod es el código del producto vendido y cant, la cantidad vendida de dicho producto.

Si suponemos que en un recibo puede haber varios productos tendremos el modelo

empresaA

ESQUEMA [

prod(cod,nom,precio) : D(4 A, 10 A, 6 N0); K(cod)

reci(num,cprod,cant) : D(4 N0, ⊆cod.prod, 3 N0);

K((num,cprod))

]

FORMS [

producto { cod.prod → { nom.prod, precio.prod } }

recibo { num.reci →> { cprod.reci = ^producto', cant.reci } }

]

Según este modelo, la empresa siempre venderá a precio de lista.

Si la empresa no vendiese a precio de lista, esto es, si negociase el precio en cada venta el modelo sería:

empresaB

ESQUEMA [

prod(cod,nom): D(4 A, 10 A); K(cod)

reci(num,cprod,cant,precio): D(4 N0, ⊆cod.prod, 3 N0, 6 N0);

K((num,cprod))

]

FORMS [

```

producto { cod.prod → nom.prod }
recibo { num.reci →> { cprod.reci = ^producto', precio.reci, cant.reci }
}
]

```

Como vemos, ahora el precio ya no depende del producto sino del producto y el recibo conjuntamente, es el precio a que se vende un determinado producto en una transacción comercial concreta, en un recibo concreto.

Pero un hecho no infrecuente en la práctica de las empresas es que aunque tienen una lista de precios establecida y casi siempre venden al precio de la lista, puede haber algunos casos en que sea necesario no seguir esta norma. En esta situación tenemos que, por una parte el precio debe cumplir con el esquema que hay en el modelo de empresaA, pero por otra parte, para poder reflejar el hecho de que no se ha vendido al precio prefijado, el modelo válido es el de la empresaB.

En este caso el esquema tendría que ser

```

prod(cod,nom,precio)      :   D( 4 A, 10 A, 6 N0 ); K( cod )
reci(num,cprod,cant,precio) :   D( 4 N0, c cod.prod, 3 N0, 6 N0 );
                           K( (num,cprod) )

```

siendo precio.prod el precio de venta sugerido y precio.reci el precio de venta a que se ha vendido el producto

Si quisiéramos definir un form en el que el ordenador nos sugiriese el precio del producto permitiéndonos modificarlo, pondríamos en lugar de precio.reci

```
precio.reci ← precio.prod
```

Al especificar esto, el ordenador, cuando el campo precio está vacío, lo rellena con el precio de lista, pero el usuario puede modificarlo a su voluntad.

Si quisiéramos que desde un form el precio propuesto no fuera modificable (aunque si desde otros posibles forms) deberíamos poner

```
precio.reci' ← precio.prod
```

Estableciéndose así una ligadura entre dos atributos

Podríamos definir un modelo que tuviera distintos forms, uno en el que no se pudiese modificar el precio, otro en el que se entrase libremente y otro en el que se sugiriese.

empresa

ESQUEMA [

```
prod(cod,nom,precio)      :      D( 4 A, 10 A, 6 N0 ); K( cod )
reci(num,cprod,cant,precio) :      D( 4 N0, 3 cod.prod, 3 N0, 6 N0 );
                                K( (num,cprod) )
```

]

FORMS [

```
producto { cod.prod → nom.prod }
```

```
recibo1 { num.reci
```

```
    →> { cprod.reci = cod.prod → { nom.prd',(precio.prod) },
        precio.reci' ← precio.prod , cant.reci
        }
```

```
    }
```

```
recibo2 { num.reci
```

```
    →> { cprod.reci = ^producto',
        precio.reci,
        cant.reci
        }
```

```
    }
```

```
recibo3 { num.reci
```

```
    →> { cprod.reci = ^producto',
        precio.reci ← precio.prod,
        cant.reci
        }
```

```
    }
```

]

Desde el form recibo1 únicamente se podría vender a precio de lista.

Desde recibo2 se podría vender a cualquier precio.

Desde recibo3 también, pero el ordenador, cuando ve el campo precio vacío, pone el precio de lista permitiendo que sea modificado.

Los símbolos de sugerencia '←' y '<<←' tienen la siguiente precedencia:

{, }

(,)

^

”

::

Σ

*, /

+, -

← y <<←

→, →>

=

,

Especialización de forms

Si queremos que un usuario que utiliza un form sólo pueda 'ver' y actualizar una parte de los datos podemos declarar el form con una restricción

Por ejemplo si queremos que un usuario únicamente pueda consultar y actualizar los clientes de Barcelona escribiremos

client { cod.cli → { nom.cli, cprv.cli } [cprv.cli = 'B'] }

Las restricciones se ponen entre corchetes al final del form.

Una restricción se puede hacer por igual '=', menor '<', menor o igual '<=', mayor '>', mayor o igual '>=' o por distinto '<>'.

Los operandos de la restricción pueden ser expresiones aritméticas.

Se pueden unir restricciones mediante los operadores lógicos O (representado por el símbolo '|'), Y (por el '&'), y NO (!).

Ejemplo : Especializar el form anterior para los clientes de Barcelona o Madrid.

client { cod.cli → { nom.cli, cprv.cli } [cprv.cli = 'B' | cprv.cli = 'M'] }

Se puede hacer uso de paréntesis para variar el orden de cálculo.

Las precedencias de estos operadores de menor precedencia a mayor, |, & y !.

{, }

(,)

^

»

::

!

Σ

*, /

+, -

← y <<

→, →>

=, <, <=, >, >=, <>

&

|

=

,

Definición formal del lenguaje en notación BNF

Para presentar la gramática de la notación de manera estrictamente formal utilizaremos una técnica empleada en la construcción de lenguajes de programación. Un lenguaje de programación se establece definiendo una sintaxis y su 'semántica' (las acciones que ejecuta el ordenador al leer una expresión).

Una notación muy utilizada para establecer la sintaxis de un lenguaje es la BNF (Backus-Naur Form), que permite establecer la gramática de una amplia familia de lenguajes, los denominados lenguajes de gramática libre de contexto (context-free grammar) (Aho et al., 1986)p.25

Para construir el analizador sintáctico, se empleó el programa Yacc (Levine et al., 1995; Kernighan and Pike, 1987). Yacc es un generador de analizadores sintácticos de las gramáticas BNF del tipo LALR (LookAhead Left to Right) (Aho et al., 1986).

Este programa lee la notación BNF de la especificación del lenguaje que el diseñador establece y crea un programa que analiza sintácticamente las expresiones válidas del lenguaje establecido. Esto es, valida las fórmulas bien formadas.

Yacc va acompañado de otro programa, Lex, que es un analizador léxico. La utilidad de este último es que permite definir símbolos formados por más de un carácter.

Definición del vocabulario

"}", FINP,
"→", FUNS,
"→>", FUNM,
"←", AFUNS,
"←←", AFUNM,
"=", EQI,
"=?", EQ,
">", GT,
"<", LT,
">=", GE,

"<=", LE,
 "<>", NE,
 "+", SUM,
 "-", RES,
 "*", MUL,
 "/", DIV,
 "Σ", TOT,
 "!", NOT,
 "&", AND,
 "|", OR,
 "::", SINO,

Definición de la sintaxis

%token <nom> NUMBER
 %token <nom> ESQUEMA FORMS DOMS KEYS
 %token <nom> NOMATR NOMATRC
 %token <tok> EQ GT GE LT LE NE NOT AND OR
 %token <kte> NUM STRING FECHA
 %token <cam> NOM ATREL ATRPA ATRV
 %token FINP

%type <nom> lonatr foatr nomsin
 %type <tok> oprel
 %type <cam> nomvw atrv attr atrp
 %type <vw> cte dom lcte dcte
 %type <vw> dcl lupd restr vw lvw dw
 %type <pref> campo expr
 %type <lvw> evisual ldclvw defvw dclvw

%left OR
%left AND
%left EQ EQI GT LT GE LE NE
%left '['
%left FUNS FUNM
%left AFUNS AFUNM
%left SUM RES
%left MUL DIV
%left TOT
%left NOT
%left SINO

```

modelo : nommod
    ESQUEMA
    '[' dclesq ']'
    FORMS
    '[' evisual ']'
    ;
dclesq:      dclrel
    | dclesq dclrel
    ;
dclrel:  nomrel
    '(' listatr ')'
    ':' listspec
    ;
listatr: nomatr
    | listatr ',' nomatr
    ;
listspec: specrel
    | listspec ';' specrel
    ;
specrel: DOMS '(' listdom ')'
    | KEYS '(' listkey ')'
    ;
listdom: specdom
    | listdom ',' specdom
    ;
specdom: lonatr foatr
    | '⊂' nomatrc
    | '=' nomatrc
    ;
listkey: speckey

```

```
    | listkey ',' speckey
    ;
speckey: nomref
    | '(' listref ')'
    ;
listref: nomref
    | listref ',' nomref
    ;
nommod:  NOMATR
    ;
nomrel:  NOMATR
    ;
nomatr:  NOMATR
    ;
nomatrc: NOMATRC
    ;
nomref:  NOMATR
    ;
nomsin:  NOMATR
    ;
foatr:   NOMATR
    ;
lonatr:  NUMBER
    ;
evisual: ldclvw
    ;
ldclvw: defvw
    | ldclvw defvw
    ;
defvw:  nomvw '{' dwv '}'
```

```

    | nomvw '{' dvw FINP
    | nomvw '(' dvw ')'
;
dvw:   vw
;
vw:    dcl
    | dom
    | '{' lvw '}'
    | '{' lvw FINP
    | vw FUNS void
    | vw FUNM void
    | vw FUNS vw
    | void FUNM vw
    | vw FUNM vw
    | vw EQ vw
    | vw EQI vw
    | vw AFUNS vw
    | vw AFUNM vw
    | vw '[' restr ']'
    | '(' vw ')'
    | '^' nomvw
;
lvw:   vw
    | lvw ',' vw
    | '(' lvw ')'
;
dcl:   expr
    | atrv SINO dcl
;
expr:  campo

```

```
| expr SUM expr
| expr RES expr
| expr MUL expr
| expr DIV expr
| TOT '(' campo ')'
| '(' expr ')'
;
campo: cte
      | atrv
      | attr
      | atrp
;
restr:  expr oprel expr
      | NOT restr
      | restr AND restr
      | restr OR restr
      | '[' restr ']'
;
dom:   '!' lcte '!'
;
lcte:  dcte
      | lcte ',' dcte
;
dcte:  cte
      | cte TITOS cte
;
nomvw: NOM
;
cte:   NUM
      | STRING
```

```

    | FECHA
    ;
void:
    ;
attr:  ATREL
    ;
atrp:  ATRPA
    ;
atrv:  NOM
    ;
oprel: EQ
    | GT
    | GE
    | LT
    | LE
    | NE
    ;

```

Al construir el compilador asociado a una gramática se pueden incluir tantos procedimientos (algoritmos) como se desee y aumentar así las reglas de conformación de los forms válidos. La ventaja de esta manera de hacer es que da una gran libertad para verificar todas las restricciones que se requieran para caracterizar los forms que se consideren correctos. La desventaja es que toda esta serie de reglas quedan ‘ocultas’ dentro de los algoritmos. La estructura de los cálculos formales (vocabulario, gramática, axiomas y reglas de inferencia) permiten explicitar todas las restricciones que caracterizan los forms correctos.

En los próximos apartados se presenta la notación con estructura de cálculo, pero antes, cabe aclarar lo que entendemos por form válido o correcto. Para ello, primero definiremos la noción de validez tal como se entiende en lógica (cálculo) de proposiciones (Schöning, 1989) y a continuación lo aplicaremos a nuestro cálculo.

Una fórmula F del cálculo proposicional es ‘satisfacible’ si tiene al menos un modelo. Esto es, si existe al menos una proposición que la hace cierta. Sea p cualquier proposición (por ejemplo, ‘la Luna es de queso’), entonces p es una fórmula satisfacible pues hay por lo menos una sustitución que la hace verdadera. En este caso, p es una fórmula satisfacible pues la proposición ‘la Luna es un satélite de la Tierra’ la hace cierta.

Hay fórmulas ‘insatisfacibles’, por ejemplo $p \wedge \neg p$. (Siendo \wedge el ‘y’ lógico y \neg la negación). No importa la proposición por la que sustituyamos p , la fórmula nunca puede ser verdadera. (Siempre que aceptemos el principio del tercio excluso, esto es, que no puede haber algo que sea y no sea a la vez).

Las fórmulas ‘válidas’ son aquellas que son siempre verdaderas. Por ejemplo, $p \vee \neg p$. Cualquier proposición es cierta o no es cierta. A las fórmulas válidas también se les llama ‘tautologías’ o ‘teoremas’ del cálculo.

En un cálculo formal, los axiomas son fórmulas válidas por definición y los teoremas son las fórmulas que se pueden obtener mediante la utilización de una serie de reglas de transformación aplicadas a los axiomas, que se llaman reglas de inferencia.

Cabe aclarar que en lógica formal la noción de verdad, o ‘hacer verdadera’ es puramente formal, no hace ninguna referencia a las interpretaciones del mundo exterior a ella misma, a la verdad empírica, como correspondencia de la proposición con el mundo. Pero entrar en el mundo de la verdad empírica supone abandonar el mundo seguro de la lógica matemática para adentrarnos en el pantanoso de la filosofía. Una medida de la dificultad filosófica la ofrecen los libros de Blackburn y Simmons (1999) y Richard (2003). Los dos libros son una recopilación de artículos de diferentes filósofos debatiendo la noción de ‘verdad’, en el primero, y la noción de ‘significado’, en el segundo. Para no alargar innecesariamente la explicación los hemos mezclado (de manera poco rigurosa) tal como se hace en algunos manuales de lógica aplicada.

Para construir el cálculo distinguiremos cuatro nociones de ‘verdad’: la verdad deductiva la verdad subjetiva y la verdad empírica y la verdad intersubjetiva. Esta clasificación es meramente práctica. Aunque los términos elegidos puedan evocar significados filosóficos de la noción de

verdad, no se pretende aquí entrar en este difícil debate que recorre toda la historia de la filosofía.

El propósito del cálculo es que mediante la aplicación mecánica de las reglas de inferencia se obtengan teoremas que al ser interpretados como formularios de empresa produzcan unos significados en el usuario que estén de acuerdo con su verdad empírica, esto es, que reflejen adecuadamente su percepción de la realidad y, además, que la máquina se comporte conforme a sus expectativas (verdad subjetiva). Pero cada usuario también debe validar, comprobar, si los formularios producidos responden a su verdad empírica, por ejemplo si el término 'dirección del cliente' que aparece en un documento producido denota la realidad tal como él la percibe en dicho formulario. Además, dado que los esquemas conceptuales pretenden expresar un vocabulario compartido por todos los usuarios de la organización, el significado de cada término concreto debe ser el mismo para cada usuario que lo utilice en todos los formularios en donde aparezca, debe denotar lo mismo (verdad intersubjetiva).

En otras palabras, el problema es conciliar la verdad matemática (según la acepción del término que tienen David Hilbert, Bertrand Russell y Albert Einstein, aunque no compartida por todos los matemáticos) con la verdad social en sus tres vertientes, la subjetiva, la empírica y la intersubjetiva. (Respecto a esta última división todavía hay menos acuerdo entre los filósofos.)

Vocabulario del cálculo.

El vocabulario está formado por dos tipos de símbolos, los fijos, que son los propios de la notación que proponemos; y los particulares, que son los términos utilizados para modelar el sistema de información de una empresa concreta. Estos últimos se introducen junto con los axiomas.

Consiste en 15 tipos de símbolos fijos

1. Constantes :

Numéricas. Ej. 23,4

Alfanuméricas. Ej. "hola"

Fechas. Ej. 23/11/48:13:21:15

2. Atributos de pantalla : se representan por una palabra acabada en un punto Ej. factor.
3. Atributos de form (o sinónimos) : se representan por una palabra Ej. total
4. Atributos de relación : se representan por una palabra un punto y otra palabra Ej. codigo.cliente
5. Protección '
6. Establecimiento de sinónimo ::
7. Agregación { }
8. Agregación interna ()
9. Concatenación ,
10. Restricción []. Dentro de una restricción pueden aparecer los símbolos de comparación =, >, >=, <, <=, <> y los conectores booleanos negación, & y lógico, | o lógico.
11. Operadores aritméticos: +, -, *, /, Σ , .count, .max, .min, .avg
12. Operadores funcionales (se representan genéricamente con el metasímbolo Φ) :
 - funcional
 - multifuncional
13. Operador conexión (genéricamente, el metasímbolo Θ) :
 - = conexión de forms
 - =? Conexión de forms abierta
14. Operador Sugerencia (genéricamente, el metasímbolo σ)
 - ← Sugerencia simple
 - ←← Multisugerencia
15. Símbolos de entorno
Por ejemplo, **.us** Usuario, **.hoy** Fecha del día, **.null** Campo vacío

Los cuatro primeros tipos de símbolos son particulares de cada modelo. El resto son comunes, propios de la notación.

Gramática del cálculo

Definición de expresión.

Un atributo (de pantalla , de form o de relación) es una **expresión**.

Si e y x son expresiones de dominio numérico, también es una expresión $(e + x)$, $(e - x)$, $(e * x)$, (e / x) y los operadores de agregación $\Sigma(e)$, $.C(e)$, $.AVG(e)$, $.MAX(e)$, $.MIN(e)$

Definición de restricción.

Si e y x son expresiones entonces $[e = x]$, $[e < x]$, $[e \leq x]$, $[e > x]$, $[e \geq x]$, $[e <> x]$ es una **restricción**.

Si r y s son restricciones también son restricciones $[!r]$, $[r \& s]$, $[r | s]$.

Definición de form.

Una expresión es un form (fbf), el vacío es un form,

si v y w son forms entonces también son forms :

$\{ v' \}$	protegido
$\{ (v) \}$	interno
$\{ v[] \}$	restringido
$\{ atrv :: v \}$	sinónimo
$\{ v , w \}$	concatenación
$\{ v \Phi w \}$	dependencia
$\{ v \Theta w \}$	conexión
$\{ v \sigma w \}$	sugerencia

Precedencias de asociación de símbolos

Para evitar el excesivo uso de paréntesis y llaves se establecen las siguientes precedencias de menor precedencia a mayor

, \ominus Φ σ | $\&$ \neg + - * / ::

La precedencia entre dos símbolos que tengan la misma se resuelve atribuyendo mayor precedencia al que está más a la izquierda.

Axiomas del cálculo

El problema que se presenta en la notación que propuesta en el capítulo anterior es que los axiomas, el esquema conceptual de la base de datos del sistema de información, no son fórmulas bien formadas según la gramática empleada para definir los forms. Esto se puede obviar si consideramos que el esquema expresado en notación relacional se puede traducir a la notación de forms.

Convención: las letras minúsculas representarán un único atributo, las letras mayúsculas, conjuntos de atributos.

El esquema conceptual de un sistema de información, esto es, el conjunto de relaciones normalizadas, forman el conjunto de axiomas de dicha empresa.

Sea $r_i(a_{i1}, a_{i2}, a_{i3}, \dots)$ un conjunto de relaciones normalizadas

sea $\mathcal{K}(K_{i1}, K_{i2}, \dots)$ las claves de la relación r_i ,

donde $K_{ij} \in \mathcal{P}\{a_{i1}, a_{i2}, a_{i3}, \dots\}$ (siendo \mathcal{P} el conjunto potencia)

en donde K_{i1} es la clave primaria de la relación r_i , y K_{ij} para $j > 1$, las claves alternativas de la relación r_i

y sea $\mathcal{D}(\delta_{i1}, \delta_{i2}, \delta_{i3}, \dots)$ el conjunto de dominios en los que δ_{i2} , puede ser un dominio primitivo (numérico, etc.), o bien una clave externa $S_{ij} \subset K_{i1}$ o $S_{ij} = K_{i1}$ para $S_{ij} \in \mathcal{P}\{a_{i1}, a_{i2}, a_{i3}, \dots\}$

entonces el esquema:

$r_i(a_{i1}, a_{i2}, a_{i3}, \dots) : \mathcal{K}(K_{i1}, K_{i2}, \dots); \mathcal{D}(\delta_{i1}, \delta_{i2}, \delta_{i3}, \dots)$

es equivalente al conjunto de fórmulas:

$$K_{i1} \rightarrow \{B_{i1}\}$$

$$\text{y } (K_{im} \rightarrow \{B_{im}\}) \text{ para } m > 1$$

donde $B_{ij} = A_i - K_{ij}$ (diferencia de conjuntos), siendo A_i el conjunto de atributos de la relación r_i

La especificación de claves externas se puede expresar mediante un conjunto de restricciones

$S_{ij} \subset K_{i1}$ o $S_{ij} = K_{i1}$ para indicar las claves externas

siendo $S_{ij} \in \mathcal{P} \{a_{i1}, a_{i2}, a_{i3}, \dots\}$

Para el resto de atributos se expresa simplemente su dominio, $a_{ij} \subset \delta_{ij}$

Reglas de inferencia del cálculo

El propósito de las reglas de inferencia es permitir obtener deductivamente los forms mediante los cuales los usuarios no puedan entrar valores que violen las restricciones a los datos especificadas en los axiomas y que, además, las acciones de actualización sean percibidas por los usuarios como las naturales.

El problema de la actualizabilidad formal de vistas (view updatability) ha sido muy estudiado en la literatura, por ejemplo (Bancilhon and Spyrtos, 1981). (Keller and Ullman, 1984 ; Chen and McLeod, 1989 ; Barsalou et al., 1991; Date, 1995; Lechtenbörger, 2003) En la revisión efectuada, más amplia que la del ejemplo, no se ha encontrado una solución definitiva. Las reglas que aquí se proponen tampoco pretenden ser completas.

La dificultad radica, a nuestro juicio, en que además de la parte formal del problema está el componente convencional. Establecer cómo se ha de comportar la máquina cuando el usuario realiza una acción de actualización de datos tiene un doble componente convencional (en dos de las acepciones del término convención: como regla impuesta y como uso social). La primera es el establecimiento de una serie de reglas, más o menos arbitrarias, de comportamiento de la máquina. La segunda, que estas reglas convencionales sean aceptadas (entendidas) por los usuarios, esto es, que sean coherentes con las convenciones sociales, con los usos existentes.

En algún caso se ha comprobado que algún comportamiento de la máquina no se correspondía con el esperado por los usuarios. Dado el estado embrionario del cálculo que proponemos, la validación social de las reglas de inferencia propuestas queda fuera del alcance de este trabajo.

Reglas Φ

$$\Phi 1 - \{ A, B \} \rightarrow C \Rightarrow \{ B, A \} \rightarrow C$$

$$\Phi 2 - A \rightarrow \{ X, Y \} \Rightarrow A \rightarrow \{ Y, X \}$$

$$\Phi 3 - A \rightarrow X \text{ y no } (X \rightarrow A) \Rightarrow X \rightarrow\rightarrow A$$

$$\Phi 4 - \{ A, B \} \rightarrow C \Rightarrow A \rightarrow\rightarrow \{ B, C \}$$

$$\Phi 5 - A \rightarrow \{ B, C \} \text{ y no } \{ B \rightarrow A \} \Rightarrow B \rightarrow\rightarrow \{ A, C \}$$

$$\Phi 6 - A \rightarrow \{ X, Y \} \Rightarrow A \rightarrow X$$

Dado que los axiomas son forms íntegros, y que estas reglas no permiten perder atributos de la clave primaria, aplicándolas sólo se pueden deducir forms íntegros.

Reglas Θ **Definición de Θ -compatibilidad de forms**

Dos forms v y w tienen compatibilidad de conexión si sus identificadores tienen el mismo dominio. $D(\text{Id}(v)) \equiv D(\text{Id}(w))$

$\Theta 1$ - La conexión $v \Rightarrow w$ de dos teoremas v y w , es otro teorema.

$\Theta 2$ - Si f es una subfórmula bien formada de un teorema v , y w es un teorema, y f y w son Θ -compatibles, entonces al sustituir f por $f \Theta w$ se obtiene otro un teorema.

Reglas σ

Definición de σ -compatibilidad de forms: Dos forms tienen compatibilidad de sugerencia si son isomorfos y los atributos que se corresponden tienen el mismo dominio.

$\sigma 1$ - Si f es una subfórmula bien formada de un teorema v , y w es un teorema, y f y w son σ -compatibles, entonces al sustituir f por $f \sigma w$ se obtiene otro un teorema.

Reglas ρ

$\rho 1$ - Si v es un teorema entonces $v[R]$ es un teorema. Siendo R una restricción.

Reglas Ω

$\Omega 1$ – Si f es una subfórmula bien formada de un teorema v , al sustituir f por f' o por (f) , se obtiene otro teorema

Capacidad expresiva

Para mostrar la capacidad expresiva del lenguaje propuesto presentamos un ejemplo en el que se analiza y especifica una pequeña aplicación, un sistema de gestión de pedidos de clientes de una empresa.

Para la parte de análisis, para obtener el esquema conceptual, emplearemos la notación Idef1x (IEEE ,1999). Esta notación, descendiente de la Entity-Relationship (Chen, 1976), expresa de forma gráfica el esquema conceptual. Las notaciones gráficas, en general son las que se llaman de modelado conceptual, tienen la ventaja de ser más intuitivas y por esta razón permiten obtener el esquema conceptual de manera heurística. El inconveniente que tienen es que no son formales como lo es el modelo de datos relacional. En este ejemplo presentaremos el esquema conceptual gráficamente teniendo presente que es posible traducirlo a la forma textual del modelo relacional. Han sido propuestos distintos procedimientos para traducir estas notaciones gráficas a la notación del modelo relacional (Bagui and Earp, 2003; Andersson, 1994 ; Markowitz and Shoshani, 1992; Ridjanovic et al., 1987 ; Teorey et al., 1986).

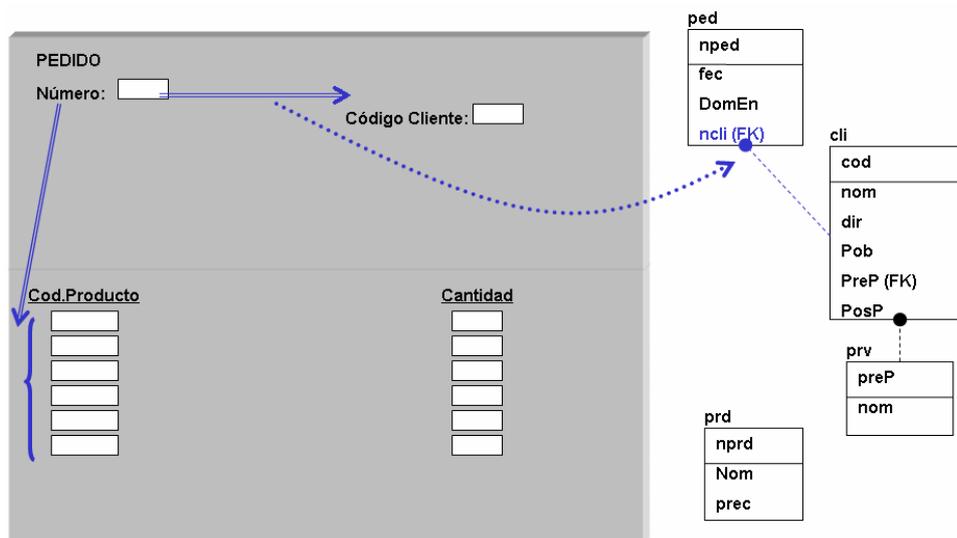
Supongamos un pedido de una empresa como el que se presenta a continuación:

PEDIDO				
Número:	<input type="text"/>	Código Cliente:	<input type="text"/>	
Fecha:	<input type="text"/>	Nombre :	-----	
Domicilio Entrega		Dirección :	-----	
<input type="text"/>		Población :	-----	
		Cod.Postal: -----		
		Nombre Provincia: -----		
Cod.Producto	Nombre	Precio	Cantidad	Importe
<input type="text"/>	-----	-----	<input type="text"/>	-----
<input type="text"/>	-----	-----	<input type="text"/>	-----
<input type="text"/>	-----	-----	<input type="text"/>	-----
<input type="text"/>	-----	-----	<input type="text"/>	-----
<input type="text"/>	-----	-----	<input type="text"/>	-----
<input type="text"/>	-----	-----	<input type="text"/>	-----
			Total:	-----

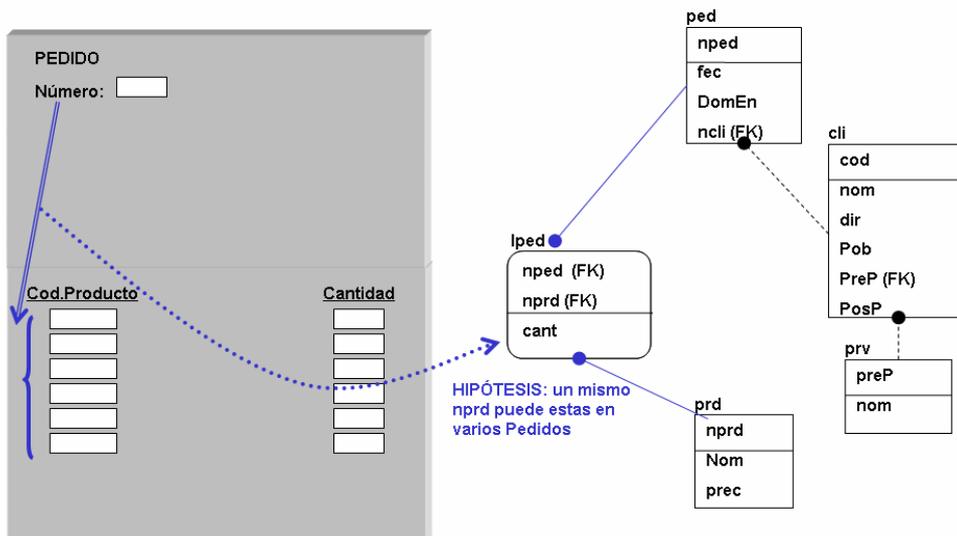
Han sido propuestos muy pocos métodos heurísticos, que permiten inducir (sería más adecuado emplear el término ‘abducir’ del filósofo Charles Sanders Peirce, cuyo significado se aproxima al de conjeturar) el esquema conceptual a partir de los formularios (Choobineh et al., 1992; Tseng and Mannino ,1988).

El método consiste en ‘descubrir’ de qué clases de entidades habla el documento. En el siguiente gráfico están señalados. Los círculos agrupan los atributos que describen características de dichas entidades (utilizaremos, por brevedad, el término ‘entidad’ en el sentido de clase de entidades, de objetos concretos). En cada entidad distinguiremos el, o los, atributos que actúan de identificador de cada objeto concreto. Por ejemplo, en el gráfico siguiente hemos distinguido la entidad ‘cliente’, identificada por su código, y dentro de él hemos señalado otra entidad, provincia, identificada por ‘cod.Postal’. También hemos distinguido la entidad pedido, y la entidad producto. Los campos importe y total se suprimen pues, al ser el resultado de operaciones aritméticas, no es necesario guardarlos en la base de datos.

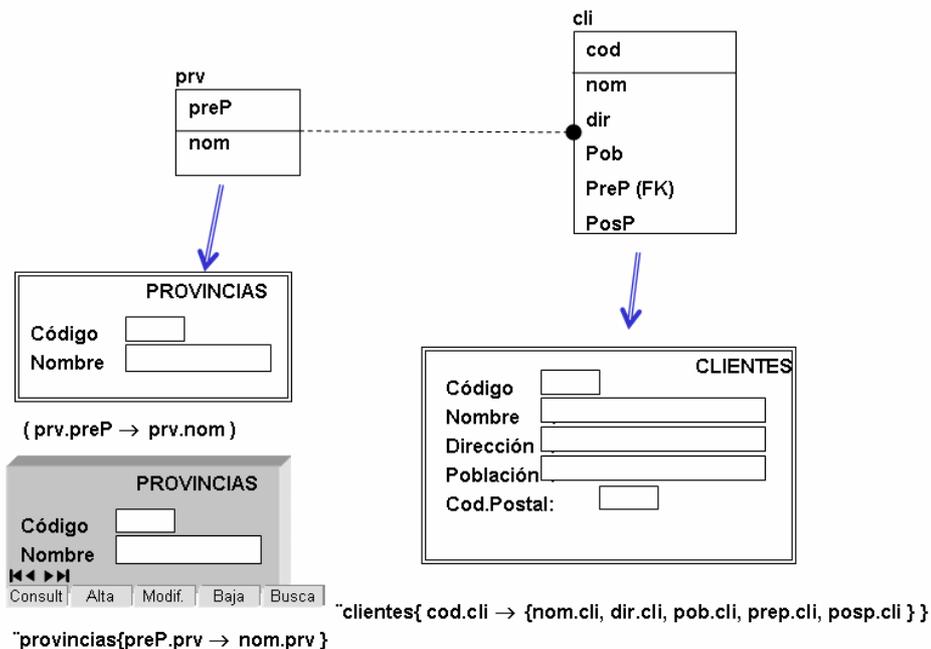
Como se puede ver en el gráfico siguiente, en el documento todavía quedan atributos que proporcionan información. Cada pedido lo realiza un cliente. Este atributo se incorpora a la entidad pedido (ped). La FK que se pone a la derecha del atributo indica que es clave externa (Foreign Key). Lo cual significa que no se puede hacer un pedido a un cliente que haya sido previamente dado de alta como cliente. Lógicamente, esta norma debe ser elicitada del usuario de esta aplicación.



Por último, nos quedan los atributos del diagrama siguiente, que parecen indicar que a cada número de pedido le corresponden varios códigos y cantidades de producto. Esto se traduce en la entidad lped (por líneas de pedido). En el gráfico se ha indicado una HIPÓTESIS pues, aunque el documento expresa claramente que un pedido puede tener varios productos, no dice nada de si un mismo código de producto puede aparecer en distintos pedido. Esta hipótesis debe ser resuelta por el empleado de la empresa que lleva la gestión de los pedidos.



Con esto se ha obtenido el esquema conceptual del sistema. Ahora lo que se ha de verificar es si a partir de este esquema es posible generar los forms que especifican las pantallas de ordenador que el usuario validará. De las dos entidades prv y cli se pueden obtener los forms completos provincias y clientes.



Pero el form clientes no tiene toda la información adecuada, es conveniente superponer el form provincias para ofrecer una información más completa de la ficha de clientes

The diagram shows a window titled 'CLIENTES'. It contains several input fields: 'Código', 'Nombre', 'Dirección', 'Población', and 'Cod. Postal'. A sub-form is overlaid on the 'Cod. Postal' field, containing two fields: 'Código' and 'Nombre'.

```

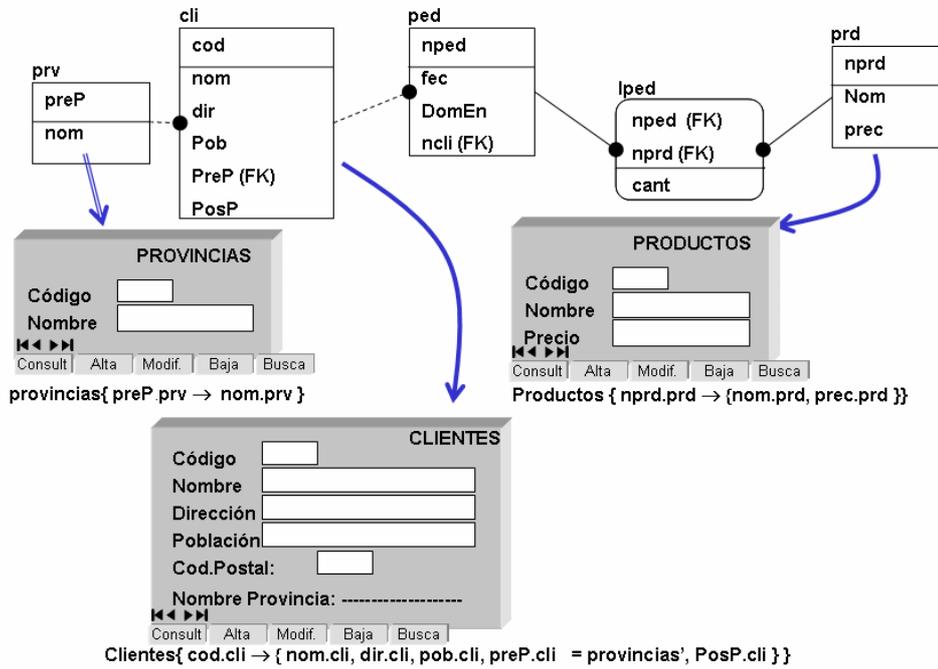
clientes{
cod.cli → {
nom.cli
, dir.cli
, pob.cli
, prep.cli = {preP.prv → nom.prv }
, posp.cli
}}
    
```

The diagram shows a window titled 'CLIENTES'. It contains input fields for 'Código', 'Nombre', 'Dirección', 'Población', and 'Cod. Postal:'. Below these is a field for 'Nombre Provincia: -----'. At the bottom, there are navigation buttons: 'Consult', 'Alta', 'Modif.', 'Baja', and 'Busca'.

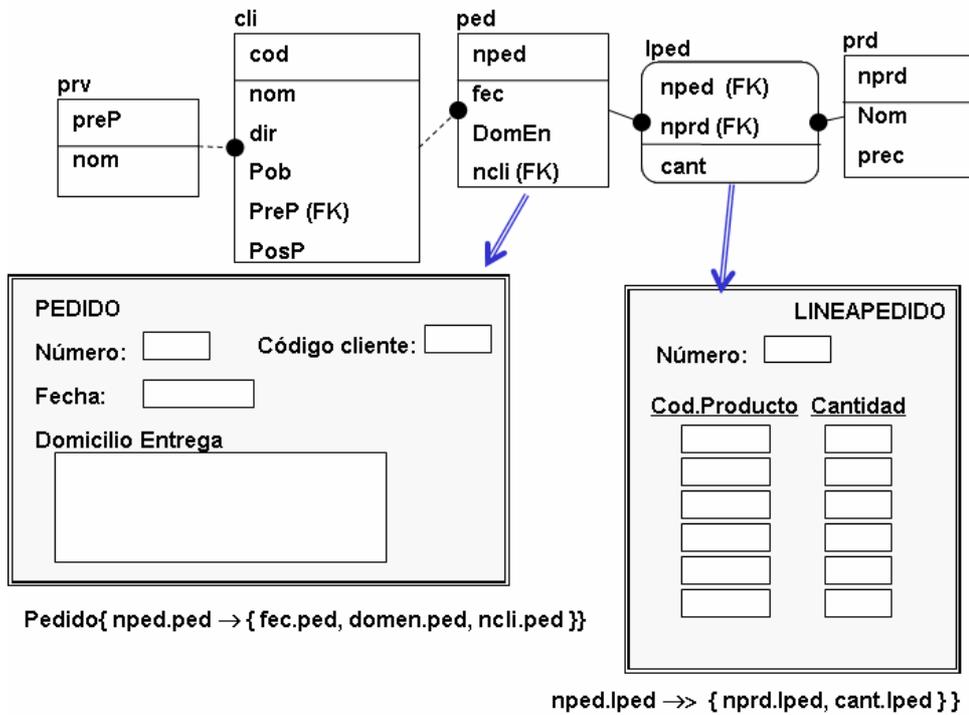
```

"clientes{ cod.cli → {nom.cli, dir.cli, pob.cli, prep.cli = ^provincias',posp.cli } }
    
```

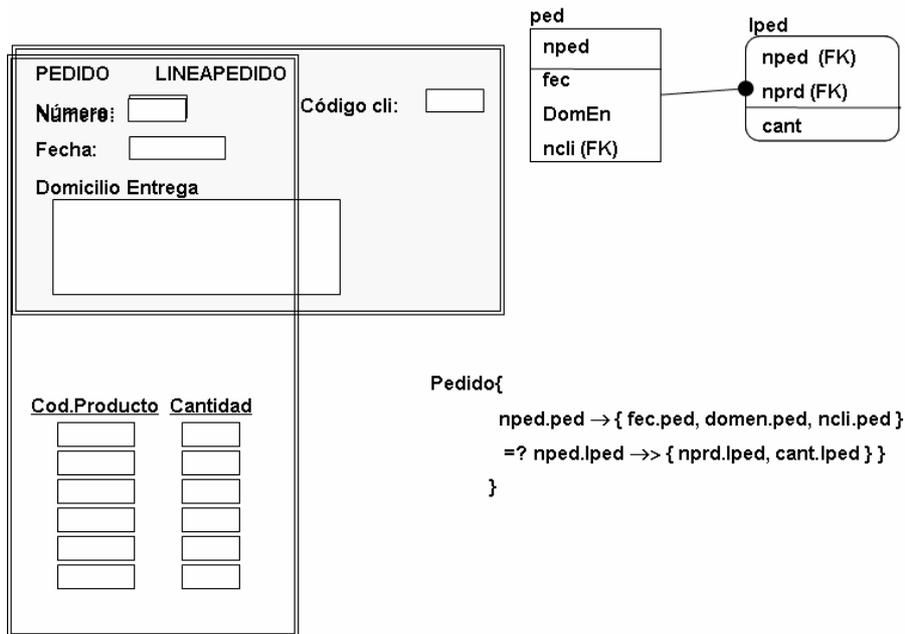
Se procede de manera análoga con el form productos y obtenemos las tres pantallas básicas, que el usuario podrá validar, para alimentar los datos necesarios para realizar los pedidos, pantalla que se especifica a continuación.



Como se indica en el gráfico siguiente, de la relación ped se deduce el fom pedido, y de lped, el de línea de pedido



Pero un pedido es la conexión de los dos forms anteriores. Con esto obtenemos el ‘esqueleto’ de un pedido, la información que lo define, dado el esquema conceptual del que partimos.



Para conseguir el pedido de manera más legible por el usuario, simplemente debemos visualizar la información del cliente y de los productos conectándolo con los forms correspondientes, así como indicar las operaciones aritméticas que sean adecuadas. Esto se muestra en la página siguiente.

PEDIDO

Número:

Fecha:

Domicilio Entrega

Código Cliente:

Nombre :-----

Dirección :-----

Población :-----

Cod.Postal: -----

Nombre Provincia: -----

<u>Cod. Producto</u>	Nombre	Precio	Cantidad
<input type="text"/>	-----	-----	<input type="text"/>
<input type="text"/>	-----	-----	<input type="text"/>
<input type="text"/>	-----	-----	<input type="text"/>
<input type="text"/>	-----	-----	<input type="text"/>
<input type="text"/>	-----	-----	<input type="text"/>

```

Pedido{
  nped.ped → {
    fec.ped,
    domen.ped,
    ncli.ped = ^clientes'
  }

  =? nped.lped → {
    nprd.lped = ^productos'
    , cant.lped } }
  }
    
```

Que especifica el form definitivo:

PEDIDO

Número:

Fecha:

Domicilio Entrega

Código Cliente:

Nombre :-----

Dirección :-----

Población :-----

Cod.Postal: -----

Nombre Provincia: -----

Cod.Producto	Nombre	Precio	Cantidad	Importe
<input type="text"/>	-----	-----	<input type="text"/>	-----
<input type="text"/>	-----	-----	<input type="text"/>	-----
<input type="text"/>	-----	-----	<input type="text"/>	-----
<input type="text"/>	-----	-----	<input type="text"/>	-----
<input type="text"/>	-----	-----	<input type="text"/>	-----
<input type="text"/>	-----	-----	<input type="text"/>	-----
Total:				-----

```

pedido{ { nped.ped → {fec.ped,DomEn.ped, ncli.ped = CLIENTES' }
        =? { nped.lped →> { nprd.lped =productos', cant.lped, importe :: prec*cant } }
        -> total :: sum(importe)
}

```

Así, la especificación completa del sistema a construir queda:

ESQUEMA [

prv(preP, nom) :K(preP)

cli(cod, nom, dir, pob, preP, podP)

:K(cod); D(ent, text, text, text, ⊂ preP.prv, ent)

prd(nprd, nom, prec) :K(nprd); D(ent, text, real)

ped(nped, fec, domEn, ncli) :K(nped); D(ent, date, text, ⊂ cod.cli)

lped(nped, nprd, cant)

:K((nped, nprd)); D(⊂ nped.ped, ⊂ nprd.prd, ent)

]

FORMS [

provincias{ preP.prv → nom.prv }

```

clientes{ cod.cli →
      { nom.cli, dir.cli, pob.cli, preP.cli = provincias', PosP.cli }
    }
productos { nprd.prd → {nom.prd, prec.prd }}
pedido{ nped.ped → {fec.ped, DomEn.ped, ncli.ped = clientes' }
      =? nped.lped →>
      { nprd.lped =productos', cant.lped, importe :: prec*cant }
      → total :: sum(importe)
    }
]

```

Valoración de la notación

Esta especificación es estrictamente formal, al igual que lo son las propuestas por los métodos formales. Esto quiere decir que el programador que ha de construir la máquina, el sistema de software, no necesita conocer ningún significado de los términos utilizados. De hecho se podrían sustituir éstos por letras, a, b, c, etc. y se construiría la misma máquina. Únicamente se debería posteriormente cambiar los textos que aparecen en la pantalla. La especificación formal reduce drásticamente la posibilidad de que el programador malinterprete los requerimientos.

Frente a los métodos formales, la notación propuesta tiene tres ventajas y una limitación. En primer lugar la especificación es extremadamente concisa, característica tanto más deseable cuanto más grande sea el sistema a construir. Además, es sencilla e intuitiva para el diseñador, pues consiste en generar forms base (que son pantallas de ordenador) y superponerlas hasta obtener el documento deseado). Por último, es entendible por los usuarios que la deben validar, pues las fórmulas tiene una traducción directa e inmediata a las pantallas de ordenador, que es lo que el usuario realmente entiende. Esta última característica puede atenuar los malentendidos entre usuario y analista.

La limitación es que no tiene tanta capacidad expresiva como el cálculo de predicados. Esto la confina a una familia de problemas que desde el punto de vista computacional es muy pequeña –los sistemas de

software para la automatización de aplicaciones transaccionales de empresa, las que se realizan mediante formularios— pero de una importancia económica muy grande, tal como se mostró en la introducción de este trabajo.

Posiblemente no exista una solución universal, para todo tipo de problemas, al problema de conseguir especificaciones que sean formales, concisas, intuitivas y entendibles por los usuarios. Sin embargo, es posible encontrar buenas soluciones a tipos de problemas. Un ejemplo de una solución acotada pero brillantísima es la notación BNF, que hemos empleado para definir nuestro lenguaje, junto a toda la teoría de compiladores, de construcción de lenguajes de programación (Zave, 1996).

En cuanto a la completitud del cálculo propuesto, obviamente no tiene la capacidad expresiva de un lenguaje de programación, de una máquina de Turing. Esto es, no es computacionalmente completo. Por esta razón, aunque una vez construido el compilador, no es difícil traducirlo en programas ejecutables, no se propone como un lenguaje de programación. Comparte esta limitación con el modelo de datos relacional, pero ni siquiera es relacionalmente completo.

El cálculo propuesto tiene menor capacidad expresiva que el álgebra o cálculo relacional. Pero esto, dejando aparte las limitaciones que pueda tener debidas al mal diseño del mismo, a nuestro juicio, es una característica que debe poseer. En efecto, el álgebra y cálculo relacional están concebidos para ser lenguajes de interrogación de la base de datos, por ello deben ser capaces de generar de los axiomas todas las fórmulas satisfacibles. El cálculo que proponemos debe caracterizar únicamente las fórmulas válidas, que, obviamente, son un subconjunto de las primeras. Por lo tanto, es necesario que tenga menos capacidad expresiva.

La diferencia fundamental de la aproximación de los métodos de modelado conceptual frente al que proponemos se puede traducir en términos de axiomatización. Mientras los primeros se basan en una axiomatización euclidiana (los axiomas son verdades evidentes), en este trabajo se propugna una axiomatización hilbertiana (los axiomas no son evidentes, son una conjetura que será correcta si es capaz de generar los teoremas evidentes para los usuarios). Este es un procedimiento similar

al método científico, tal como lo defiende Popper (1963), el método hipotético-deductivo, o de conjeturas y refutaciones.

Al intentar comparar nuestra propuesta con otros métodos existentes tuvimos que enfrentarnos a dos problemas. El primero, ya citado en la introducción, es el gran número de métodos existentes. Esto se puede paliar agrupándolos en familias. Pero una vez hecho esto surgió un problema más difícil, que tiene algún parecido con el que señaló Kuhn sobre la inconmesurabilidad de los paradigmas subyacentes a teorías físicas que compiten entre sí. El problema es que aunque todos los métodos utilizan la noción de información, las características que le presuponen, muchas veces sin explicitarlas, son radicalmente distintas. La clave para descifrar el problema la encontramos en la historia de la filosofía. Nuestra piedra de Rosetta fue la ‘batalla de los universales’ que tuvo lugar entre distintos filósofos en la Edad Media en torno a decidir cuál es el referente (el objeto referenciado) por el significado de las palabras. Éste ha sido un problema recurrente y básico en toda la historia de la filosofía. En el próximo capítulo se presentan las distintas posiciones filosóficas y se infieren y comparan con las de las distintas familias de métodos en el área de sistemas de información.

Capítulo 4: La información como representación de la realidad⁹

Ron Weber, desde su tribuna de editor jefe de la revista MIS Quarterly, la de mayor relevancia del área de sistemas de información, destaca el hecho de que esta disciplina es de aluvión, pues se basa mayoritariamente en otras, como la informática (computer science), la ingeniería del software, el management, la economía, la sociología, la lingüística o la filosofía. Esto le lleva a preguntarse si existe algún campo de estudio que sea propio en exclusividad de los sistemas de información, que forme el núcleo identificador y discriminador frente a las otras disciplinas. Su conclusión es que la única clase de fenómenos que las otras materias no explican adecuadamente es la asociada a la construcción de modelos conceptuales de los sistemas de información y el diseño de bases de datos.

Para dicho autor la esencia de los sistemas de información es la ‘representación’ de la realidad, esto es, el modelo conceptual. La ‘representación’ permite el control de la realidad modelada sin necesidad de la observación directa de la misma. Por ejemplo, un sistema de información de gestión de pedidos, permite hacer el seguimiento del estado del cliente y de los productos pedidos sin tener que acudir al almacén ni al cliente (Weber, 2003a).

Si bien es cierto que posiblemente el modelado conceptual sea el núcleo central de la disciplina de sistemas de información, lo que no es tan claro es que la teorización de este núcleo de fenómenos sea explicada de manera autónoma por la disciplina. De hecho, lo que mostraremos es

⁹ Este capítulo fue publicado en una versión reducida a 12 páginas en la European Conference on Information Systems. (Sesé et al., 2006)

que la confluencia de diversas disciplinas en el área genera tal variedad de propuestas, con presupuestos metafísicos, meta-teóricos, heredados de la disciplina de origen, tan dispares que causan perplejidad. Esta disparidad va desde la Ingeniería Ontológica, que en algunos casos se apoya directamente en la metafísica y está próxima al idealismo platónico, hasta posturas de subjetivismo extremo, pasando por algunas aproximaciones que se parecen al positivismo lógico como el área de los métodos formales de especificación de software y a la tradicional de ingeniería del software, que adopta una posición que se podría calificar como realista.

Las presuposiciones metafísicas, que algunas veces son asumidas explícitamente por los autores, e incluso empleadas directamente en la propuesta de métodos, en la mayoría de los casos aparecen de forma implícita. Algunos autores simplemente ignoran o desprecian las consideraciones metafísicas, sin embargo, la adopción de una postura es inevitable aunque la califiquen de posiciones metateóricas.

En algunas disciplinas se puede obviar el problema. Un ingeniero de caminos, por ejemplo, no necesita plantearse ningún problema metafísico para desarrollar correctamente su trabajo técnico. Incluso en física, cuyo avance a veces pone en cuestión creencias metafísicas anteriormente asumidas, las discusiones metafísicas se soslayan, se sitúan fuera del ámbito de la disciplina, el método prevalece sobre la creencia. Hay multitud de casos de cuestionamiento de creencias metafísicas por parte de la física, piénsese por ejemplo en la comprobación experimental de Franklin de que los rayos eran corrientes eléctricas producidas por el rozamiento de masas de aire, que hizo muy poco plausible la posición ontológica, o mejor, mitológica, de que eran producidos por combates entre dioses; o en cómo la comprobación experimental de la teoría de la relatividad puso en entredicho la forma de conocimiento sintético a priori propuesto por Kant, al mostrar que uno de los ejemplos de dicho tipo de conocimiento, la geometría euclídea, no necesariamente era la del espacio real.

Un ejemplo actual de manifiesta discrepancia metafísica entre dos eminentes físicos, Roger Penrose y Stephen Hawking, se puede constatar en el siguiente fragmento:

Basically, he's (Roger Penrose) a Platonist believing that there's a unique world of ideas that describes a unique physical reality. I, on the other hand, am a positivist who believes that physical theories are just mathematical

models we construct, and that it is meaningless to ask if they correspond to reality, just whether they predict observations. (Hawking, 1997)

Sin embargo, este desacuerdo no les ha impedido colaborar fructíferamente en el desarrollo de teorías físicas.

En el área de sistemas de información la situación no es tan satisfactoria. A su elemento básico, a la información, que no es tangible como lo es la materia, y que constituye el elemento fundamental de la comunicación humana, se le atribuyen propiedades o características dispares, que surgen de asunciones metafísicas enfrentadas y que llevan a sus autores a métodos que en algunos casos son radicalmente distintos.

La divergencia en las concepciones va desde una propuesta como *‘The Semantic Web is not a separate Web but an extension of the current one, in which information is given well-defined meaning’*, hecha por el inventor del World Wide Web, Berners-Lee et al. (2001), hasta la consideración de que es un error asumir que las palabras tienen un significado preciso: *‘collapse of the myth of linguistic precision’* (Hirschheim et al., 1995)

En este capítulo se elabora un mapa de las distintas concepciones sobre la naturaleza de la información que son explícita o implícitamente asumidas por distintas comunidades profesionales relacionadas con los sistemas de información. Para ello se seguirá el procedimiento empleado por Burrell y Morgan (1979) para clasificar las distintas teorías de la organización de empresa. Dichos autores proponen una clasificación basada en componentes ontológicas, epistemológicas, metodológicas y acerca de la naturaleza humana, que explicitan mediante un breve recorrido por la historia de la filosofía, para luego configurar, mediante dos dimensiones conceptuales objetivismo-subjetivismo y orden-conflicto, cuatro paradigmas sociológicos que intentan englobar las distintas teorías de la organización.

Este esquema tiene la ventaja de agrupar a partir de presuposiciones básicas la gran variedad de teorías de la organización, basadas en teorías sociológicas, lo que permite una primera comprensión global. Sin embargo, su gran generalidad, conseguida por la indeterminación y ambigüedad de los conceptos, hace imprecisos los criterios clasificatorios (Deetz, 1996; Chua, 1986).

Dicho esquema ha sido adaptado al campo de los métodos de desarrollo de sistemas de información con distintos propósitos: para

*Propuesta de un método de validación de esquemas conceptuales
y análisis comparativo de la noción de información
en los métodos de desarrollo de sistemas de información*

proponer un paradigma de investigación (Iivari, 1986), para clasificar las distintas metodologías existentes (Hirschheim and Klein, 1989) (Iivari, 1991), para la clasificación de los distintos enfoques en modelos de datos (Hirschheim et al., 1995) y también, con las variantes introducidas por Chua (1986), para clasificar las distintas perspectivas adoptadas por los investigadores del área de sistemas de información por Orlikowski y Baroudi (1991).

Frente a la innegable utilidad de estos estudios por la misma razón que el de Burrell y Morgan, pero que también hereda las mismas limitaciones, hay una dificultad añadida producida por el trasplante de unas clasificaciones pensadas para estudiar una realidad social a otra realidad que aunque tiene, como se mostrará más adelante, un fuerte componente sociológico, también tiene otras características específicas.

Previo a una clasificación de metodologías en paradigmas basados en los presupuestos ontológicos y epistemológicos sobre la naturaleza de los sistemas de información, hay un supuesto mucho más básico, el de cuál es la naturaleza de la información. Las presuposiciones que se asuman condicionan el nivel de ambición y posibilidades de éxito de los proyectos de sistemas de información, así como la elección o diseño de la metodología que será empleada.

Esta clasificación que proponemos no pretende sustituir a la citada. Por ejemplo, la mayor parte de los enfoques que consideraremos caen dentro de lo que los anteriores autores llaman paradigma funcionalista.

Se presenta un mapa utilizando dos dimensiones, la ontológica y la epistemológica. La primera dimensión tiene dos extremos, la postura realista y la nominalista. Cabe destacar que cada uno de estos dos términos puede tener significados distintos dependiendo del ámbito a que se apliquen. No es lo mismo una postura ontológica respecto al mundo físico, al mundo social o con respecto a la información. Por ejemplo, el filósofo Searle es realista respecto al mundo físico. Esto se puede confirmar en (Searle, 1992), donde el autor hace un alegato contra el posmodernismo. Sin embargo, frente a la realidad social mantiene una posición convencionalista (Searle, 1995), que se aproxima al idealismo, y, teniendo en cuenta que considera el lenguaje como una institución social, también se puede afirmar que adopta una postura próxima al convencionalismo respecto a la información, que en este caso cabría llamar nominalismo. Otro ejemplo de ambigüedad, en este caso en la

dimensión epistemológica, es Bertrand Russell, que generalmente es considerado un empirista pero que, como veremos más adelante, su postura frente a la información está próxima al idealismo platónico, claramente asimilable a la tradición racionalista.

Por esta razón nos centramos únicamente en cómo conciben la información distintos grupos de investigadores. El método que emplearemos es el siguiente. En primer lugar, tal como hacen Burrell y Morgan (1979), elaboramos una clasificación basándonos en las distintas doctrinas filosóficas. En segundo lugar, siguiendo un procedimiento similar al utilizado en (Hirschheim et al., 1997), presentamos las diferentes posiciones asumidas por una serie de métodos del área de sistemas de información. El procedimiento consiste en interpretar las metodologías estudiadas basándose en los objetivos y métodos propuestos por sus creadores para intentar así explicitar las presuposiciones ontológicas y epistemológicas asumidas por los mismos, apoyando dicha interpretación en la presentación de fragmentos de textos significativos (Iivari et al., 1998; Iivari et al., 1999 ; Iivari et al., 2000)

En concreto pretendemos fijar el significado del término ‘concepto’, o más exactamente, mostrar que actualmente es utilizado por diversos autores del área de sistemas de información con distintos significados, atribuyéndole naturalezas distintas que guardan una notable semejanza con las distintas concepciones que ha tenido a lo largo de la historia en el campo de la filosofía.

En esta parte se pretende mostrar que las posiciones ontológicas asumidas tácitamente por distintas áreas de investigación actuales no se diferencian demasiado de las posiciones asumidas por estos filósofos en la antigua Grecia, planteamientos que se han ido repitiendo con variantes a lo largo de la historia de la filosofía.

Construcción del mapa: La dimensión ontológica

Los antiguos griegos

Si en el campo de las ciencias naturales, de las matemáticas o la medicina, se puede afirmar claramente que hemos superado a los antiguos griegos, a juzgar por la situación actual que mostraremos, es

dudoso realizar la misma afirmación en relación a las teorías sobre la naturaleza de la información. Parafraseando a Alfred North Whitehead, que afirmó que “*the safest general characterization of the European philosophical tradition is that it consists of a series of footnotes to Plato*” (Whitehead, 1929; p.39), en lo concerniente a la naturaleza de la información, la situación actual parece un conjunto de notas a pie de página de las disputas de los antiguos griegos.

Aunque el conjunto de posiciones filosóficas fue mucho más diverso y sutil, lo reduciremos a tres escuelas que fueron contemporáneas, cuyos representantes fueron Platón, Aristóteles y los sofistas Gorgias y Protágoras.

Platón estableció su teoría de las ideas intentando resolver el problema de definir el conocimiento. El conocimiento requiere la clasificación de los hechos y cosas. El conocimiento de hechos particulares no es relevante, por ejemplo, el conocimiento de las particularidades de mi perro. El conocimiento relevante, el conocimiento en sí, es sobre cosas o hechos generales. Por ejemplo, el conocimiento de las características de objetos generales, como los perros, o mejor expresado, de lo perruno. Estos objetos generales fueron llamados posteriormente *universales*.

El problema que se planteó Platón es que dado que no existen dos objetos particulares iguales, para clasificarlos en un mismo objeto general del cual podamos predicar características, antes debemos definir o descubrir cuales son las características esenciales, generales, de dichos particulares. Esta esencialidad, esta esencia, es a lo que llamó ‘idea’ o ‘forma’. Además, afirmó que estas esencias tenían una existencia independiente y anterior a los objetos concretos. En realidad, estos últimos no son más que un pálido reflejo de las Ideas, que son eternas y universales.

Según dicho filósofo se accede al conocimiento mediante la anamnesis, la reminiscencia de los conocimientos que todo ser humano ya tiene. Uno de los métodos para esta evocación es el socrático de la mayéutica, en el que el maestro no enseña al alumno sino que mediante oportunas preguntas hace que vaya recordando los conocimientos que ya alberga en sí mismo. Según resume la fórmula medieval, Platón afirmó que los universales tienen una existencia anterior a las cosas ‘*Universalia ante rem*’.

Aristóteles refutó las afirmaciones de Platón. Argumentó que las esencias existen en las cosas, no antes que ellas, *'Universalia in re'*. Las Ideas no son las causas primeras de los fenómenos. Aristóteles concluye su *Metafísica* afirmando:

'Again, it is not the ideal numbers that are the causes of musical phenomena and the like (for equal ideal numbers differ from one another in form; for even the units do); so that we need not assume Ideas for this reason at least.' (Aristotle, 350b B.C.; p.1728)

En la misma obra, Aristóteles también refuta la anamnesis. Al conocimiento, a la verdad, no se puede llegar sólo mediante la razón:

"And how could we learn the elements of all things? Evidently we cannot start by knowing anything before. For as he who is learning geometry, though he may know other things before, knows none of the things with which the science deals and about which he is to learn, so is it in all other cases. Therefore if there is a science of all things, such as some assert to exist, he who is learning this will know nothing before. Yet all learning is by means of premises which are (either all or some of them) known before,—whether the learning be by demonstration or by definitions; for the elements of the definition must be known before and be familiar; and learning by induction proceeds similarly. But again, if the science were actually innate, it were strange that we are unaware of our possession of the greatest of sciences. (Aristotle, 350b B.C.; p.1569)

En su libro *Últimos analíticos* propone el método empírico:

To resume our account of the right method of investigation: We must start by observing a set of similar —i.e. specifically identical— individuals, and consider what element they have in common. (Aristotle, 350c B.C.; p.160)

En la misma época de la antigua Grecia existió otra escuela, la de los sofistas, que mantenía posturas antagónicas a las platónicas y aristotélicas. Algunos sofistas sostenían posiciones relativistas y escépticas respecto al conocimiento. Se atribuye al sofista Gorgias una de las declaraciones más contundentes en contra del conocimiento generalizable:

There is no truth; if there were, it could not be known; if known, it could not be communicated. (Farrington, 1944; p.86)

*Propuesta de un método de validación de esquemas conceptuales
y análisis comparativo de la noción de información
en los métodos de desarrollo de sistemas de información*

En este breve enunciado se manifiesta el escepticismo más absoluto. En la primera frase es negada la posibilidad ontológica de la verdad; en la segunda, la posibilidad epistemológica; y en la tercera, la metodológica. Resumiéndola: la verdad es inexistente, inescrutable e inefable.

Protágoras, otro destacado sofista, lo expresó más sutilmente con el aforismo: *'Man is the measure of all things'*. (Farrington, 1944; p.86) Que suele ser interpretado como una postura subjetivista. El conocimiento objetivo no existe, siempre tiene una base subjetiva.

Platón y Aristóteles descalificaron la posición sofista argumentando que si afirmaban que la verdad no existe, entonces el único propósito que podían tener al argumentar en un debate era vencer al oponente, para lo que sería válido utilizar cualquier medio de persuasión, esto es, reduciendo todo el problema del conocimiento a una cuestión retórica, disciplina en la que los sofistas eran maestros.

Estas posiciones filosóficas enfrentadas se han ido repitiendo a lo largo de la historia. Por ejemplo, Kant, Husserl y Russell apoyaron las ideas platónicas. En su 'Crítica de la razón pura', Kant llama *idea* al concepto puro, al concepto de la razón, distinguiéndolo de los conceptos empíricos que se obtienen de la experiencia. Para ello recurre directamente a Platón:

Platón se sirvió del término idea, de suerte que se ve sin duda que entendía por él algo que no sólo no se toma nunca de los sentidos sino que además va mucho más allá de los conceptos del entendimiento, de los cuales se ocupaba Aristóteles, pues en la experiencia no se halla nunca nada congruente con ese algo. (Kant, 1781b; p.429)

Y, aunque critica la concepción mística de Platón, defiende su tesis de las ideas eternas e inmutables que 'ya están en nuestra propia cabeza':

Quien quisiera obtener de la experiencia los conceptos de la virtud y quisiera (como realmente hicieron muchos) convertir en modelo de fuente del conocimiento lo que en ningún caso puede servir sino de ejemplo para una explicación imperfecta, convertiría la virtud en algo absurdo y ambiguo, variable según los tiempos y las circunstancias, que no podría utilizarse para regla alguna. En cambio, todos nos percatamos de que si alguien nos es presentado como modelo de virtud, en nuestra propia cabeza se encuentra sin embargo el verdadero original con el cual comparamos ese presunto modelo y sólo de este modo lo apreciamos. (Kant, 1781b; p.430)

Y recrimina veladamente a los empiristas ingleses el que hubieran alterado el significado del término idea:

..ruego a aquellos que se toman a pecho la filosofía... tomen bajo su protección el término idea en su sentido originario. (Kant, 1781b; p.433)

A lo largo de la historia, diversos filósofos han mantenido posiciones antagónicas a las de Platón-Kant respecto a las ideas. Esta refutación se ha producido en tradiciones tan distintas como el empirismo de Locke, Berkeley y Hume, el pragmatismo de Pierce, la filosofía hermenéutica de Schleiermacher y Gadamer, los inclasificables Nietzsche y Wittgenstein, o los actuales autores posmodernos como Latour.

Uno de los ataques más directos a la concepción platónica de las ideas es el de Nietzsche en su ensayo 'Sobre verdad y mentira en sentido extramoral'. Nietzsche, además de especialista en filología clásica, fue profesor de retórica antigua (Nietzsche, 1872), la disciplina en la que los sofistas eran maestros. Después de una inteligente y bella argumentación concluye:

Pero pensemos especialmente en la formación de los conceptos. Toda palabra se convierte de manera inmediata en concepto en tanto que justamente no ha de servir para la experiencia singular y completamente individualizada a la que debe su origen, por ejemplo, como recuerdo, sino que debe encajar al mismo tiempo con innumerables experiencias, por así decirlo, más o menos similares, jamás idénticas estrictamente hablando; en suma, con casos puramente diferentes. Todo concepto se forma por equiparación de casos no iguales. ...Decimos que un hombre es "honesto". ¿Por qué ha obrado hoy tan honestamente?, preguntamos. Nuestra respuesta suele ser así: a causa de su honestidad. ¡La honestidad! Esto significa a su vez: la hoja es la causa de las hojas. Ciertamente no sabemos nada en absoluto de una cualidad esencial, denominada "honestidad", pero sí de una serie numerosa de acciones individuales, por lo tanto desemejantes, que igualamos olvidando las desemejanzas, y, entonces, las denominamos acciones honestas; al final formulamos a partir de ellas una qualitas occulta con el nombre de "honestidad". (Nietzsche, 1873)

Schleiermacher además de ser, como Nietzsche, un prominente filólogo clásico, también fue un precursor de la filosofía hermenéutica y una autoridad mundial en teología. Aunque, obviamente, sus concepciones teológicas difieren totalmente del ateísmo de Nietzsche,

*Propuesta de un método de validación de esquemas conceptuales
y análisis comparativo de la noción de información
en los métodos de desarrollo de sistemas de información*

coincide con éste en atacar las ideas platónicas. A pesar de ser un renombrado traductor de los escritos de Platón, Schleiermacher afirmó que los conceptos verdaderos no preexisten en el sentido platónico sino que son constituidos normativamente en las comunidades. (Schleiermacher, 1828)

Se pueden encontrar posturas parecidas a la anterior en otras corrientes filosóficas completamente distintas, como las de Peirce, el fundador del pragmatismo, quien aseveró que las palabras *'have become associated with their meanings by usage and live in the mind of those who use it'* (Peirce, 1894). O en el segundo Wittgenstein sosteniendo que *'for a large class of cases... the meaning of a word is its use in the language'* (Wittgenstein, 1953; p.20)

Actualmente, la actitud sofista parece ser la adoptada frente al conocimiento científico por algunos filósofos posmodernistas como Bruno Latour:

Rhetoric is the name of the discipline that has, for millennia, studied how people are made to believe and behave and taught people how to persuade others. Rhetoric is a fascinating albeit despised discipline, but it becomes still more important when debates are so exacerbated that they become scientific and technical.

This appeal to higher and more numerous allies is often called the argument of authority. It is derided by philosophers and by scientists alike because it creates a majority to impress the dissenter even though the dissenter 'might be right'. Science is seen as the opposite of the argument from authority. A few win over the many because truth is on their side. The classical form of this derision is provided by Galileo when he offers a contrast between rhetoric and real science. After having mocked the florid rhetoric of the past, Galileo opposed it to what happens in physics:

'But in the physical sciences when conclusions are sure and necessary and have nothing to do with human preference, one must take care not to place oneself in the defence of error; for here, a thousand Demosthenes and a thousand Aristotles would be left in the lurch by any average man who happened to hit on the truth for himself.'

This argument appears so obvious at first that it seems there is nothing to add. However, a careful look at the sentence reveals two completely different arguments mixed together. Here again the two faces of Janus we have

encountered in the introduction should not be confused even when they speak at once. One mouth says: 'science is truth that authority shall not overcome'; the other asks: 'how can you be stronger than a thousand politicians and one thousand philosophers?' On the left side rhetoric is opposed to science just as authority is opposed to reason; but on the right, science is a rhetoric powerful enough, if we make the count, to allow one man to win over 2000 prestigious authorities! (Latour, 1987; pp.30-31)

Latour parece reducir el método de contrastación experimental de la física a, aunque poderosa, simple retórica. Sokal y Bricmont (1998) realizaron una crítica minuciosa y contundente de algunas afirmaciones realizadas por autores posmodernos sobre la ciencia. La conclusión de estos autores se puede sintetizar con las palabras de Aristóteles cuando señala que algunos autores emplean la argucia retórica de la ambigüedad *'cuando no tienen nada que decir, pero quieren dar la impresión de que dicen algo'*. (Aristóteles, 350 B.C.; p.254)

El debate medieval sobre los universales y el triángulo del significado de Ogden-Richards

Las disputas griegas respecto al conocimiento reaparecieron en la Edad Media en torno a la existencia de los universales (las Ideas, Formas o Esencias) en forma de tres doctrinas a las que los historiadores han llamado Realismo, Conceptualismo y Nominalismo. El conflicto ontológico de la Edad Media buscando una teoría que explicase cómo adquieren las palabras su significado puede resumirse utilizando el triángulo del significado de Ogden-Richards (citado en (Sowa, 1984) (Verrijn-Stuart, 2001))

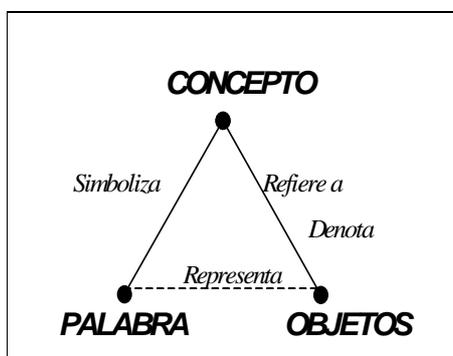


Diagrama 11. Triángulo del significado de Ogden-Richards

En este esquema se diferencia entre los objetos concretos, las palabras y los conceptos. El vértice izquierdo es la palabra, el término, el símbolo, el signo o significador. La palabra simboliza el vértice superior, que es el concepto, la idea, esencia o universal, la intensión, el pensamiento o el sentido, o el significado; en el vértice derecho tenemos el objeto u objetos, la extensión, lo referido, el referente o lo significado. Las palabras simbolizan los conceptos y éstos se refieren a los objetos. Las palabras representan a los objetos.

El Realismo, tanto el platónico como el aristotélico, afirma la existencia de los conceptos, llamados ideas por los primeros y esencias por los segundos. En esta tradición filosófica se asume que las ideas o esencias tienen una existencia independiente de la consciencia humana. Los platónicos construyen ontologías, los aristotélicos emplean taxonomías.

Para el conceptualismo los conceptos son inventados por el ser humano y reflejan en mayor o menor medida similitudes entre distintos objetos, pero de alguna manera son precisos e independientes del contexto y la situación.

Los nominalistas afirman que no hay conceptos, sólo palabras que actúan como términos generales representando, a veces de manera indeterminada, a un conjunto de individuos particulares. Para esta doctrina los términos generales no tienen significados intrínsecos que se puedan desenmarañar, sólo un uso fluctuante de las palabras cuyo significado se debe promediar. (Quine, 1990) (Wilshire, 2002)

Estas tres posturas básicas en torno a los universales (el significado o esencia de los conceptos) es transversal a muchas doctrinas filosóficas. Por ejemplo, entre los filósofos fenomenológicos, la postura de Husserl, con su método de reducción eidética, se sitúa en un extremo. Dicho método, no muy distante de la mayéutica y la anamnesis de Platón, se utiliza para descubrir la ‘esencia pura de las cosas’, a las que Husserl también llama ‘esencias invariantes’, o ‘universales’, o *eidós*, que en griego significa ‘forma’ (Smith, 2003). En el otro extremo está Gadamer, que a veces es adscrito a la tradición fenomenológica, pues fue discípulo de Heidegger, quien, a su vez, fue alumno de Husserl. Sin embargo, la postura de Gadamer está más próxima al nominalismo, que se corresponde con su afiliación a la doctrina hermenéutica de Dilthey y Schleiermacher, cuya tesis central es el ‘círculo hermenéutico’, en el que el significado de las palabras depende del contexto (entendiendo ‘contexto’ en su acepción más amplia) y viceversa.

Para acabar de delimitar las tres posturas básicas y de paso mostrar la transversalidad de las posiciones respecto a los universales dentro de una misma tradición filosófica, presentaremos las afirmaciones de los principales empiristas británicos, Locke, Berkeley, Hume y Russell. La ventaja de basarse en estos filósofos es que, además de ser claros y directos (a diferencia de la opacidad de los argumentos y conceptos de algunos fenomenólogos), hacen referencia a este problema de forma explícita y extensa.

El empirismo británico: Locke, Berkeley, Hume y Russell

El inglés John Locke adoptó el término idea de Platón pero cambiándole el significado (lo que provocó, como hemos visto, el reproche de Kant) al rechazar su independencia de la mente humana. Consideró una idea como “*some immediate object of the mind, which it perceives and has before it, distinct from the sound it uses as a sign of it*” (Locke, 1690; p.14) y no como algo independiente del ser humano.

En el capítulo que dedica a los términos generales ‘*Of General Terms*’ se aleja de las actitudes platónicas y aristotélicas, y defiende una forma de conceptualismo. Los términos generales, los universales, son una construcción humana ‘*the workmanship of the understanding*’:

“General and universal are creatures of the understanding, and belong not to the real existence of things. [...] it is easy to observe, that the essences of

the sorts of things, and, consequently, the sorting of things, is the workmanship of the understanding that abstracts and makes those general ideas.” (Locke, 1690; p.372)

Pero, aunque productos del entendimiento humano, considera a los conceptos como algo concreto, esto es, reifica los conceptos, las ‘ideas abstractas’, como intermediarios entre las palabras y las cosas. Los conceptos tienen significados precisos:

“...when general names have any connexion with particular beings, these abstract ideas are the medium that unites them: so that the essences of species, as distinguished and denominated by us, neither are nor can be anything but those precise abstract ideas we have in our minds.’ (Locke, 2004; p.373)

El obispo irlandés George Berkeley rechazó la noción de ‘idea abstracta’ de Locke. En su libro ‘*Principles of Human Knowledge*’ se pregunta ‘*what it is that inclines the men of speculation to embrace an opinion so remote from common sense as that seems to be.*’ (Berkeley, 1710; p.12) Y fundamenta la persistencia de los significados en el uso ‘*constant and familiar use*’ (Ibid.; p14), rechazando la precisión de los conceptos:

“...it is thought that every name has, or ought to have, one only precise and settled signification, which inclines men to think there are...abstract, determinate ideas that constitute the true and only immediate signification of each general name... Whereas, in truth, there is no such thing as one precise and definite signification...they all signifying indifferently a great number of particular ideas” (Ibid. p.18)

El escocés David Hume llevó al empirismo hasta sus límites al proseguir la obra de Berkeley pero despojándola de sus componentes teológicos. También dedicó un capítulo a las ideas abstractas ‘*Of Abstract Ideas*’ en su obra ‘*A Treatise of Human Nature*’ en el que haciendo referencia a la posición nominalista de Berkeley la ratifica y afirma ‘*As I look upon this to be one of the greatest and most valuable discoveries that has been made of late years in the republic of letters, I shall here endeavour to confirm it by some arguments, which I hope will put it beyond all doubt and controversy.*’ (Hume, 1740; p.64) En su libro posterior ‘*An Enquiry concerning Human Understanding*’, aunque mostrando alguna perplejidad por la apariencia de precisión, exactitud y aprioricidad de la geometría, remarca:

‘...that there is no such thing as abstract or general ideas, properly speaking; but that all general ideas are, in reality, particular ones, attached to

a general term, which recalls, upon occasion, other particular ones, that resemble, in certain circumstances, the idea, present to the mind' (Hume, 1758; p.205).

Sorprendentemente, Bertrand Russell, que generalmente es calificado de empirista (Crawshay-Williams, 2003; Reale and Antiseri, 1983), y también él mismo se autocalifica de esta manera *'I, as an empiricist...'* (Russell, 1908; p.69), respecto a los conceptos abandona la tradición empirista y suscribe el realismo platónico. Su argumentación es análoga a la de Kant.

Por ejemplo, dice Russell (1912) consideremos la noción de justicia. Para explicar lo que es esta noción, lo natural es mostrar una serie de actos justos para que nuestro oyente capte lo que tienen en común. Todos ellos comparten una naturaleza común que no está en los actos que no son justos. Esta naturaleza común que les hace ser actos justos es la esencia de la justicia. Pero, cada acto justo particular no es idéntico a otro semejante ni a la idea de justicia, sólo se aproxima en mayor o menor medida a ella. Por lo tanto, la esencia de la idea de justicia no es algo que esté enteramente en cada particular, luego debe ser algo universal que no está en el mundo de los sentidos, de los hechos u objetos particulares.

Hasta aquí, la postura de Russell se puede caracterizar como 'conceptualista', reifica una propiedad de un conjunto de actos u objetos, la convierten en algo independiente de los particulares, da existencia a entes abstractos. Pero, ¿dónde existen estos entes abstractos? ¿en las mentes de las personas? —se pregunta Russell. Y ofrece una argumentación que hipostasía los conceptos, les atribuye una existencia independiente, no sólo de los objetos particulares sino de las mentes de las personas. Si no fuera así, arguye, sería imposible la comunicación humana, pues un acto de pensamiento es necesariamente diferente de una persona a otra, incluso diferente en la misma persona en distintos momentos. Por ello, concluye Russell, las Ideas o, como les llama él, los universales, 'no son transitorios y cambiantes como los objetos de los sentidos: son eternamente ellos mismos, inmutables e indestructibles'.

The 'idea' justice is not identical with anything that is just: it is something other than particular things, which particular things partake of. Not being particular, it cannot itself exist in the world of sense. Moreover it is not

fleeting or changeable like the things of sense: it is eternally itself, immutable and indestructible. (Russell, 1912; p.52)

Platón afirmó que la verdadera realidad es el mundo de las ideas, de las formas. El espíritu humano puede aprehenderlas, pues mediante el razonamiento puede ‘recordarlas’. Russell tacha de mística esta afirmación y ofrece una versión más moderada recurriendo a la distinción entre esencia y existencia.

We shall find it convenient only to speak of things existing when they are in time, that is to say, when we can point to some time at which they exist (not excluding the possibility of their existing at all times). Thus thoughts and feelings, minds and physical objects exist. But universals do not exist in this sense; we shall say that they subsist or have being, where 'being' is opposed to 'existence' as being timeless. The world of universals, therefore, may also be described as the world of being. The world of being is unchangeable, rigid, exact, delightful to the mathematician, the logician, the builder of metaphysical systems, and all who love perfection more than life. (Russell, 1912; p.57)

Russell escribió esta obra en 1912. En su libro ‘*An Inquiry into Meaning and Truth*’, publicado en 1950, continuaba defendiendo los universales (Russell, 1950).

En el siglo XX volvió a reaparecer el mismo debate en el ámbito de las matemáticas formándose tres posiciones enfrentadas cuyos nombres son Logicismo, Intuicionismo y Formalismo. Son exponentes de la primera postura Frege, Russell, Whitehead, Church y Carnap; de la segunda, Poincaré, Brouwer y Weyl; y de la tercera, Hilbert. Sus diferentes posiciones se pueden reducir a que los logicistas asumen que, en el ámbito de las matemáticas, las clases existen y son descubiertas por los investigadores, los conceptualistas creen que son inventadas por la mente humana, y los terceros, los formalistas, consideran las matemáticas como un juego puramente sintáctico de notaciones sin significado, que a pesar de eso pueden resultar útiles en el desarrollo de la física y la tecnología (Quine, 1953). La envergadura intelectual de cualquiera de los nombres citados invita cuando menos a la prudencia a la hora de decantarse por una posición.

Frege muestra explícitamente su postura logicista (platónica) (Frege, 1918). Para él, el vehículo de la comunicación no son los términos, sino

las sentencias, mediante las cuales se transmiten significados. A estos significados transmitidos por las sentencias Frege los llama ‘gedanke’. Como ya dijimos en la introducción, emplearemos el término ‘proposición’ para traducirlo.

Frege utiliza el ejemplo del teorema de Pitágoras para mostrar que este teorema no necesita una consciencia subjetiva para ser cierto, ya existe antes de que haya sido descubierto por alguien, igual que un planeta. Aunque no lo afirma explícitamente, es el mundo de las formas, de las esencias, de las ideas de Platón.

Thus the thought, for example, which we expressed in the Pythagorean theorem is timeless true, true independently of whether anyone takes it to be true. It needs no bearer. It is not true for the first time when it is discovered, but is like a planet which, already before anyone has seen it, has been in interaction with other planets. (Frege, 1918; p.96)

Para Frege, el teorema de Pitágoras es cierto a priori, no hace falta hacer ninguna prueba experimental para comprobar que es verdadero. Se puede demostrar a partir de los cinco postulados de Euclides, que son verdades de la razón, evidentes. Este debate pervive en el área de las matemáticas planteado en otros términos (Shapiro, 2005).

Palabras y objetos. Distinción entre idealismo y nominalismo

Los trabajos citados de Iivari, Hirschheim, Klein y Lyytinen, que se basan en la propuesta de Burrell y Morgan, caen en el error de confundir –o al menos, de no distinguir—entre idealismo y nominalismo. El esquema que Burrell y Morgan se aplica al campo de las teorías de la organización. Para adaptarlo al campo de los sistemas de información, Iivari et al. amplían los ejes clasificatorios de Burrell y Morgan con una nueva dimensión, la de información/datos, presentando una dicotomía: los ‘hechos descriptivos’ y los ‘significados constitutivos’. Estos autores presentan las dos posiciones como opuestas: *‘the difference is whether one believes that a data model ‘reflects’ reality, or consists of subjective meanings and thereby constructs reality’* (Klein and Hirschheim, 1987; p.9). Pero esta dimensión es claramente la de realismo/idealismo, que, como explicamos a continuación, es ortogonal a la de realismo/nominalismo. Este malentendido se agrava cuando (Burrell and Morgan, 1979; p.4) utilizan

*Propuesta de un método de validación de esquemas conceptuales
y análisis comparativo de la noción de información
en los métodos de desarrollo de sistemas de información*

los términos realismo/nominalismo de un modo poco usual. Llaman nominalismo a lo que normalmente se entiende como idealismo.

Las dimensiones idealismo/realismo y nominalismo/realismo, aunque a veces se confunden, son independientes. El idealismo sostiene que *'whatever exists, or at any rate whatever can be known to exist, must be in some sense mental'* (Russell, 1912; p.19) En esta definición se pueden distinguir dos tipos de idealismo, el idealismo ontológico, que afirma que la existencia de los objetos depende de que alguien los observe; y el idealismo epistémico (el relacionado con el conocimiento), que mantiene que no tenemos acceso a 'la cosa en sí' (el kantiano *ding an sich*) y por lo tanto, todo lo que conocemos es una construcción mental. Un representante del primer tipo de idealismo es Berkeley, quien postuló que sólo lo que es percibido existe (*'esse est percipi'*). Un representante de la segunda forma de idealismo es Kant, que argumentó que nuestra percepción de la realidad es el producto de los filtros a priori de la cognición humana, y, por lo tanto, tratar de entender la naturaleza última de las cosas es un propósito inútil.

El opuesto conceptual del idealismo es el realismo, que también es multiforme. Por ejemplo, una forma de realismo es el materialismo, que proclama que el componente básico del mundo es la materia. Otra posición realista es el 'realismo ingenuo' —opuesto al idealismo de Kant— que declara que la realidad última es tal como la percibimos.

Esta dimensión realismo/idealismo es ortogonal a la que hemos presentado en torno a los universales. La dicotomía realismo/nominalismo es una posición ontológica referente a la existencia de los conceptos. Llamemos realismo_O a la posición que se opone al idealismo, la que dice que los objetos materiales existen, y realismo_P, a la opuesta al nominalismo, la que mantiene que el significado de las palabras existe independientemente de la cognición humana. Veamos un ejemplo que muestra la independencia de las dos dimensiones. Dentro del empirismo británico se encuentran todas las posibles combinaciones: (realismo_P, realismo_O), (realismo_P, idealismo), (nominalismo, idealismo) y (nominalismo, realismo_O). En el siguiente diagrama se resume las posiciones de los filósofos de dicha escuela. Russell es realista_P, pero Locke, Berkeley y Hume son nominalistas. Sin embargo, mientras que Berkeley es idealista, Hume, Locke y Russell son realistas_O. La postura de Hume es discutible debido a la naturaleza

escéptica de su doctrina, sin embargo, algunos autores lo adscriben a la postura realista (Beauchamp, 1999). El cuadrante (realismo_O, idealismo) está representado por Platón, Kant y Husserl. Los tres están de acuerdo en defender la existencia de los universales (las ideas platónicas), el realismo_O.

Realismo _O	Idealismo	Platón Kant Husserl	Berkeley
	Realismo _O	Russell	Locke Hume
		Realismo _P	Nominalismo

Diagrama 12. Independencia de las dimensiones Idealismo y Nominalismo

Los conflictos filosóficos entre el realismo_O y el idealismo son más relevantes en el mundo de la sociología y, aunque el estudio de los sistemas de información también tiene componentes sociológicos, dado que el área de modelos conceptuales, que es el que nos ocupa en este trabajo, es el estudio de la faceta señalada por Goldkuhl y Lyytinen (1982) que estudia los sistemas de información como ‘...*formal linguistic systems for communication between people which support their actions*’, es mucho más relevante el estudio de la dimensión realismo_P / nominalismo.

Construcción del mapa: La dimensión epistemológica

Burrell y Morgan caracterizan la dimensión epistemológica mediante dos extremos, el positivismo y el anti-positivismo. Utilizan el término positivismo para caracterizar las epistemologías que intentan predecir y explicar los fenómenos del mundo social buscando regularidades con métodos similares a los de las ciencias naturales. Por anti-positivismo entienden los métodos cuyo propósito es ‘comprender’ lo que ocurre en las sociedades mediante la ocupación del universo simbólico, del marco de significados, de los actores de dicha sociedad.

Esta dimensión epistemológica, que puede ser relevante para distinguir distintas tradiciones en ciencias sociales, obviamente está fuera de lugar para ser aplicada a las ciencias naturales. Para nuestro problema de los conceptos tampoco parece adecuada, no permite discriminar las diferencias entre Aristóteles y Platón, ambos realistas (las esencias existen ‘ahí fuera’, existen independientemente de la mente humana, pero pueden ser descubiertas por ella). No es adecuada cuando la aplicamos al problema de la información, al cómo se puede descubrir, capturar o fijar el significado de los conceptos, que son los vehículos para transmitir información.

Dado que el conocimiento, los juicios, también tienen como vehículo los conceptos, parece más adecuada la división utilizada por Kant, entre el conocimiento a priori y a posteriori, que son las etiquetas que emplea para distinguir entre el racionalismo de Platón y Leibnitz (la razón y la deducción lógica son la única fuente segura del conocimiento) y el empirismo de Aristóteles y Hume (la experiencia y la observación son la fuente del conocimiento), o para emplear los nombres con que los califica, entre la postura dogmática y la escéptica.

Dogmático viene de dogma, que en sentido etimológico significa ‘doctrina fijada’. Por tanto, sería un término adecuado para calificar la posición de algunos autores que presentamos a continuación, pero dado que tanto el término ‘dogmático’ como el ‘escéptico’ tienen connotaciones peyorativas. Y dado que nuestro propósito es eminentemente práctico, definimos las posiciones ‘preceptivo’ y ‘pragmático’. Llamaremos ‘preceptivos’ a aquellos autores que para desarrollar sus propuestas parten de una doctrina canónica, tal como la metafísica analítica, o también de un conjunto de preceptos y reglas, tales como la lógica formal.

Llamaremos ‘pragmáticas’ a aquellas posiciones más escépticas, que muestran algún rasgo de agnosticismo epistémico y proponen soluciones *ad hoc* a los problemas. Este sentido que proponemos del término es el que se utiliza vulgarmente, aunque el DRAE no lo recoge; esto es, con el término ‘pragmático’ no queremos significar que la posición sostenga las doctrinas del pragmatismo filosófico. Ejemplos de solución de problemas ‘pragmática’ la ejemplifican Stroustrup, el diseñador del lenguaje de programación C++, y Gabriel, uno de los diseñadores de CLOS (otro lenguaje de programación orientado a objetos). Cuando se

les preguntó qué reglas utilizaban para identificar las clases (los conceptos), Stroustrup contestó: *It's the Holy Grail. There is no panacea*. Gabriel respondió: *That's a fundamental question for which there is no easy answer. I try things*. (Booch, 1994; p.145)

Esta dimensión epistemológica nos permite la construcción de un mapa con seis posibles posiciones mostradas en la tabla 1. Una revisión de la literatura nos permitió identificar prominentes autores que ejemplifican las seis posturas. El propósito de esta revisión no ha sido tanto obtener una clasificación exhaustiva —objetivo a nuestro juicio inalcanzable pues más que clases disjuntas existe un continuo de posiciones— como el mostrar la sorprendente diversidad de posiciones mantenidas por investigadores actuales.

Para realizar la revisión se interrogó a los artículos con las siguientes preguntas relativas a la naturaleza de la información como representación: ¿Existen ahí afuera conceptos independientes de la cognición humana mediante los cuales se describe la realidad? Y, si es así, ¿existen procedimientos que permiten descubrir dichos conceptos? En el caso de negar la primera pregunta, si se mantiene una postura de relativismo conceptual, esto es, en el supuesto de que los conceptos se consideren una creación humana, ¿tienen un significado preciso, invariable y son independientes del contexto? ¿existe algún criterio para validar dicha precisión e independencia? En el caso de que el autor niegue cualquier clase de objetividad a los conceptos, esto es, en el caso de que la postura suscriba, además del relativismo conceptual, el relativismo semántico, ¿existe algún método para resolver el problema? La Tabla 1 resume los rasgos básicos de cada posición.

Posición	Naturaleza atribuida al significado	Cuerpo de preceptos
1. Realismo Preceptivo	Ideas platónicas. Existen 'ahí afuera' y se descubren a priori.	Metafísica analítica
2. Realismo Pragmático	Clases. Existen 'ahí afuera' y se descubren a posteriori.	Empirismo. Método Ad Hoc
3. Conceptualismo Preceptivo	Conceptos. Invención humana. Relativismo conceptual. Los significados son precisos y de alguna manera son independientes del contexto.	Filosofía
4. Conceptualismo Pragmático		Ad Hoc.

5. Nominalismo Preceptivo	Términos generales. Invención humana. Relativismo conceptual y semántico. Los significados son difusos y dependientes del contexto y la situación.	Lógica matemática. El significado se define.
6. Nominalismo Pragmático		Ad Hoc. El significado se negocia.

Tabla 1. Rasgos básicos de las distintas posiciones.

A continuación presentamos los distintos ejemplares, paradigmas, de las posiciones de la tabla anterior. El orden de presentación no coincide con el analítico de la tabla. Hemos preferido presentarlos siguiendo un orden de afinidades conceptuales. Así, primero presentamos el realismo preceptivo, que utiliza las ontologías como constituyentes de las especificaciones de los sistemas de información, y continuamos con el conceptualismo pragmático, que aunque desde distintos presupuestos, también utiliza las ontologías para constituir las especificaciones. Seguimos con el conceptualismo preceptivo que emplea los esquemas conceptuales como especificación pero se apoya en la ontología como disciplina para sustentar sus propuestas. De éste pasamos al realismo pragmático que ignora las ontologías pero utiliza los diagramas de clases, que son similares a los esquemas conceptuales. Posteriormente, tratamos el nominalismo preceptivo, que es percibido por algunos autores de la posición anterior como un ideal de rigor matemático en su disciplina. Y, por último, acabamos con el nominalismo pragmático.

Los distintos paradigmas en el área de sistemas de información

Paradigma de Realismo Preceptivo (a priori): Ingeniería ontológica

Ante la diversidad de campos en donde se aplican los métodos ontológicos, no es de extrañar que presenten una gran diversidad y ambigüedad en su definición (Smith and Welty ,2001) (Zúñiga, 2001). Veremos ahora una de las formas más extremas de aplicación.

En la introducción a la *Second International Conference on Formal Ontology and Information Systems*, Smith y Welty (2001) lamentan la divergencia en las definiciones del término ‘ontología’ y proponen la ‘ontología filosófica’ (a la que también llaman ‘metafísica analítica’):

“...what we now refer to as philosophical ontology has sought the definitive and exhaustive classification of entities in all spheres of being. It can

thus be conceived as a kind of generalized chemistry. The taxonomies which result from philosophical ontology have been intended to be definitive in the sense that they could serve as answers to such questions as: What classes of entities are needed for a complete description and explanation of all the goings-on in the universe? Or: What classes of entities are needed to give an account of what makes true all truths? They have been designed to be exhaustive in the sense that all types of entities should be included, including also the types of relations by which entities are tied together. ...information systems ontology is itself an enormous new field of practical application that is crying out to be explored by the methods of rigorous philosophy”

La conclusión final que presentan es que los informáticos (computer scientists) ‘están empezando a reconocer que el establecimiento, de una vez y para siempre, de una ontología de referencia general y robusta produciría ventajas significativas frente a las soluciones ad-hoc, los métodos desarrollados para casos particulares que se han utilizado hasta la fecha.’

Dentro de este grupo de investigadores destaca, tanto por su notoriedad como por su esencialidad metafísica, Nicola Guarino. Frente a lo que se entiende por ‘ontología formal’ en el ámbito de la Inteligencia Artificial, ‘un artefacto de ingeniería basado en lógica de predicados de primer orden que especifica los significados de los términos de un vocabulario que describe una realidad’, este autor reserva el término para un campo de investigación filosófica (Guarino, 1995; Guarino, 1998) que puede ser entendido como una teoría de las ‘distinciones a priori’ entre entidades del mundo y las categorías de orden superior que se utilizan para crear modelos del mundo (Guarino and Welty, 2002; Guarino and Welty, 2004).

Su propósito es convertir la ontología, una ‘arcana forma de arte’, en una rigurosa disciplina de ingeniería: la ingeniería ontológica. Para ello introducen ‘nociones formales’ como esencia, permanencia o rigidez. Por ejemplo, definen ‘esencia’ y ‘permanencia’

The first formal notion we will discuss is essence. A property of an entity is essential to that entity if it must hold for it. This is a stronger notion than one of permanence, that is, a property of an entity is not essential if it just happens to be true of it, accidentally, for all time. For example, consider the property of being hard. We may say that it is an essential property of hammers, but not of sponges. Some sponges (dry ones) are hard, and

some particular sponge may be hard for its entire existence, however this does not make being hard an essential property of that sponge. The fact is that it could have been soft at some time, it just happened that it never was.

En el artículo que fue premiado como el mejor de la 19 International Conference on Conceptual Modeling ER-2000, los autores desarrollan otras nociones ontológicas formales (Guarino and Welty, 2000). También hay propuestas de cómo debe ser el proceso de ingeniería ontológica (Kishore et al., 2004b)

Paradigma Conceptualista Pragmático: Diseño de ontologías

Otro destacado autor del área de las ontologías es Thomas Gruber (1995). Para este autor, una ‘conceptualización’ es una visión abstracta y simplificada del mundo que queremos representar con algún propósito. Una ontología es una especificación de la conceptualización. En la práctica, una ontología comunitaria (*common ontology*) consiste en un vocabulario que permite la comunicación entre distintos agentes que han realizado un compromiso ontológico, esto es, que se han puesto de acuerdo en utilizar un vocabulario compartido de una manera consistente y coherente.

Los conceptos, el vocabulario compartido, son una invención humana, el resultado de una elección: *Formal ontologies are designed. When we choose how to represent something in an ontology, we are making design decisions.* Y su posición es pragmática, niega explícitamente el razonamiento a priori para decidir la bondad de una conceptualización: *To guide and evaluate our designs, we need objective criteria that are founded on the purpose of the resulting artifact, rather than based on a priori notions of naturalness or Truth.*

Hasta aquí, se podría calificar al autor como nominalista, a fin de cuentas considera que la ontología es un conjunto de términos definidos que tienen el significado pretendido, del que se espera un uso mínimamente consistente y coherente para que la comunicación entre los agentes sea eficaz. Sin embargo, da un giro conceptualista al reificar, al objetivar, el significado de los términos y hacerlos independientes del contexto: *An ontology should effectively communicate the intended meaning of defined terms. Definitions should be objective. ...the definition should be independent of social or computational context. Formalism is a means to this end. When a definition can be stated in logical axioms, it should be. ...All definitions should be documented with natural language.*

Un ejemplo cristalino de la diferencia entre una posición conceptualista, frente a la nominalista, en la que los conceptos son independientes de los términos empleados para designarlos nos la ofrecen

Ontology is a representation vocabulary, often specialized to some domain or subject matter. More precisely, it is not the vocabulary as such that qualifies as an ontology, but the conceptualizations that the terms in the vocabulary are intended to capture. Thus, translating the terms in an ontology from one language to another, for example from English to French, does not change the ontology conceptually. (Chandrasekaran et al., 1999)

Y en (Spyns et al., 2002) ‘...a conceptualisation of a domain ontology is supposed to consider the subjects separately from the problems or tasks that may arise or are relevant for the subject’

‘Ontologies provide a means for defining the "concepts" and therefore the semantics of the data to be exchanged.’ (Bussler et al., 2002)

No obstante, la posición de Gruber no es dogmática. Reconoce que no está claro cómo puede un grupo de personas llegar a un consenso en las conceptualizaciones comunes, ni cómo se pueden definir los significados de los términos independientemente de su contexto de utilización. Y defiende su posición conceptualista manifestando que *‘The utility of common ontologies as a sharing mechanism is a hypothesis, the subject of collaborative studies.’* (Gruber, 1993)

En (Kishore et al., 2004a; Sharman et al., 2004) se presenta un *survey* de este tipo de ontologías, que los autores llaman ontologías computacionales.

Paradigma Conceptualista Preceptivo

Para representar este paradigma mostraremos algunas de las concepciones de dos autores relevantes en el área de sistemas de información, Yair Wand y Ron Weber, que en los últimos quince años han elaborado una teoría para juzgar y guiar el desarrollo de modelos conceptuales.

Sintetizan el proceso de construcción de sistemas de información como la traducción de las percepciones de los usuarios a un texto (script) ejecutable por un ordenador. Consideran que *‘the first step in this process is*

to create a well-defined model of the real system (as perceived). This is the conceptual model.' (Wand et al., 1995)

Además de servir como especificación de lo que se ha de construir, el modelo conceptual también es útil, según los autores, como instrumento de comunicación entre los analistas del sistema y los usuarios puesto que *'Conceptual models or semantic data models were developed to capture the meaning of an application domain as perceived by someone.'* (Wand et al., 1999)

El problema del modelado conceptual, según Weber (2003a), es que, a pesar de que se ha investigado mucho, no existe una teoría sólida que la sustente, a diferencia de la teoría relacional de normalización de datos, que sí está basada en sólidas bases teóricas. Para solucionar el problema proponen fundamentar la teoría de modelado conceptual en la ontología del filósofo y físico Mario Bunge.

Bunge reports that his aim was to draw upon ample ontological traditions arising from the work of Aristotle, Aquinas, Descartes, Spinoza, Leibniz, Hobbes, Helvetius, d'Holbach, Lotze, Engles, Peirce, Russell, and Whitehead. He further claims his work differs in a "matter of method" because it also draws upon contemporary research and has been elaborated using mathematics. Bunge intends his ontology to be "both exact and scientific." (Wand et al., 1999; p.497)

En (Wand, 1989; p.539) son presentados algunos de los principios ontológicos de dicho filósofo

- *The world is composed of things. Consequently, the sciences of reality (natural or social) study things, their properties and changes.*
- *Forms are properties of things... properties are represented by predicates defined on domains that are, at least in part, sets of concrete objects.*
- *Nothing comes out of nothing and no thing reduces to nothingness.*
- *Every thing abides by laws. Whether natural or social, laws are invariant relations among properties.*
- *The world is viewed as composed of things or two kinds: concrete things that are called entities or substantial individuals, and conceptual things. For our purpose we deal with substantial individuals, and it is important that they are concrete things and not concepts or types of things.*

Estos principios parecen indicar que el ámbito de dicha ontología es el de las ‘ciencias de la realidad’, de los objetos materiales (‘individuos substanciales’, cosas materiales). En el último principio, podemos leer, se excluye explícitamente de la ontología los conceptos o tipos de cosas.

La aplicación de esta ontología, que pretende ser realista, a una realidad de los sistemas de información que tiene un fuerte componente social y subjetivo produce algunos problemas y se ven en la necesidad de ampliarla, tal como se puede ver en (Wand et al., 1999; p.497,) en donde establecen un conjunto de reglas para el modelado conceptual basadas en la ontología. De los cuatro postulados que proponen en este artículo el primero es:

POSTULATE 1. The world is made of things that possess properties.*

Ontology distinguishes between concrete things, which are called substantial individuals or entities, and conceptual things (e.g., mathematical concepts such as sets and functions). We assume that any domain can be described in terms of concrete things and the linkages that exist among them. Therefore, we use the word “thing” to refer only to substantial individuals or concrete things.

Pero dado que un sistema de información debe reflejar la realidad, no tal cual es sino tal como la perciben los usuarios, se ven obligados a dar un giro subjetivista y ampliar el concepto de cosa no sólo a lo que existe objetivamente (según Bunge) sino a lo que existe en la mente de alguien y a lo que existe como realidad social, por ejemplo una cuenta bancaria.

We further assume that domain modeling is based upon someone’s view of existing or possible reality. Therefore, the notion of a concrete thing applies to anything perceived as a specific object by someone, whether it exists in physical reality or only in someone’s mind. In this light, a bank account is considered a thing, as well as a product that has been designed but not yet produced. Both are concrete things in someone’s mind.

Esta tensión ontológica queda explicitada también en (Wand, 1997; p.39) en donde afirma que hay nociones que existen únicamente en la ‘realidad social’ pero que desde el punto de vista del grupo social que las asume son indistinguibles de la ‘realidad objetiva’.

Un ejemplo de que su teoría ontológica permite basar los modelos conceptuales en bases teóricas más sólidas, de tal modo que permite prescribir o proscribir determinadas construcciones, lo defienden en

*Propuesta de un método de validación de esquemas conceptuales
y análisis comparativo de la noción de información
en los métodos de desarrollo de sistemas de información*

To illustrate the problem of classifying phenomena, consider the constructs of 'things' and 'properties of things'. Determining whether a certain phenomenon should be designated as a thing or a property is often difficult. For example, should the color 'red' be modeled as a thing with properties (e.g. hue) or a property of some other thing (e.g., a car)? Some researchers adopt a relativist viewpoint. They argue that users should be allowed to choose which construct to use. Others assert that a phenomenon must be either a thing or a property of a thing -- it cannot be both (Weber 1996). Researchers who adopt this latter stance need to provide guidance to practitioners, therefore, on how to distinguish things from properties of things. (Wand and Weber, 2002; p.369)

Esta teoría les lleva a distinguir entre la ‘estructura profunda’ del sistema de información que expresa el significado del mundo real que se modela y la ‘estructura superficial’ que es la interfaz entre el sistema de información y el entorno organizativo de los usuarios (Butler, 2000) (Truex and Baskerville, 1998). También les proporciona una base para establecer criterios para juzgar la calidad de los datos en los sistemas de información (Wand and Wang, 1996) y de la calidad de los modelos conceptuales en lo que respecta a su validación por los usuarios (Shanks et al., 2003).

Aunque los autores parten de una ontología realista, cabe considerarlos conceptualistas pues siempre presuponen que el modelo conceptual a construir debe expresar los conceptos tal como los perciben los usuarios.

También pueden ser considerados preceptivos pues parten de una teoría ontológica ya elaborada, aunque no son dogmáticos (o racionalistas en el sentido de Leibnitz de obtener las verdades a priori y no a posteriori) dado que presentan las prescripciones de su teoría como un conjunto de proposiciones que se han de validar empíricamente (Bodart et al., 2001; Weber, 1996).

Paradigma Realista Pragmático

Esta posición se puede encontrar formulada de manera explícita en algunos grupos de modelado conceptual: *Following Aristotle, we assume that the universals exist in the individuals (in re) but not independently from them. Therefore any universal, in order to exist, must possess instances, implying that the poetic concept Unicorn does not correspond to a universal.* (Guizzardi et al., 2002)

Pero para ejemplificarla emplearemos la que adoptan, aunque implícitamente, una gran mayoría de ingenieros de software. De hecho, la aspiración de los ingenieros de software es emular a los ingenieros de otras especialidades:

That is why we compare ourselves to aeronautical and electrical and automotive and chemical engineers, and aspire to emulate their enviably well-established repertoires of 'theoretical foundations and practical disciplines'. They too are concerned to make useful physical devices. (Jackson, 1995; p.283)

The practice of every engineering discipline --be it civil, mechanical, chemical, electrical, or software engineering-- involves both science and art. (Booch, 1994; p.21)

Por esta razón: *'We must understand what goes into making 'industrial processes' successful and then apply this knowledge in an appropriate manner to the software industry.'* (Jacobson et al., 1992; p.1) A continuación, los autores ponen como modelo la industria de la construcción para considerarla como una guía de la ingeniería del software.

En terminología de Kuhn (1962) este paradigma representa la 'ciencia normal' o, en este caso, de 'ingeniería normal'. Esta doble característica, de ciencia normal y de ingeniería, hace que los presupuestos ontológicos y epistemológicos sean soslayados, a diferencia de los métodos que utilizan ontologías o los del nominalismo pragmático que, al ser propuestas relativamente nuevas, se ven obligados a explicitar en mayor o menor medida sus asunciones ontológicas y epistemológicas.

Para ejemplificar este paradigma utilizaremos el Unified Modeling Language, UML, probablemente, el método más conocido en la actualidad. Este método, creado en el año 1995, es la fusión de otros tres anteriores, el OOSE de Jacobson et al. (1992), el OMT de Rumbaugh et al. (1991) y el de Booch (1994). La ambición de sus autores es crear un estándar de hecho y evitar así la inevitable torre de Babel de métodos y lenguajes a la que tiende la disciplina cuando surge una nueva tecnología, en este caso la programación orientada a objetos. El ámbito de aplicación del método pretende ser universal, válido para todo tipo de proyectos y en todos los dominios de aplicación, lo cual incluye las aplicaciones intensivas en datos de los sistemas de información y el modelado de empresa. (UML, 2003).

*Propuesta de un método de validación de esquemas conceptuales
y análisis comparativo de la noción de información
en los métodos de desarrollo de sistemas de información*

La pretensión de aplicabilidad universal del método queda en parte justificada por la diferente procedencia de sus autores. El método de Booch era popular para aplicaciones informáticas en ingeniería y era especialmente apto para las fases de diseño y construcción de los sistemas. El OMT, de Rumbaugh, era especialmente apropiado para aplicaciones de sistemas de información de empresa; según confiesan dichos autores (Rumbaugh et al., 1991; p.267), su método es una mejora del Entity-Relationship de Chen (1976), que es el precursor de u gran parte de los modelos de datos semánticos, los modelos conceptuales. El método OOSE, de Jacobson, cuyo origen fue el desarrollo de sistemas informáticos para las telecomunicaciones, proporcionaba una base buena para el análisis de requerimientos al utilizar sus ‘casos de utilización’ (use-cases).

Su posición es claramente realista, por ejemplo, de los cinco principios de modelado que proponen, el tercero es:

The best models are connected to reality.

A physical model of a building that doesn't respond in the same way as do real materials has only limited value. (Booch et al., 1999; p. 9)

No consideran problemático el establecimiento de los conceptos que constituyen el modelo conceptual (o, tal como lo llaman en UML, el diagrama de clases): *For users, most abstractions are not that hard to identify because, typically, they are drawn from the things that users already use to describe their system. ...make sure that each class is crisply defined.* (Booch et al.,1999; p.55)

Los conceptos tienen un significado claro y bien delimitado: *We define an object as a concept, abstraction, or thing with crisp boundaries and meaning for the problem at hand.* (Rumbaugh et al., 1991; p.21)

Si se denominan adecuadamente, el nombre de las clases refleja su significado: *Naming things properly --so that they reflect their semantics-- is often treated lightly by most developers, yet is important in capturing the essence of the abstractions we are describing.* (Booch, 1994; p.163)

Booch en su libro de 1994 dedica un capítulo entero a reflexionar sobre la problemática de la conceptualización, de la categorización. Para identificar las clases y objetos de un dominio, dicho autor presenta brevemente los enfoques clásicos y modernos. El clásico es el que llama

la categorización clásica: *All the entities that have a given property or collection of properties in common form a category. Such properties are necessary and sufficient to define the category.* El autor reconoce que en algunos casos este enfoque produce problemas: *There are some abstractions that have neither clearly bounded properties nor concepts;* y presenta dos enfoques modernos alternativos: el ‘conceptual clustering’ y el ‘prototyping theory’. Sorprendentemente, este último, basado en el trabajo de Lakoff (1987), es una descalificación de la categorización clásica. Sin embargo el autor no desarrolla conclusiones prácticas de dichos enfoques.

Algunos autores califican la posición realista como ‘realismo ingenuo’, aunque cabe no olvidar que este ‘realismo ingenuo’ ha producido el impresionante desarrollo tecnológico de las últimas décadas. Este paradigma no produce problemas relevantes en el desarrollo de aplicaciones de software para control de dispositivos físicos pues en estos sistemas los conceptos y significados, que se basan en magnitudes físicas, se pueden establecer con precisión. Piénsese, por ejemplo, en los ‘conceptos’ utilizados para desarrollar el software de control de una estación meteorológica, un caso que plantea Booch en su libro, como son la temperatura, humedad, presión barométrica, dirección y velocidad del viento.

Paradigma Nominalista Preceptivo: Métodos Formales

Los métodos formales son una continuación natural del realismo pragmático de la ingeniería del software, son una de las maneras de alcanzar el *‘enviably well-established repertoires of theoretical foundations and practical disciplines’* de las ingenierías tradicionales, que aplican las matemáticas. (Jackson, 1995; Saiedian, 1996)

Así lo explicitan en la presentación del método formal Z: *‘Formal methods, based upon elementary mathematics, can be used to produce precise, unambiguous documentation, in which information is structured and presented at an appropriate level of abstraction. ... The mathematical basis is different from that of civil or mechanical engineering, but it has the same purpose: to add precision, to aid understanding, and to reason about properties of a design. Whatever the discipline, the use of mathematics can be expensive, but is our experience that it can actually reduce costs.’* (Woodcock and Davies, 1996; p.2)

Para representar el nominalismo preceptivo elegiremos a la comunidad de métodos formales, la que utiliza la lógica matemática

como base de los lenguajes de especificación formal. En el ámbito de la lógica matemática el término ‘formal’ tiene un significado concreto. Como explica Quine: *...these characterizations are formal, in that they speak only of the typographical constitution of the expressions in question and do not refer to the meanings of those expressions.* (Quine, 1981; p.283)

En este paradigma no se habla de conceptos, clasificaciones o categorizaciones, y mucho menos de ontologías; normalmente emplean la noción de ‘términos’ y ‘términos primitivos’. Cabe considerarlos nominalistas pues reconocen que los términos de la especificación no tienen más significado que el convencional de la notación, no se refieren al mundo real. Para que lo hagan, es necesario establecer la correspondencia de los términos con los objetos del mundo real, y para ello la única manera de hacerlo es dar una definición informal, que puede ser difícil de establecer con claridad, precisión y sin ambigüedad.

Any formal representation uses primitive terms with no inherent formal meaning. In requirements engineering, the meaning of these terms lies in the real world, and the validity of any formal assertion relies on it. ...The only way to establish the meaning of a primitive term is to provide an informal explanation of it. This explanation must be clear and precise; it must be written down; and it must be maintained as an essential part of the requirements documentation. We call such an explanation a ‘designation’. (Zave and Jackson, 1997; p.3)

Aunque, más adelante, los autores reconocen la dificultad de la tarea:

It is well known that pinning down the real-world meaning of a primitive term can be extremely difficult --it can be fraught with subtleties and ambiguities. Everyone with experience in requirements engineering has, in some situation, used a term freely for a long time before noticing that its meaning is unclear. Yet very few requirements methods address this problem by enforcing the presence of a designation for every primitive term.

Lamsweerde coincide con el planteamiento anterior, y explica por qué la definición de los términos primitivos debe ser necesariamente informal: para evitar una regresión infinita.

Formal specifications are meaningless without a precise, informal definition of how to interpret them in the domain considered. A formalization involves terms and predicates which may have different meanings. The specification thus makes sense only if the meaning of each term/predicate is

stated precisely, by mapping function/predicate names to functions/relations on domain objects. This mapping must be precise but necessarily informal (to avoid infinite regression). This fairly obvious principle is often neglected. (Lamsweerde, 2000; p.151)

Quine explica el problema de la regresión infinita:

What makes ontological questions meaningless when taken absolutely is not universality but circularity. A question of the form "What is an F? can be answered only by recourse to a further term: "An F is a G". The answer makes only relative sense: sense relative to the uncritical acceptance of "G". (Quine, 1969; p.53)

Los caracterizamos como nominalistas pues reconocen que los términos no tienen un significado en sí mismos y no recurren a conceptos u ontologías para resolver el problema. Son preceptivos pues asumen el empleo de la lógica matemática como solución a los problemas de especificación formal.

Paradigma Nominalista Pragmático

Algunos de los autores del paradigma anterior también podrían ser considerados nominalistas pragmáticos pues en algunos momentos defienden la idea de abandonar la utilización canónica de la lógica matemática y afirman que se han de desarrollar métodos formales con notaciones ad hoc, propias para cada dominio de problemas (Lamsweerde ,2000) (Zave, 1996).

Posiblemente, el mejor exponente de desarrollo ad hoc de un método formal sea el de la teoría de bases de datos relacionales de Codd, que trata del problema de diseño de bases de datos y se ha convertido en un estándar comercial (Codd, 1970; Codd, 1979; Codd, 1981; Codd, 1982)

Como hemos visto en el capítulo anterior, el modelo de datos relacional tiene una estructura muy próxima a un calculo lógico, lo que lo hace especialmente apto para ser implementado en un ordenador, aunque tiene una serie de limitaciones en cuanto a su expresividad, lo que provocó el desarrollo de modelos de datos que superasen sus limitaciones, los modelos de datos semánticos. Codd siempre ha opinado que la mayor parte de ellos no pasan de ser meras notaciones gráficas carentes de fundamento teórico. Weber (2003a) también reconoce este hecho y por ello, como hemos visto en el apartado del paradigma

conceptualista preceptivo, propone buscar el fundamento teórico de los modelos conceptuales en la ontología, lo que, si observamos la historia de la filosofía, no es necesariamente una garantía de éxito científico.

La discrepancia de Codd, desde su enfoque tecnológico de diseñador de máquinas, esto es, cálculos lógicos, y por lo tanto nominalista, que suscribe el relativismo semántico frente a las posturas conceptualistas, la cita el propio Weber:

When seeking to distinguish between entities and relationships, (Codd, 1990; p.477) also refers to the problem of semantic relativism: "... one person's entity is another person's relationship... If there are 10 people in a room and each is asked for definitions of the terms "entity" and "relationship", 20 different definitions are likely to be supplied for each term"(Weber, 1996; p.138)

La posición de relativismo semántico se puede observar también, aunque es muy minoritaria, en el área de ingeniería de requerimientos en los trabajos de Goguen (Goguen, 2002 ; Luqi and Goguen, 1997; Goguen ,1996; Goguen, 1994). El autor propone una definición de información que es relativista: *An item of information is an interpretation of a configuration of signs for which members of some social group are accountable.* (Goguen, 1997; p.4)

En los trabajos sobre Intercambio Electrónico de Datos (EDI), en lo referente a la necesaria estandarización del vocabulario que permita intercambiar electrónicamente mensajes, también existen posiciones críticas frente a la mayoría: *It is a common misconception that the meaning of words is relatively fixed over time and distance; they may be captured and later referenced in a dictionary. But word meanings are always subject to negotiation, interpretation and subtle adjustment in use.* (Damsgaard and Truex, 2000; p.180)

En el área de desarrollo de sistemas de información este paradigma ha ido aumentando su número de seguidores, así como en los trabajos que se refieren a métodos de investigación de campo en sistemas de información, en donde cabría destacar el de Klein y Myers (1999)

Como paradigma de esta posición utilizaremos los trabajos de Ronald Stamper (Stamper, 1987; Verrijn-Stuart ,2001). Dicho autor es el que presenta su postura de manera más concisa, contundente y explícita. Posiblemente esto sea debido a que ha colaborado durante más de diez

años en el IFIP WG8.1 Task Group FRISCO (acrónimo de Framework of Information Systems Concepts) cuyo objetivo es establecer un marco de conceptos de sistemas de información fundamentados, además de en la informática, en otras disciplinas como las de filosofía, semiótica, ontología y ciencias de la organización (Verrijn-Stuart, 2001). El informe final podría caracterizarse como una posición intermedia entre el conceptualismo y el nominalismo pragmáticos. Sin embargo, Stamper, desde una postura más nominalista, presentó un artículo manifestando su disentimiento con partes fundamentales del informe:

If we define meaning as the relationship established by people in a language community between thing-A (sign) standing for thing-B (object), we see that meanings always depend on the interpreter(s). (Stamper, 1998) p.195

In this new paradigm, every view of the world is associated with some agent, usually a collective agent, a community that has evolved a shared vision of the world. Different cultures, different language groups, different professions, different organizations, even different teams within the same organization will use words with different meanings. Their worlds are indeed intrinsically different. 'Indeed' indeed, because their meanings depend upon deeds not other words.

A system developed on the basis of this position of radical, socially-based subjectivism allows for semantic diversity. Thus it respects one important aspect of the informal human information system.

...Meanings are not the possession of the words themselves, they have to be provided by identifiable agents. Unlike classical logical languages, you cannot introduce names and predicate expressions arbitrarily, to be understood by the analyst and whoever has understood him. Every word in the system has to be linked to a responsible agent (individual, group, or role) and the meaning has to be explicable in terms of action, not just defined using other words. Different agents are entitled to their own interpretations. (Consider what 'customer' means to different people in a company -- the market specialist, the salesman, the accountant, the company lawyer, the production engineer, etc.) Also, actions depend upon other actions and this gives rise to a structure that makes it easy to disambiguate an expression used in many ways. (Stamper et al., 1991; p.75)

*Propuesta de un método de validación de esquemas conceptuales
y análisis comparativo de la noción de información
en los métodos de desarrollo de sistemas de información*

El nominalismo no es una posición bien delimitada, incluye un amplio espectro de posiciones. Por ejemplo, la posición de Stamper, próxima a la del sofista Protágoras (‘el hombre es la medida de todas las cosas’), está bastante alejada de la del sofista Gorgias cuando niega la existencia de la verdad y la posibilidad de comunicarla.

Conclusiones de este capítulo

En el diagrama 12 se presenta un resumen de las conclusiones de nuestro análisis.

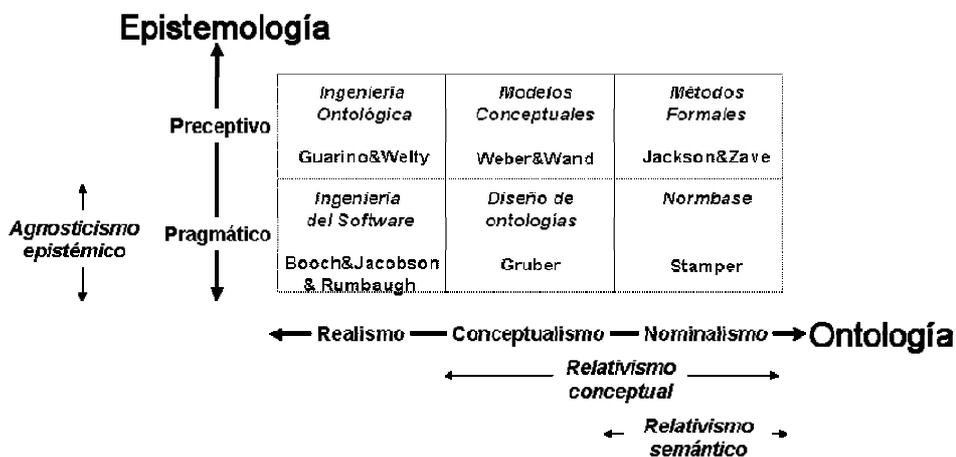


Diagrama 13. Posiciones ontológicas y epistemológicas en Desarrollo de Sistemas de Información.

Nuestro estudio evidencia la grave divergencia de posiciones metateóricas (metafísicas) respecto a la noción de información –noción fundamental en el área de sistemas de información— que mantienen los investigadores de la muestra que hemos presentado.

Cuando se propone una teoría, la adopción de una posición metafísica es inevitable. Toda teoría está basada en alguna, incluidas las teorías que ignoran o rechazan la metafísica. Creemos que una ventaja del estudio filosófico que hemos presentado es que al explicitar los pre-juicios de los distintos autores podemos contemplar las creencias como conjeturas. La diferencia entre una creencia y una conjetura es sólo de actitud del que la propone.

La discusión de creencias básicas siempre es causa de encendidos debates. Esto se pudo comprobar al presentar este estudio en la 14th European Conference on Information Systems, en la que, a pesar de que en el artículo se presentan las distintas posiciones de la manera más imparcial posible, se abrió una acalorada discusión entre algunos de los asistentes defendiendo o atacando sus respectivas posiciones.

Las conjeturas son creencias que se presentan explícitamente y que reclaman una prueba de utilidad empírica. Cuando se inició este estudio nuestra posición estaba sesgada hacia un nominalismo pragmático moderado. De hecho lo que lo inició fue el presenciar, con un cierto horror, la aparición de algunas de las propuestas que se han presentado en los últimos años, como las propuestas puramente metafísicas. Después de este estudio el 'horror' se ha atemperado, no se debe descartar precipitadamente ninguna posición, todas pueden tener un ámbito en el que manifiesten su utilidad.

Por ejemplo, la postura platónica se debe tener presente en muchas instituciones sociales, como el juego del ajedrez o el dinero. Estos dos ejemplos se aproximan al ideal platónico, la realidad es producto de la idea. O en las ciencias del diseño, en las que el sistema o artefacto es producto de las ideas preconcebidas. En estos casos los conceptos, o términos generales, pueden alcanzar una gran precisión. Otro ejemplo son las matemáticas actuales, que renunciaron a la exactitud en aras de la precisión, pero cuya utilidad es indiscutible tanto en la ciencia como en la tecnología. Por otra parte, las posturas aristotélicas encuentran su mejor justificación en la tabla periódica de los elementos químicos, una taxonomía de gran poder explicativo y predictivo.

Las posiciones conceptualistas también tienen su lugar. La física es una prueba de que no es imposible conseguir precisión y convergencia en el significado de los términos compartidos por una comunidad en algún dominio de problemas, características esenciales para la creación de ciencia acumulativa y rigurosa.

Por otra parte, las disciplinas jóvenes nos muestran que sus términos generales tienen significados difusos e inestables, todavía no tienen la categoría de conceptos. Sin embargo, la ciencia avanza mediante el refinamiento continuo del significado de sus términos generales. Por ejemplo, la masa y el peso eran sinónimos antes de Newton y posteriormente Einstein mostró que el concepto de masa tiene también

varios significados. Un ejemplo de término difuso es, como hemos discutido, el de información en nuestro campo.

Sin embargo, en el campo de la construcción de esquemas conceptuales concretos para empresas y organizaciones, pensamos que la postura que más se ajusta a la realidad, y la más prudente, es a partir del nominalismo pragmático intentar llegar a conseguir un esquema conceptual único mediante la negociación del significado de los términos; esto es, aproximarse a unas posiciones cuyo ideal refleja la postura del conceptualismo pragmático.

Limitaciones y contribuciones del esquema propuesto

La primera limitación es que el mapa, o esquema, propuesto, no es una clasificación. Creemos que no existen —o no hemos sabido encontrar— suficientes ‘regularidades’ para categorizar a los distintos autores. Lo más que hemos pretendido es fijar determinados ‘aires de familia’.

Es difícil, si no imposible, categorizar a algunos autores. Un ejemplo notable es el de Wittgenstein, que efectuó un cambio radical en su postura filosófica desde su *Tractatus* hasta su *Investigaciones Filosóficas*, de tal modo que para referirse a sus concepciones filosóficas se recurre frecuentemente a dos filósofos, Wittgenstein1 y Wittgenstein2, personificados en un mismo individuo.

Otro ejemplo, esta vez de un filósofo que a lo largo de su carrera no cambió sustancialmente de posición es Quine, al que, después de haber leído buena parte de su obra, no nos atreveríamos a encasillar. Pero la dificultad de clasificarlo no es sólo nuestra. En la entrada ‘Nominalism’ de la *Encyclopaedia Britannica* (2003), Quine es considerado un destacado exponente del nominalismo moderno —aunque el propio autor rechaza explícitamente ser nominalista (Quine, 1960; p.243)— y es también considerado como platónico en la misma enciclopedia, en el apartado de ‘Contemporary Materialism’.

En el caso de posicionamiento de autores del área de sistemas de información, la dificultad es mayor pues normalmente no son tan explícitos como los filósofos —cuyo trabajo consiste precisamente en discutir estas nociones.

El propósito de este trabajo no ha sido reflejar la postura de unos determinados autores, el objetivo ha sido utilizar la posición que mantuvieron en determinados artículos en un momento del tiempo como ejemplo, como paradigma, de la diversidad de concepciones de la información como representación. Esta diversidad muchas veces no explicitada dificulta la comprensión de los trabajos.

El ser consciente de las distintas concepciones permite entender mejor a los autores. Pero ésto no presupone que puedan haber cambiado. Además, los ejemplos no quieren reflejar las creencias del autor, sólo su posición en un proyecto de investigación ante la clase de problemas a que se enfrentan. Por ejemplo, Gruber dice explícitamente que frente a las dudas, adopta su postura como una hipótesis. Lo mismo hacen Weber y Wand, que, aunque parten de la ontología de Mario Bunge, no afirman que sea necesariamente cierta, intentan validarla mediante experimentos empíricos. Por ejemplo en (Weber, 1996) se hace un experimento no para comprobar si las entidades y los atributos son necesariamente distintos, sino para comprobar que la memoria humana tiende a distinguir las entidades y los atributos como dos constructos diferentes.

A raíz del paradigma realista, considerado algunas veces como realismo ingenuo, cabe puntualizar que cada ámbito de problemas, desde un punto de vista práctico, puede tener su paradigma adecuado. La ingenuidad no está en el paradigma sino en la aplicación del mismo a una clase de fenómenos para la que no es adecuado. Y puede ser tan ingenuo el realista al aplicarlo a realidades sociales como el subjetivista al aplicarlo al ámbito de las ciencias físicas o matemáticas.

A pesar de todas las limitaciones de este tipo de esquemas pensamos que son útiles si nos atenemos al propósito que tienen, resaltar el importante grado de divergencia que existe en las concepciones sobre las características de la información. Normalmente, tanto en la investigación como en las aplicaciones prácticas de desarrollo de sistemas de información, se tiende a dar por supuesto, en lugar de hacer explícita, la noción de información utilizada. Detallar los presupuestos ontológicos y epistemológicos de la noción de información no es únicamente un ejercicio académico alejado de la utilidad práctica, sino que es un prerequisite para entender todas las consecuencias de cualquier método de desarrollo de sistemas de información. Además, el debate es una

condición necesaria si queremos conseguir la convergencia de significados en la comunidad de investigadores y profesionales de sistemas de información para, así, construir ciencia acumulativa. En ciencia, las taxonomías normalmente preceden a las teorías. Este estudio ha pretendido dar un primer paso en la construcción de una taxonomía robusta de la noción de información como representación en el área de desarrollo de sistemas de información.

Pero la noción de información en su acepción de representación, aunque es primordial en el desarrollo de sistemas de información, puesto que uno de los principales usos de los sistemas de información es ayudar a sus usuarios a referirse a conjuntos de objetos, propiedades y sucesos — la acción humana de referir es inevitable—, y aunque los modelos conceptuales de datos, o las ontologías, se restrinjan a esta noción, existen otras nociones de información, algunas de las cuales son relevantes para la especificación y construcción de sistemas de información. En el próximo capítulo se presenta una relación de distintas nociones existentes.

Capítulo 5: Distintas acepciones del término información

It is useful to have examined the number of meanings of a term both for clearness' sake (for a man is more likely to know what it is he asserts, if it has been made clear to him how many meanings it may have), and also with a view to ensuring that our reasonings shall be in accordance with the actual facts and not addressed merely to the term used. For as long as it is not clear in how many senses a term is used, it is possible that the answerer and the questioner are not directing their minds upon the same thing...

...for if we know the number of meanings of a term, we shall certainly never be misled by false reasoning... (Aristotle, 350d B.C; p.180.)

Hace más de tres décadas, Ashenhurst (1973) se lamentaba de que la ausencia de una buena definición del término ‘información’ imposibilitaba un tratamiento analítico definitivo tal como ocurre en física con el término ‘energía’. Dos décadas después, Denning (1995) seguía preguntándose qué es información. Señalando que ‘es una de las preguntas fundamentales de nuestro campo’ y ‘para aquellos que quieren basar nuestra disciplina en unas bases científicas rigurosas, necesaria de responder’.

Hoy, diez años más tarde, la pregunta sigue abierta, y la utilización del término igual de confusa. La situación es parecida a la del término ‘energía’ en la física del siglo XIX aunque el problema es el inverso. En aquel siglo existían distintos tipos de energía, como la calorífica, la mecánica, la eléctrica o la magnética. Se produjo un proceso de convergencia y se logró reducirlas a un único concepto al comprobar que se podía transformar cualquiera de ellas en las otras.

*Propuesta de un método de validación de esquemas conceptuales
y análisis comparativo de la noción de información
en los métodos de desarrollo de sistemas de información*

Con la palabra información la situación actual es la inversa, en lugar de haber distintos términos que se refieren a lo mismo, hay un único término, información, que se refiere a fenómenos distintos. Shannon reconoce que la palabra información tiene muchos significados diferentes y considera dudoso que un único concepto pueda ser empleado satisfactoriamente en todos los posibles campos de aplicación.

The word "information" has been given many different meanings by various writers in the general field of information theory. It is likely that at least a number of these will prove sufficiently useful in certain applications to deserve further study and permanent recognition. It is hardly to be expected that a single concept of information would satisfactorily account for the numerous possible applications of this general field. (Shannon, 1950; p.180)

Tarski plantea el mismo problema con el término ‘verdad’. Esta noción y la de ‘información’ para algunos autores tienen una estrecha relación. Afirman, por ejemplo, que la oración ‘hay 100 libros en el almacén’ transporta información si es verdadera. Un problema añadido es que, aunque la noción usual de ‘verdad’ de una oración es su correspondencia con los hechos, o como más sutilmente la define Aristóteles ‘*To say of what is that it is not, or of what is not that it is, is false, while to say of what is that it is, and of what is not that it is not, is true.*’ (Aristotle, 350b B.C.; p.1597), estamos lejos del consenso. Véase, por ejemplo en (Blackburn and Simmons, 1999), una muestra de las distintas nociones de ‘verdad’ que se proponen en filosofía.

Aunque no nos sirva para fijar la noción de información, Tarski, en su intento de demarcar la noción de ‘verdad’ nos sugiere el método que emplearemos en este capítulo.

What is the right conception of truth? I must confess I do not understand what is at stake in such disputes; for the problem itself is so vague that no definite solution is possible. In fact, it seems to me that the sense in which the phrase 'the right conception' is used has never been made clear. In most cases one gets the impression that the phrase is used in an almost mystical sense based upon the belief that every word has only one 'real' meaning (a kind of Platonic or Aristotelian idea), and that all the competing conceptions really attempt to catch hold of this one meaning; since, however, they contradict each other, only one attempt can be successful, and hence only one conception is the 'right' one.

Disputes on this type are by no means restricted to the notion of truth. They occur in all domains where –instead of an exact, scientific terminology— common language with its vagueness and ambiguity is used; and they are always meaningless, and therefore in vain.

It seems to me obvious that the only rational approach to such problems would be the following: We should reconcile ourselves with the fact that we are confronted, not with one concept, but with several different concepts which are denoted by one word; we should try to make these concepts as clear as possible (by means of definition, or an axiomatic procedure, or in some other way); to avoid further confusions, we should agree to use different terms for different concepts; and then we may proceed to a quiet and systematic study of all concepts involved, which will exhibit their main properties and mutual relations. (Tarski, 1944; p.133)

En este capítulo nos proponemos, en primer lugar, caracterizar, sin pretensión de exhaustividad, algunas de las acepciones del término información. Siguiendo a Tarski, utilizaremos diferentes términos para denotar distintos fenómenos. Para ello, presentamos brevemente las distintas acepciones del término, la información como soporte, señal, regularidad, signo, símbolo, sentido, significación, sentimiento, síntoma, intención comunicativa, convención, acto de habla y acto social. Cabe aclarar que estos trece términos los empleamos como meras etiquetas. Sería más correcto haber empleado un único término, información, con trece subíndices, información₁, información₂,... pero esta opción, aunque más rigurosa, requiere realizar un esfuerzo de memoria mayor, la memoria humana es fundamentalmente semántica.

El inconveniente de la terminología que utilizaremos es que puede llevar a confusión. Si acudimos al diccionario para discriminar entre las palabras ‘significado’, ‘sentido’ y ‘significación’ nos encontramos con definiciones circulares. Este problema se agrava por el hecho de que distintos filósofos utilizan los términos con significados diferentes e, incluso, opuestos. Por ejemplo, Frege (1892) distingue, como veremos más adelante, entre ‘sense’ y ‘reference’ (este último término nosotros lo etiquetamos como ‘información como símbolo’), y también distingue entre ‘sense’ y ‘idea’. El problema es que utiliza el término ‘sense’ en el sentido de idea platónica, de significado objetivo, y emplea ‘idea’ en el sentido de significado subjetivo. *‘The same sense is not always connected, even in the same man, with the same idea. The idea is subjective: one man's idea is not that of*

another. ...This constitutes an essential distinction between the idea and the sign's sense which may be the common property of many and therefore is not a part or a mode of the individual mind. (Ibid., p.39). Esto es, utiliza los dos términos con las acepciones opuestas a las de Platón y Kant. En este escrito utilizaremos 'significado' como un término genérico, neutro, que incluye a 'sentido', al que asignamos la acepción subjetiva, esto es, a lo que Frege llama 'idea', y utilizamos 'significación' denotando el significado 'objetivo', o para ser más precisos, el significado público, intersubjetivo.

No utilizaremos la tradicional clasificación de los diferentes aspectos de la información: sintáctica, semántica y pragmática. Ni la más ampliada de de información física, empírica, sintáctica, semántica, pragmática y social, por considerarlas insuficientes. (Stamper, 1991 ; Stamper and Liu, 1994 ; Stamper et al., 1997 ; Falkenberg and others ,1998)

Si, como dice Wittgenstein, el significado de las palabras está en su uso, veamos cómo es utilizado por distintos autores en distintos ámbitos. El propósito es caracterizar mínimamente los distintos fenómenos a los que hacen referencia.

Información como soporte

En (Zhirnov and Herr, 2001) podemos leer:

Each atom receives information, and then processes and transfers it to other atoms to yield crystal lattices with perfectly defined symmetries.

En este ejemplo, los autores utilizan información con el significado de energía. A esta acepción Shannon le llama 'carrier', esto es, portador o soporte. Y manifiesta que la información se transporta por medio de la materia o de la energía, que actúan de soportes. En el caso de enviar un libro, la información tiene un soporte material; en el caso de una emisión radiofónica el soporte es energía electromagnética y, para la voz humana, el sonido. La información, para Shannon, no es el soporte sino las marcas o la modulación de las señales que se hacen en el soporte (Shannon, 1962). A esta acepción la llamaremos señal.

Información como señal

El problema que plantea Shannon en su teoría matemática de la información es el de la comunicación de un mensaje entre un emisor y

un receptor suponiendo ambos conocen el conjunto de posibles mensajes a comunicarse.

Para expresarlo de una manera simple, que no desvirtúa la noción de información que presenta la teoría, lo presentaremos, tal como lo expone Feynman (1996), en el caso particular en el que todos los mensajes del conjunto son igualmente probables. En el caso de que tanto el emisor como el receptor hayan numerado los mensajes utilizando el mismo criterio, bastará con transmitir el número de orden del mensaje. Si tenemos N mensajes y expresamos este número en binario, el número de dígitos binarios (bits) necesarios para numerar los N mensajes será un I tal que $2^I = N$, por lo que $I = \log_2 N$

Shannon llamó información a este I . Esto es, información es el mínimo número de bits necesarios para codificar un mensaje (no exactamente el mensaje sino el número de orden del mensaje).

De esta manera ya podemos medir la información, la señal, al igual que medimos las magnitudes físicas como masa o energía. La unidad de medida es el dígito binario, el bit. Nombre que Shannon, a sugerencia de J.W. Tukey, estableció como abreviación de 'binary integer' en su famoso artículo "*A Mathematical Theory of Communication*" (Shannon, 1948).

Shannon generalizó este concepto para calcular el número de bits necesarios para codificar un mensaje teniendo en cuenta la frecuencia de utilización de los signos empleados. Así, si tenemos un alfabeto para escribir mensajes en el que la probabilidad de aparición de cada signo es diferente, Shannon define la información contenida en cada símbolo como $-\log_2 p_n$, en donde p_n es la frecuencia estadística de uso de dicho símbolo.

Esto quiere decir que si queremos minimizar el número de bits utilizado deberemos codificar los símbolos más frecuentes con menos bits que los más frecuentes. Extrapolando imaginativamente este hecho se puede decir que los sucesos más improbables contienen más información que los más probables. Lo que quiere decir en sentido estricto, en términos de Shannon, es que, si queremos optimizar la transmisión de bits, codificaremos los sucesos más probables con menos bits, esto es, con menos 'información'. O sea, con menos marcas, 'señales'.

El bit también da una medida de la incertidumbre. Para simplificar la explicación, retomemos el ejemplo inicial de mensajes igualmente probables. Si sólo llega el primer bit de un mensaje, no se sabe cuál es el mensaje, pero se ha reducido la incertidumbre a la mitad, pues el conjunto de mensajes que tienen dicho bit con el valor recibido es la mitad. Cuantos más bits se conocen, cuanta más ‘información’ se tiene, menor es la incertidumbre, esto es, menor es el conjunto de mensajes posibles entre los que está el mensaje enviado.

Shannon aclaró numerosas veces que su concepto de información no se refiere a los significados de los mensajes sino al número de bits necesarios para transmitirlos.

"The fundamental problem of communication is that of reproducing at one point either exactly or approximately a message selected at another point. Frequently the messages have meaning; that is they refer to or are correlated according to some system with certain physical or conceptual entities. These semantic aspects of communication are irrelevant to the engineering problem. The significant aspect is that the actual message is one selected from a set of possible messages" (Shannon and Weaver, 1963)

Pero el concepto de información de Shannon no sólo no es semántico, ni siquiera expresa el contenido en bits de los mensajes. Únicamente es una medida de la libertad de elección al elegir un mensaje de un conjunto prefijado. Por ejemplo, como señalan Li y Vitányi (1993; p.61), supongamos que todo nuestro conjunto de mensajes está formado por sólo dos posibles mensajes a enviar: el contenido de un libro y el texto “hola”. En este caso, la información necesaria para codificar cada mensaje, dado que el conjunto total está compuesto de sólo dos mensajes, es exactamente un bit, el bit a cero representará el libro, el bit a uno, el texto “hola”.

La información como regularidad (pattern)

En la década de los años sesenta del siglo pasado, Chaitin, Solomonoff y Kolmogorov inventaron de forma independiente una nueva teoría matemática, la teoría de la información algorítmica, que presenta una noción de información diferente a la de Shannon. Entre los varios nombres con que se conoce esta teoría está el de teoría de la ‘complejidad descriptiva’(Li and Vitányi, 1993; p.vi).

El problema planteado en esta teoría es calcular el mínimo número de bits necesarios para describir un objeto: *The algorithmic information content of an individual object X is defined to be the size of the smallest program to calculate X.* (Chaitin, 1982)

En esta teoría se supone que tanto el emisor como el receptor de los mensajes, en lugar de disponer de todo el conjunto de mensajes posibles, como en el caso de Shannon, únicamente tienen capacidad algorítmica, esto es, disponen de un lenguaje programación. Ahora el mensaje puede contener tanto el ordinal del mensaje —del modelo de Shannon expresado en sistema binario— como cualquier texto (por ejemplo un libro cuyos caracteres han sido codificados en ASCII, también en binario). En esta situación, habrá mensajes en los que la configuración de bits presentará regularidades. Por ejemplo, en el caso de Shannon de N mensajes que requieren I bits para codificarlos ($I = \log_2 N$) habrá un mensaje que tendrá la configuración

010101....01 ($I/2$ veces)

si se constata que es una repetición de la configuración '01', será suficiente con que se envíe como mensaje

repite '01' $I/2$ veces

que, codificando los caracteres del algoritmo en ASCII, tendrá unas pocas decenas de bits. El receptor del mensaje, que también dispone del mismo lenguaje de programación podrá reconstruir (en el argot informático diríamos 'descomprimir') el mensaje original ejecutando el algoritmo. En el caso de que I sea un número grande se consigue un gran ahorro de bits, la compresión es grande.

Algunos mensajes tendrán pequeñas irregularidades, por ejemplo, el anterior mensaje pero con el sexto bit puesto a 0 en lugar de 1. En este caso, bastará con enviar el algoritmo anterior añadiendo la instrucción 'pon el sexto bit a 0'. El mensaje será algo más largo pero la compresión todavía será considerable. Si continúan aumentando las irregularidades, se tendrán que añadir instrucciones hasta que la ganancia en bits sea nula. Estos textos son incompresibles.

Solomonoff, cuyo interés se concentra en el problema de la inducción científica, en la construcción de teorías y la predicción científica, representa las observaciones científicas como un conjunto de datos que

se pueden representar mediante una cadena de dígitos binarios, de bits. La explicación científica, el descubrimiento de una teoría, consistirá en encontrar un algoritmo que sea capaz de generar dicha serie de bits y, si la teoría tiene carácter predictivo, a extenderlos a futuras observaciones (Chaitin, 1999). Piénsese, por ejemplo, en el descubrimiento de Kepler, que redujo a tres fórmulas los miles de observaciones astronómicas de las posiciones de los planetas que contenían las tablas de Tycho Brahe.

Según este modelo matemático, entre dos teorías competidoras con la misma capacidad de predicción, será mejor la que tenga el algoritmo más pequeño, el que requiera menos bits para ser expresado. Esto recuerda al criterio científico expresado por la clásica navaja de Ockham. (Chaitin, 1975)

Desde este modelo matemático es posible decidir entre dos teorías competidoras cuál es la mejor, o más exactamente, permite definir con precisión qué significa mejor, estableciendo el principio de mínimo tamaño de la descripción: *'The best theory to explain a set of data is the one which minimizes the sum of: the length, in bits, of the description of the theory; and the length, in bits, of data when encoded with the help of the theory.'* (Li and Vitányi, 1993; p.308)

Chaitin reduce la comprensión a la compresión: sólo podemos comprender aquello cuyas observaciones son comprimibles. Un mundo completamente aleatorio, caótico, sería incompresible y por ello incomprensible e impredecible.

The smaller the program, the more comprehensive the theory and the greater the degree of understanding. Observations that are random cannot be reproduced by a small program and therefore cannot be explained by a theory. In addition the future behavior of a random system cannot be predicted. For random data the most compact way for the scientist to communicate his observations is for him to publish them in their entirety. (Chaitin, 1975)

Las ciencias naturales buscan regularidades, patrones de repetición, que se llaman invariantes, leyes de conservación o principios generales. En un mundo donde no percibiéramos similitudes, regularidades, entre objetos o entre fenómenos sería imposible la formación de conceptos o términos generales y por tanto de teorías.

Mientras que en las ciencias naturales el positivismo, con esta actitud nomotética, no produce demasiados problemas, en las ciencias sociales

es más difícil percibir regularidades, lo que lleva a los métodos interpretativistas a la actitud ideográfica, la que considera cada fenómeno único e irrepetible. Esto se puede apreciar en el siguiente fragmento: *'The crux of our argument concerning inconsistency is that the very quest for consistent, generalizable results leads us directly and inevitably to inconsistent results, because the nature of the phenomenon is so varied and multifaceted that it resists the reduction and categorization sought in the name of this quest.'* (Gopal and Prasad, 2000; p.539)

Este argumento, que recuerda al razonamiento sofista, es rebatido por Ron Weber con un razonamiento, que también rememora el de Aristóteles y Platón:

First, I have never been enamored with extreme postmodernism. In my view, ultimately extreme postmodernism leads to nihilism. If there are no stable patterns of phenomena in the world and there are no ways of describing phenomena that enable at least some groups of humans to reach a shared understanding about the phenomena, what is the point of doing research? Moreover, what is the point to having journals that publish research papers? For extreme postmodernists to even seek publication of their ideas somehow seems hypocritical. As a scholar, why should I be interested in some phenomenon that someone claims is unique? (Weber, 2003b; p.vi)

La tensión entre la actitud nomotética y la ideográfica que se produce en los métodos interpretativistas de investigación se muestra claramente en el artículo que fue premiado como el mejor del año de la revista MIS Quarterly, en el que se proponen una serie de principios para los métodos de investigación interpretativistas, uno de los cuales es:

The principle of abstraction and generalization: The two previous principles emphasize those features unique to the particular situation under study. Whereas it is true that interpretive research values the documentation of unique circumstances and is highly suspicious of any claim that human affairs are governed by natural laws that are culturally independent, this is not the whole story. One outcome of the extensive debates in philosophy is that there is a philosophical basis for abstraction and generalization in interpretive field studies. (Klein and Myers, 1999; p.75)

A nivel de lenguaje, que es un fenómeno estrictamente social, es evidente la necesidad de que existan unas ciertas regularidades o pautas de uso. Para que las palabras sirvan para comunicarse es necesario que en la comunidad que las emplea haya una asociación regular entre el objeto

designado o el significado pretendido y el signo empleado para designarlo.

La información como signo

Mediante señales, ya sean puntos y rallas, ceros y unos, marcas de tinta, o fonemas, podemos transmitir signos, letras, palabras, combinaciones de palabras, expresiones; esto es, elementos de un lenguaje considerado a nivel sintáctico, de un lenguaje dotado de unas reglas que establecen las expresiones, las combinaciones de signos, que se consideran bien formadas. En las lenguas naturales este componente sintáctico, formal, está estrechamente entrelazado con el componente semántico y pragmático, sin embargo en los lenguajes artificiales es posible separar nítidamente los aspectos de construcción mecánica de expresiones de los aspectos relacionados con el significado de las expresiones construidas. Emplearemos el término ‘construcción mecánica’ en el sentido de construcción desinterpretada, sin utilizar significados. Actualmente disponemos de máquinas, los ordenadores, que nos permiten materializar este ideal.

Hay propuestas que tratan de reducir la semántica a la sintaxis, o, dicho de dicho con otras palabras, que intentan reducir los significados a un tratamiento mecánico. Por ejemplo, las investigaciones que intentan conseguir la integración automática de esquemas conceptuales (véase un *survey* sobre esta área de investigación en (Rahm and Berstein, 2001). Otro ejemplo son las iniciativas para conseguir la traducción automática de un idioma a otro. A este respecto se produjo un debate en 1990. Uno de cuyos máximos exponentes fue el filósofo Searle, que en la revista *Scientific American* defendió la imposibilidad de la reducción de la semántica a la sintaxis, al tratamiento mecánico, mediante el siguiente argumento:

Axiom 1: Computer programs are formal (syntactic).

Axiom 2: Human minds have mental contents (semantics).

Axiom 3: Syntax by itself is neither constitutive of nor sufficient for semantics.

Conclusion 1: Programs are neither constitutive of nor sufficient for minds.

To specify a language, one needs to specify at least a syntax and a semantics for that language. A syntax for a language lists and categorizes the words and other symbols (e.g., punctuation marks) of that language (a lexicon), and specifies rules that determine the set of permissible combinations of those words (a grammar).

It does not say what the words and combination of words mean. (For example, a syntax might specify that “blorg,” “et,” and “dring” are words in a certain language, that “blorg” is a name, “et” a predicate, and “dring” a general noun, and that “blorg et” is a permissible combination of these words but “dring blorg” is not.) A semantics would associate an interpretation with these symbols. (For example, a semantics for the language just described might specify that “blorg” refers to Clinton, “et” means “is male,” and that “dring” refers to the class of dogs.) To know what a language means, you have to know not only its lexicon and grammar (syntax), you also have to know its semantics. The notions of truth, reference, and meaning are all semantic notions. (Searle, 1990)

En el mismo número de dicha revista, Paul y Patricia Churchland (Churchland and Churchland, 1990) refutaron convincentemente el axioma 3 y por tanto desmontaron la obligatoriedad de la conclusión. Podemos reformular el argumento de Searle para mostrar la posición que asumimos en este trabajo:

Axiom 3: In the present state of technology, syntax by itself is neither constitutive of nor sufficient for semantics.

Es importante mantener esta acepción pues cuando se afirma que los ordenadores, los programas informáticos, procesan información, lo que se está diciendo es que, al menos en la actualidad, procesan signos de manera mecánica, desinterpretada.

Hasta aquí hemos visto algunas acepciones físicas y matemáticas del término. Ahora nos quedan las relacionadas con el significado como fenómeno humano. Para seguir un hilo argumental emplearemos el artículo de Frege (1892) en el que señala, aunque no desarrolla, cinco acepciones de significado. Significado como referencia, como sentido (significado subjetivo), como significación (significado intersubjetivo o público) y como sentimiento. La quinta que señala, las órdenes, la desglosaremos en otras componentes.

La información como símbolo (referencia)

Buena parte de las palabras que utilizamos sirven para referirnos a cosas, a objetos de la realidad extralingüística. Por ejemplo, ‘Barcelona’ es un término particular que *designa* una ciudad concreta; ‘persona’ es un término genérico, no designa un objeto concreto sino a un conjunto de ellos, diremos que *denota* al conjunto de animales racionales (Quine, 1995).

Un símbolo es un signo que tiene asociado una entidad observable en el mundo. El significado de un símbolo es aquello que representa, aquello a lo que se refiere, su referencia. Cuando afirmamos que un enunciado es verdadero, por ejemplo ‘el gato está sobre la alfombra’, es cuando entendemos las referencias de ‘gato’ y de ‘alfombra’, el significado de ‘estar sobre’ y además comprobamos que las referencias se corresponden con los hechos, con la realidad del mundo. Esta es la acepción que ya hemos estudiado en el capítulo anterior.

Pero con ésta no se acaban las acepciones del término significado. Como Kant mostró, hay un tipo de juicios que no son sintéticos, los analíticos, los cuales son ciertos o falsos únicamente en virtud del significado de las palabras, no de su referencia. Esto ocurre, por ejemplo, con la declaración ‘los solterones no están casados’. Aunque el término ‘solterón’ se refiere a un conjunto de personas, y el ‘no casado’ a otro, no hace falta comprobar empíricamente que todas y cada una de las personas llamadas solterón no están casadas. Sencillamente, si afirmásemos que existe un solterón casado, sería una contradicción entre términos, al menos con el significado usual de solterón. Las verdades analíticas son verdades necesarias (no contingentes) en virtud únicamente de su significado. A continuación mostramos la diferencia que hay entre ‘referencia’ (la información como símbolo) y significado, distinguiendo posteriormente dos de las acepciones de significado: la significación y el sentido.

La información como significación (significado objetivo)

Frege muestra que identificar el significado con la referencia no es correcto. Si el significado de un símbolo fuese igual a su referencia, no podría ocurrir que dos expresiones con distinto significado tuvieran la misma referencia. Una de las pruebas que presenta es la de “el lucero del

alba” y “el lucero vespertino”. Las dos expresiones tienen la misma referencia, el planeta Venus, por lo tanto deberían tener el mismo significado. Pero esto no es así pues en virtud del significado de las dos expresiones no podemos saber que se refieren al mismo astro. Para saberlo es necesario hacer una observación astronómica, comprobar empíricamente que el astro que brilla más al amanecer es el mismo objeto que destaca por su luminosidad al anochecer.

Obsérvese la diferencia de la proposición ‘el lucero del alba es el lucero vespertino’ con ‘el lucero del alba es el astro más brillante del amanecer’. En esta última no hace falta ninguna observación empírica para saber que se refieren al mismo objeto. Simplemente, en virtud del significado de las palabras sabemos que tienen la misma referencia. Es suficiente con consultar el diccionario para ver que ‘alba’ es sinónimo de ‘amanecer’, y ‘lucero’ quiere decir ‘astro más brillante’. En términos de Kant, la afirmación de que ‘el lucero del alba es el lucero de la tarde’ es un juicio sintético, mientras que la otra es un juicio analítico.

Además de que, como hemos visto, dos significados distintos pueden tener la misma referencia, también puede haber casos en que una misma significación tenga distintas referencias, por ejemplo, ‘tengo 100 cabezas de ganado’. Aunque si nos aferramos al sentido como referencia, siempre podremos afirmar que cuando cambia la referencia, cambia el significado. En este caso distinguiremos entre el sentido literal y el metafórico.

Putnam (1981) elabora un ejemplo sofisticado de igual significado pero distinta referencia. Argumento que no reproducimos aquí por ser muy largo. Un ejemplo trivial es el enunciado ‘Yo soy alto’. El significado es único, sin embargo la referencia cambia en función de quien lo diga. Esto ocurre con todos los términos deícticos, como por ejemplo los pronombres (Yule, 1996).

Una vez mostrada la diferencia entre referencia y significado, Quine (1953; p.130) propone la separación de la semántica en dos teorías distintas, la teoría del significado y la teoría de la referencia. La primera estudiaría los fenómenos de la sinonimia (o igualdad de significado), significación (o la posesión de significado) y analiticidad (o verdad debido al significado). La segunda se ocuparía de los fenómenos de la designación (o dar nombre a un objeto), de la verdad (lo que hace cierta

una proposición no analítica), de la denotación (o verdad de) y de la extensión.

Pero cuando Frege estudia la diferencia entre referencia y significado, se está refiriendo al significado objetivo, a la significación. A continuación, Frege discrimina entre significación y sentido (significado subjetivo).

La información como sentido (significado subjetivo)

Frege observa que la significación que tienen las palabras puede no coincidir con lo que signifiquen para una persona concreta. Llama 'idea' al significado subjetivo. Para no confundirlo con las ideas platónicas (significados hipostasiados) llamaremos 'sentido' a lo que Frege llama 'idea', al significado subjetivo, al acto de entender, a lo que pertenece al ámbito de cada consciencia humana particular. Frege observa que 'la misma significación no siempre está asociada, incluso en una misma persona, con el mismo sentido. La significación es propiedad común de muchas personas, el sentido de una sola persona.' (Frege, 1892)

El filósofo Frank Jackson argumenta la irreductibilidad de la significación al sentido, a los significados subjetivos. Tal como lo expone Chalmers al estudiar el fenómeno de la consciencia humana:

The hard problem (of consciousness), in contrast, is the question of how physical processes in the brain give rise to subjective experience. This puzzle involves the inner aspect of thought and perception: the way things feel for the subject. When we see, for example, we experience visual sensations, such as that of vivid blue. Or think of the ineffable sound of a distant oboe, the agony of an intense pain, the sparkle of happiness or the meditative quality of a moment lost in thought. All are part of what I am calling consciousness. It is these phenomena that pose the real mystery of the mind. To illustrate the distinction, consider a thought experiment devised by the Australian philosopher Frank Jackson. Suppose that Mary, a neuroscientist in the 23rd century, is the world's leading expert on the brain processes responsible for color vision. But Mary has lived her whole life in a black-and-white room and has never seen any other colors. She knows everything there is to know about physical processes in the brain - its biology, structure and function. This understanding enables her to grasp everything there is to know about the easy problems: how the brain discriminates stimuli, integrates information and

produces verbal reports. From her knowledge of color vision, she knows the way color names correspond with wavelengths on the light spectrum. But there is still something crucial about color vision that Mary does not know: what it is like to experience a color such as red. It follows that there are facts about conscious experience that cannot be deduced from physical facts about the functioning of the brain. (Chalmers, 1995)

Una vez establecida la diferencia entre ‘sense’ y ‘idea’, o utilizando los términos que hemos propuesto, entre significación y sentido, Frege se pregunta qué tipo de existencia tiene la significación, pues no es subjetiva como el sentido, no pertenece al mundo interno, no necesita un sujeto que la porte en su consciencia, pero tampoco es objetiva como los objetos que referencia, no pertenece al mundo externo de los objetos materiales. Y postula un tercer ámbito, un tercer mundo, que no es ni el subjetivo, el de los sujetos, ni el objetivo, el de los objetos, y que se parece al mundo de las ideas platónicas.

Aunque en ‘On sense and reference’ Frege hace alusión a un ámbito intersubjetivo, al ‘repositorio de significados de la humanidad transmitido de generación en generación’ como soporte de la significación de los conceptos y las proposiciones, en ‘The thought’ parece que hace alusión a un mundo parecido al platónico, un mundo en donde moran las ‘verdades eternas’:

‘...thoughts are neither things of the outer world nor ideas. ...A third realm must be recognized. What belongs to this corresponds with ideas, in that it cannot be perceived by the senses, but with things, in that it needs no bearer to the contents of whose consciousness to belong. (Frege, 1918; p.96)

Y, como hemos visto en el capítulo anterior, pone el ejemplo del teorema de Pitágoras, cuya existencia —según dicho autor— es igual de real que un planeta que todavía no haya sido descubierto por ningún ser humano.

La información como sentimiento

Aristóteles ya advirtió que hay sentencias, como las plegarias o la poesía, que no necesitan tener contenido proposicional;

There is not truth of falsity in all sentences: a prayer is a sentence but is neither true or false. The present investigation deals with the statement-

*Propuesta de un método de validación de esquemas conceptuales
y análisis comparativo de la noción de información
en los métodos de desarrollo de sistemas de información*

making sentence; the others we can dismiss, since consideration of them belongs rather to the study of rhetoric or poetry. (Aristotle, 350a B.C.; p.26)

Frege también desvela, aunque no elabora ninguna conclusión, que hay una clase de sentencias que no tienen referencia: con una poesía no se pretende transmitir información sobre hechos, ni explicar significados. Lo que un poema produce, mediante la combinación de la eufonía de las palabras y los significados que evoca en nosotros, son imágenes y sentimientos. En este caso se ve claramente que la información no se transmite sino que se evoca en el oyente. (Frege, 1892; p.42)

Este componente emocional o afectivo ha sido muy poco estudiado en el área de sistemas de información aunque podría ser una causa relevante en el fracaso de algunos proyectos de sistemas de información. (Leifer et al., 1994; Te'eni, 2001)

La información como síntoma

Si para algunos conceptualistas la unidad de significado está en los conceptos o en los universales, para Frege lo está en las proposiciones que expresan las sentencias y —como veremos más adelante— para Austin y Searle en los actos de habla, para la tradición hermenéutica de Schleiermacher, Dilthey y Gadamer está también en el trasfondo: *'The vocabulary and the history of the era of an author relate as the whole from which his writings must be understood as the part, and the whole must, in turn, be understood from the part.'* (Schleiermacher, 1828; p.24) O, como señala Dilthey utilizando la fórmula hermenéutica propuesta por Schleiermacher: *'The Formula: Understanding an Author Better Than He Understood Himself.'* (Citado en (Dilthey, 1860; p.147))

Habermas (1981) y Popper (1963, 1974) también diferencian entre 'lo dicho' y lo que se trasluce por 'lo dicho' recurriendo al modelo propuesto por el psicólogo alemán Karl Bühler referente a las tres funciones del uso de los signos en el lenguaje. Según Bühler, el signo lingüístico funciona simultáneamente como símbolo, síntoma y señal. *'It is a symbol in virtue of being correlated with objects and states of affairs, a symptom (indication, index) in virtue of its dependence on the sender, whose subjectivity it expresses, and a signal in virtue of its appeal to the bearer, whose external or internal behavior it steers like other traffic signs.'* (citado en (Habermas, 1981; p.275))

Bühler asocia el término ‘señal’ a la función apelativa del lenguaje, mientras que nosotros lo hemos asociado al de ‘señal en telecomunicaciones’; la función apelativa la estudiaremos más adelante. Lo que nos interesa resaltar ahora es la función de ‘síntoma’, la información que se comunica implícitamente. Por ejemplo, cuando una persona hace afirmaciones que no compartimos, para entenderla hemos de conjeturar bajo qué supuestos se convierten en ciertas. La información que se recibe no es sólo la del enunciado sino también la conjetura del trasfondo que hace verdadero el enunciado. Mediante las discrepancias de sentido o de referencia, se puede vislumbrar la visión del mundo del interlocutor.

Con las nueve acepciones de información presentadas no quedan agotadas éstas. Por ejemplo respecto al significado hemos distinguido entre el objetivo y el subjetivo, pero tal como lo plantea Frege esta subjetividad es la del oyente –del receptor. A continuación presentamos la subjetividad del hablante –del emisor—, que se conoce como semántica intencional. Posteriormente presentamos la información como significación –la que Frege califica como significado objetivo— pero desde un punto de vista intersubjetivo, la información como convención.

La información como intención comunicativa

‘When I use a word,’ Humpty Dumpty said in rather a scornful tone, ‘it means just what I choose it to mean —neither more nor less.’

‘The question is,’ said Alice, ‘whether you can make words mean so many different things.’

‘The question is,’ said Humpty Dumpty, ‘which is to be master —that’s all.’

Through the Looking Glass, Lewis Carroll

En este pasaje, Carroll muestra tres maneras de entender lo que significan las palabras. La de Humpty Dumpty, en la que el significado se reduce a la intención del hablante. La de Alicia, que insinúa que no se puede variar arbitrariamente el significado de las palabras, de lo que se infiere que éstas tienen un significado que es independiente del que las pronuncia. Al final, Humpty Dumpty reduce el problema a establecer

*Propuesta de un método de validación de esquemas conceptuales
y análisis comparativo de la noción de información
en los métodos de desarrollo de sistemas de información*

quién tiene poder para imponer el significado, el significado es una convención social, una norma social, y el problema consiste en saber quién tiene el poder para establecer normas.

La primera manera de concebir el significado, que se conoce como semántica intencional, ha sido desarrollada por Grice (1989). En esta teoría se intenta establecer el significado fundamentándolo en la intención comunicativa del hablante de dar a entender algo al oyente por medio de un signo. Para pasar del significado ocasional —que es el de la intención particular del hablante en cada acto de habla— al significado habitual, Grice se basa en que la repetición de actos de habla en los que se utiliza un mismo signo con la misma intención comunicativa, lleva al establecimiento convencional del significado del signo.

Esta conjetura de que el significado de las palabras está basado en su uso no es nuevo en filosofía. Ya la manifiesta Peirce: ‘...*there are symbols, or general signs, which have become associated with their meanings by usage. ... The word lives in the minds of those who use it.*’ (Peirce, 1894) y también, desde la aproximación hermenéutica, Dilthey: ‘*Linguistic usage is the basis of interpretation,...*’ (Dilthey, 1900; p.81), y Wittgenstein: ‘*For a large class of cases --though not for all-- in which we employ the word "meaning" it can be defined thus: the meaning of a word is its use in the language.*’ (Wittgenstein, 1953; p.20)

Aunque Putnam (1988) critica el definir ‘significado’ recurriendo al uso regular de las palabras, parece que, si no una condición suficiente para definirlo, sí que es una característica necesaria para la comunicación humana en, como dice Wittgenstein, un gran número de casos.

La información como convención.

El término convención tiene dos acepciones: la primera, que ya hemos explicado, como norma o práctica, tácita o explícita, que responde a una costumbre o uso social; y la segunda, con un matiz diferente, como convenio, acuerdo o negociación entre dos o más personas. Lewis (1969) se basa en esta última acepción para relacionar el significado con las convenciones que se establecen para resolver problemas recurrentes de coordinación entre personas.

Giddens sintetiza esta teoría: ‘*The meaning of a word in a linguistic community depends upon the norms or conventions which prevail in that community, to the effect that “the word is conventionally accepted to mean p”.* A convention can be

understood as a resolution of a coordination problem, as the latter is defined in game-theory. In a coordination problem, two or more people have a shared end that they wish to bring about, to do which each has to select from a series of alternative, mutually exclusive means.' (Giddens, 1976; p.97)

Sin embargo, Giddens discrepa de esta interpretación; considera que no todas las convenciones sociales deben su origen a la resolución de problemas de coordinación; presenta el contraejemplo de la convención que existe en determinadas sociedades de que los hombres no vistan faldas. Convención que no parece responder a ningún problema de coordinación.

Convenimos con Giddens que no toda convención social responde a un problema de coordinación, sin embargo, sí que es cierto que la solución de un problema de coordinación requiere el establecimiento de convenciones, ya sean éstas tácitas o explícitas y formalizadas.

Las convenciones no son únicamente regularidades del comportamiento humano. Por ejemplo, si llegamos a un país y observamos que todo el mundo circula por la izquierda, constatamos un hecho empírico, pero con esto no llegamos a saber, aunque podamos presuponer, que existe una regla, o norma, de tráfico que regule el comportamiento de sus habitantes. Las convenciones son sistemas de reglas y las reglas son motivos para que las personas que creen en ellas actúen de una forma determinada. Los sistemas de reglas producen expectativas de comportamiento de los otros. (Lewis, 1969)

En muchos casos las convenciones están estrechamente relacionadas con el significado usual de los signos. En algunos casos modifican o precisan los significados usuales de los signos. Por ejemplo, en algunos restaurantes los camareros codifican los platos del menú asignándoles un número mediante el cual realizan los pedidos al cocinero. Este es un caso en el que los mensajes pierden su significado connotativo quedándose reducidos a su significado meramente referencial.

Un caso heterogéneo es el de las señales de tráfico. Mientras que algunas son representaciones de la realidad, como por ejemplo la señal de curva a la derecha, otras, como una señal de dirección prohibida, no hacen referencia a la realidad sino que la crean. Un ejemplo en el que no existe ningún componente referencial es el juego de ajedrez. En el ajedrez se produce una inversión referencial, las palabras no actúan como

indicadores, como referencias, de los objetos, de los referentes, sino que son los objetos que actúan como referencias de las palabras que juegan el papel de referentes. Para entenderlo desarrollemos el ejemplo tal como lo esboza (Wittgenstein, 1953; p.15).

Wittgenstein se cuestiona cuál es el significado de un alfil en el juego de ajedrez. Para ello se pregunta cómo explicaríamos qué es el alfil a alguien que no sabe lo que es el ajedrez. En un primer intento se podría acudir al significado como referencia, el término alfil representaría a la pieza que está en el tablero, podríamos señalar la pieza y decir ‘esto es un alfil’, pero con esto no se entendería nada pues, en realidad, no es la palabra alfil que representa a la pieza, es la pieza que actúa como símbolo de la palabra (o término general, o concepto) alfil. Se produce una inversión, en lugar de ser la palabra que representa al objeto, es el objeto el que representa a la palabra. Si se intenta definir el significado de alfil describiendo cómo se mueve tampoco es suficiente. Para entender el concepto alfil, es necesario entender el juego, pero para entender el juego es necesario entender cómo se mueve cada una de las piezas. El juego de ajedrez es el conjunto de reglas que lo definen.

La descripción del juego de ajedrez no constata unos hechos físicos, una realidad objetiva (compuesta de objetos materiales que pueden ser percibidos por los sentidos). Empleando la terminología de Searle, la descripción crea una realidad social, una institución social. Las instituciones sociales son constituidas mediante reglas.

Searle distingue, aunque reconoce que no es capaz de establecer criterios de delimitación, dos tipos de reglas, las reguladoras y las constituyentes. Las reguladoras regulan formas de comportamiento que existen anteriormente. Las constituyentes, no sólo regulan sino que crean nuevas formas de comportamiento. (Searle, 1969; p.33)

Giddens critica esta clasificación mostrando que no existen dos clases disjuntas de reglas. Las reglas que aparentemente sólo son reguladoras, también tienen un componente constituyente. Por ejemplo, la regla ‘todos los trabajadores deben fichar a las 8 de la mañana’ es reguladora en tanto en cuanto regula el comportamiento de personas. Por otra parte, la misma regla tiene un componente constitutivo, referente al significado, a lo que nos permite entender. Este ejemplo, aduce, quizá no entre en la definición de lo que es ‘trabajo’ pero sí que nos permite

entender un concepto tal como 'burocracia industrial'. (Giddens, 1984; p.20)

Esta doble faceta de las reglas es una de las bases de su teoría de la estructuración, mediante la que intenta fundamentar una teoría de la sociedad. El carácter dual de la 'estructura', llama estructura a conjuntos de reglas que crean instituciones sociales, le permite intentar una síntesis entre las sociologías funcionalistas y estructuralistas (objetivistas) y las sociologías hermenéuticas e interpretativas (subjetivistas), más basadas en el significado. Es decir, le permite explicar cómo funciona el mundo social y también entenderlo.

Uno de los aspectos que resalta Giddens es la relación entre los sistemas de reglas y el poder en las organizaciones. La faceta constituyente, que conlleva una negociación entre partes, presupone una relación de poder entre las mismas; este hecho le lleva a afirmar que 'el lenguaje expresa asimetrías de poder y frecuentemente las normas sociales son impuestas dentro de sistemas de dominación' (Giddens, 1976; p.54). La faceta con más componente regulador también está relacionada con las formas de dominación pues una condición para la capacidad de regulación es la capacidad de sanción al incumplimiento de la regla. Por eso una regulación no puede ser conceptualizada sin tener en cuenta las formas de dominación y de poder. (Giddens, 1984; p.20)

Lo relevante de este planteamiento para nuestro propósito es que relaciona el sistema de información como sistema de reglas con, por una parte, el poder en la organización y, por la otra, con la vertiente de los problemas de coordinación y con la constitución de los significados en la transmisión de información. Esto es, muestra el papel que juega la convención en la constitución de los significados y su estrecha conexión con la capacidad de sanción.

Wittgenstein también contempla las lenguas desde este punto de vista convencional: *To obey a rule, to make a report, to give an order, to play a game of chess, are customs (uses, institutions).... To understand a language means to be master of a technique.* Wittgenstein p.81

Es obvio que el conocimiento de cómo hablar una lengua requiere la maestría de un conjunto de reglas que lleva a un uso regular y sistemático de los elementos de la lengua. Las lenguas son instituciones sociales. (Searle, 1969; p.13)

Partiendo de este hecho, Searle propone en su teoría de los actos de habla distinguir entre las distintas convenciones y considera que la unidad de significado no está en la palabra o en la oración sino en el acto de habla: *'The unit of linguistic communication is not, as has generally been supposed, the symbol, word of sentence, or even the token of the symbol, or word or sentence, but rather the production of issuance of the symbol or word or sentence in the performance of the speech act. To take the token as a message is to take it as a produced or issued token. More precisely, the production or issuance of a sentence token under certain conditions is a speech act, and speech acts (of certain kinds to be explained later) are the basic or minimal units of linguistic communication.'* (Searle, 1969; p.16) Actos de habla que pasamos a describir a continuación

La información como acto de habla

Frege reconoce que hay un tipo de sentencias que tienen significado pero no referencia, oraciones tales como las órdenes, las preguntas y las prohibiciones. Considera que no son proposiciones aunque están a su mismo nivel (Frege, 1892; p.46). Por ejemplo, la declaración 'el libro está sobre la mesa', si la transformamos en pregunta: '¿el libro está sobre la mesa?', o en una orden: 'Pon el libro sobre la mesa', pierde su carácter asertórico, constatativo, de descripción de la realidad. Ya no es correcto afirmar su veracidad o falsedad. Cuando el hablante emite la oración interrogativa está pidiéndole al oyente que confirme la verdad o falsedad de la aseveración, pero la oración tal como es emitida por el hablante no es una proposición, aunque siga teniendo un componente proposicional, es evidente que tiene otra acepción. Del mismo modo, en el caso de la sentencia imperativa, no se está describiendo una realidad, el contenido de la proposición no se corresponde con la realidad, lo que se está requiriendo es que el oyente realice una acción para que la realidad se ajuste a la proposición.

Wittgenstein también señaló que además de las proposiciones existen innumerables clases de sentencias cuya multiplicidad no está prefijada. Estos tipos de sentencias se sustentan en que hablar es una actividad y que distintas actividades requieren distintos tipos de oraciones. A estos sistemas de reglas que forman parte de las actividades que se sustentan en formas de vida les llama juegos de lenguaje. (Wittgenstein, 1953; p.11)

Austin (1955) desarrolló estas ideas en su libro *'How to do things with words'*. El autor señala que desde antiguo, los filósofos han caído en una trampa, que llama 'falacia descriptiva', consistente en asumir que las declaraciones (statements) únicamente tienen una función descriptiva, la constatación de hechos, y que éstas deben necesariamente ser verdaderas o falsas.

Muestra una gran cantidad de ejemplos de declaraciones que no son constataciones sino que son acciones, actos de habla, por ejemplo: 'declaro a esta pareja marido y mujer'. Este tipo de actos de habla, a los que llama realizativos (performative), no son ni verdaderos ni falsos, pero pueden ser 'afortunados' (*happy*) o 'desafortunados' (*unhappy*) según sean proferidos en la situación, por la persona, y con el protocolo adecuado, esto es, se sigan unas determinadas reglas que lo perfeccionen¹⁰. Posteriormente hace ver que las constataciones también son actos de habla pero argumenta que no es posible distinguir claramente las declaraciones constataciones de las realizativas, por lo que propone una clasificación transversal a ellas caracterizando los actos de habla según tres componentes el locutivo (locutionary), el ilocutivo (illocutionary) y el perlocutivo (perlocutionary).

Con el término 'locutivo' se refiere al acto de emitir la declaración, a su aspecto sintáctico y a su componente constatativo (significación y referencia). El componente ilocutivo se refiere a la distinta 'fuerza' (*force*) con la que podemos hacer una declaración. Por ejemplo la declaración 'ven aquí' puede decirse como una orden 'te ordeno que vengas aquí', como una sugerencia 'te sugiero que vengas aquí', o como un ruego 'te ruego que vengas aquí'. Aunque reconoce que la 'fuerza ilocutiva' forma parte del significado de la oración, prefiere distinguirla de lo que se llama significado tradicional (significación y referencia) (Ibid., p.100). Con el término 'perlocutivo' Austin indica los efectos que el acto de habla produce en la audiencia; por ejemplo, la frase 'te aconsejo que....' puede persuadir, preocupar, alarmar, atemorizar o hacer que el oyente salga corriendo.

Austin, mediante el examen de los verbos ingleses, propone una clasificación de los actos ilocutivos en cinco tipos cuya delimitación no

¹⁰ Si nos atenemos al significado castellano de 'correcto', según el DRAE, como 'libre de errores', 'conforme a las reglas', quizá sea más adecuado traducir *happy* como correcto o apropiado y *unhappy* como incorrecto o inapropiado.

es muy clara. Searle lo argumenta y propone otra clasificación utilizando doce criterios, de los cuales mencionaremos los dos primeros que, por otra parte, son el que el autor considera más discriminadores aunque reconoce que no son suficientes para establecer la clasificación (Searle, 1979).

Junto a la noción de fuerza ilocutiva de los actos de habla presentada por Austin, Searle introduce la noción de propósito ilocutivo '*illocutionary point*', que expresa el significado central, genérico, del acto de habla. Por ejemplo, los verbos solicitar, pedir, requerir, demandar y exigir, tienen el mismo propósito ilocutivo —conseguir que el oyente realice una acción— pero distinta fuerza ilocutiva.

El segundo criterio clasificatorio que propone es la dirección de ajuste que se establece entre las palabras y el mundo. Para entender esta idea Searle pone el ejemplo de un hombre que va al supermercado con la lista de la compra que le ha hecho su mujer. Recorrerá los pasillos buscando los productos que hay en la lista para ponerlos en el carro de la compra, de tal modo que al final habrá conseguido que el estado del mundo se ajuste a lo que indicaba la lista. Por otra parte, cuando pase por caja realizarán otra lista constatando lo que lleva dentro del carro. Esta lista ajusta las palabras al estado del mundo. Para ver más claramente la distinta dirección de ajuste entre mundo y palabra consideremos la situación en que se produce una discrepancia entre la lista y el estado del mundo. En el segundo caso, si no se corresponde la lista del cajero con los objetos que hay en el carro, se deberá modificar la lista, las palabras. En el primer caso, si cuando llega a casa se detecta un error, no se podrá subsanar modificando la palabra, se tendrá que cambiar el estado del mundo.

Las cinco clases que Searle propone son:

Asertivas: El propósito ilocutivo es constatar un hecho. Las palabras se ajustan al mundo. La fuerza ilocutiva la expresan los distintos verbos: afirmo que p, me parece que p, creo que p, estoy seguro de que p.

Directivas: El propósito ilocutivo es que un hablante pide al oyente que realice una acción, o sea, que el mundo se ajuste a las palabras, que una proposición se convierta en cierta. Te ruego que p, te pido que p, te ordeno que p, te exijo que p.

Compromisivas (*Commissives*): El hablante promete al oyente realizar una acción. El ajuste es, como en el caso anterior, del mundo a la palabra, pero la diferencia está en quién debe realizar la acción. Mientras que en las directivas es el oyente, en las compromisivas es el hablante. Procuraré que p, me propongo que p, me comprometo a que p

Declarativas: La característica principal de esta clase es que la ejecución correcta de un acto de habla declarativo produce el efecto de establecer una correspondencia entre el contenido proposicional de la sentencia y la realidad. Mediante palabras se crea algo. Por ejemplo, la declaración 'la persona X queda nombrada director general', si es un acto de habla correcto (afortunado), esto es, si se produce conforme a las reglas establecidas, produce una nueva realidad.

Expresivas: El hablante revela su estado psicológico. Lamento que p, me alegro que p, te agradezco que p. En este caso no existe ajuste entre las palabras y el mundo externo. La verdad de lo expresado en la proposición simplemente se presupone.

La información como acto social

Habermas (1981) discrepa de la clasificación de Searle y propone otra clasificación de los actos del habla. Las razones de esta discrepancia se pueden atribuir a que la finalidad que persiguen sus trabajos es distinta. El propósito de Habermas es fundamentar las ciencias sociales en la teoría del lenguaje, en la teoría de la acción comunicativa que dicho autor propone. Para ello, recurre a la teoría de los signos de Bühler, que distingue tres funciones de los mismos: la función constativa, la expresiva y la apelativa. Las dos primeras son semejantes a las de Searle, la apelativa se refiere a exigencias dirigidas a los oyentes. Habermas la diferencia en dos, los actos de habla imperativos, que se asemejan a los directivos de Searle; y los regulativos, en los que subsume los compromisivos y declarativos de Searle.

Habermas reduce los tipos sociológicos de acción a cuatro conceptos básicos y los asocia a los actos de habla que hemos visto: la acción teleológica, que se realiza mediante los actos de habla imperativos; la acción comunicativa, de los actos de habla constativos; la acción normativa, realizada mediante los regulativos; y la acción dramática, hecha con los expresivos.

La acción teleológica, orientada a fines, la divide en dos subcategorías: la acción instrumental y la estratégica. La primera, en la que el actor, para conseguir realizar algo, elige y aplica los medios adecuados. Y la estratégica, en la que el actor, para conseguir sus fines, para maximizar su utilidad o sus expectativas de utilidad, debe tener en cuenta las acciones y expectativas de utilidad de otros agentes. Este tipo de acción es la que sirve de planteamiento en la teoría de juegos y en economía. La acción instrumental se puede representar como la que hace un ingeniero para desarrollar un proyecto en el que no intervengan factores políticos, esto es, en el que no es necesaria la acción estratégica.

Ayudado de estos conceptos, Habermas formula una generalización del concepto de validez veritativa de las proposiciones, pasando del plano semántico al pragmático. El autor propone que un acto de habla es válido si, atendiendo a la acción teleológica, es eficaz, esto es, puede ser criticado, en el subcaso de acción instrumental, desde el punto de vista de la factibilidad de la acción, de su posibilidad de éxito, y en el caso de la subclase de acción estratégica, puede ser rechazado al no aceptar una pretensión de poder; si, atendiendo a la acción constatativa, es verdadero; si, respecto a la acción normativa, es legítimo, puede ser cuestionado en cuanto a su rectitud normativa; y, en relación a la acción dramatúrgica, es sincero, se puede poner en duda la pretensión de veracidad.

En esta noción generalizada de validez fundamenta su idea de racionalidad social.

It belongs to the communicative intent of the speaker

(a) that he perform a speech act that is right in respect to the given normative context (legitimate)

(b) that he make a true statement

(c) that he express truthfully his beliefs, intentions, feelings, desires, and the like (Habermas, 1981; p.307)

Dicho de forma negativa, se puede atacar el intento comunicativo de un hablante en el plano dramatúrgico, poniendo en duda su sinceridad; o en el plano normativo, cuestionando su corrección en relación a unas normas institucionales, que pueden ser tanto morales, como legales, como organizativas; o, en el plano comunicativo, se puede negar la verdad de las proposiciones, su correspondencia con el mundo; o en el

plano teleológico, se puede cuestionar las posibilidades de éxito de la propuesta, en el caso de acción instrumental, o el poder del hablante para imponer su voluntad, en el caso de acción estratégica.

Conclusiones de este capítulo

Valoración

La relevancia de la clasificación de Searle es que nos permite explicitar, categorizándolos, los diálogos que se establecen en las empresas. El pedido que hace un cliente es una directiva, la confirmación del pedido es una compromisiva, el albarán es una asertiva y la factura es una declarativa (es un hecho institucional), aspectos éstos que se omiten en los métodos de modelado conceptual.

Aunque pensamos que en el campo de los sistemas de información la utilización de los esquemas conceptuales es irrenunciable, pues ofrecen unas ventajas incomparables frente a otras técnicas en sistemas de información, lo cierto es que despiezan los documentos, los formularios de empresa, perdiéndose así la unidad del significado que expresan, que no es más que el formulario. Podríamos hablar de ‘acto de empresa’ – transacción empresarial— en vez de ‘acto de habla’.

En el área de métodos de modelado de sistemas de información podemos ver la aplicación de estos esquemas de actos de habla en los trabajos pioneros de Flores y Winograd, en los que conciben una organización como una red de conversaciones basadas en los actos de habla de Searle, del que sólo utilizan los directivos y los compromisivos (Winograd and Flores, 1986).(Flores et al., 1988; Medina-Mora et al., 1992 ; Flores; Winograd, 1987) Estas ideas se implementaron en el sistema Coordinator y posteriormente en el Action Workflow.

Los esquemas de Searle también subyacen en el método SAMPO (Speech-Act-based Modeling Approach) (Goldkuhl and Lyytinen ,1982; Lyytinen, 1985; Auramäki et al., 1988; Lehtinen and Lyytinen, 1994; Auramäki and Lyytinen, 1996 ; Goldkuhl, 1996) (Goldkuhl ,1996) e independientemente en (Johannesson, 1995).

Sin embargo, estos métodos se centran en el aspecto comunicativo pero soslayan el proposicional que capturan los esquemas (Agerfalk and Eriksson, 2004). Pensamos que la única vía de solución a este dilema es

la de definición de cálculos formales de formularios, como hemos propuesto en el capítulo 4.

Pero el limitarse a los actos de habla al diseñar un nuevo sistema de información es no contemplar toda la problemática social que comporta el instaurar un nuevo sistema de información. No basta con la adecuación veritativa del mismo. El modelo de Habermas es esclarecedor cuando se trata de recapacitar sobre las distintas acciones a tener en cuenta en la implementación de un sistema de información; presenta un marco conceptual más amplio.

Normalmente los métodos sólo tienen en cuenta las acciones instrumentales y comunicativas, pero no las dramáticas (dar sentido a la nueva realidad), estratégicas (relaciones de poder que hay y pueden ser modificadas) y las normativas (en el sentido de reglas de coordinación).

Que los factores sociales son relevantes al diseñar sistemas de información lo muestra una amplia bibliografía, por ejemplo (Pettigrew, 1972; Kling, 1980; Markus, 1983; Saunders and Scamell, 1986; Murray, 1989; Davenport et al., 1992; Chang and Lin, 2000 ; Jasperson et al., 2002)

Desde la perspectiva de Habermas, una organización no sólo es una red de conversaciones, sino también una red de compromisos en los que la validación no debe limitarse a la referencial, veritativa, sino también a la normativa, a la dramática, a la instrumental y a la estratégica.

Una aplicación del esquema de Habermas al desarrollo de sistemas de información está en (Dietz, 2001). Para un survey con referencias a otros métodos (Auramäki and Lyytinen, 1996). El problema de estos métodos, al igual que ocurre con los de actos de habla, es el de conciliar estos enfoques con los de modelos conceptuales. (Schoop, 2001)

Al utilizar los modelos de Searle y Habermas, cabe no desechar los otros aspectos constitutivos de la información. A modo de resumen presentamos en el diagrama 14 además de las acepciones de estos autores, las otras once acepciones de la información que hemos contemplado en este capítulo.

En primer lugar está lo que hemos denominado en el diagrama el triángulo de la comunicación, que responde al clásico esquema de la comunicación en el que un hablante, o emisor, intenta comunicar unos

significados (**intención** comunicativa) a un oyente, o receptor, para lo cual hace uso de un conjunto de palabras con significado público (con **significación**). La intención del hablante no siempre coincide con la significación ni con el significado que entiende el oyente (**sentido**).

La significación, los significados públicos, en el triángulo que en el gráfico hemos denominado ‘representación’, están entrelazados con su capacidad, directa o indirecta, de referencia a objetos (**símbolo**), y dentro de una gramática de un lenguaje (**signo**).

A su vez, en el triángulo que hemos llamado de formación, o conformación, la significación también está relacionada con la **convención** en su doble vertiente, como uso social de unos signos y como la forma de constitución de ‘objetos’ sociales (instituciones sociales). Pero los objetos referenciados no sólo pueden ser sociales, también se puede intentar referenciar objetos físicos, agrupados en clases, mediante la observación de **regularidades**.

Por otra parte, volviendo al triángulo de la comunicación, cuando el hablante comunica algo al oyente, le transmite la información que intenta comunicarle y además un trasfondo comunicativo (**síntoma**). Y, además, puede evocar en el oyente emociones (**sentimiento**), además del significado que se infiere del uso público de los signos.

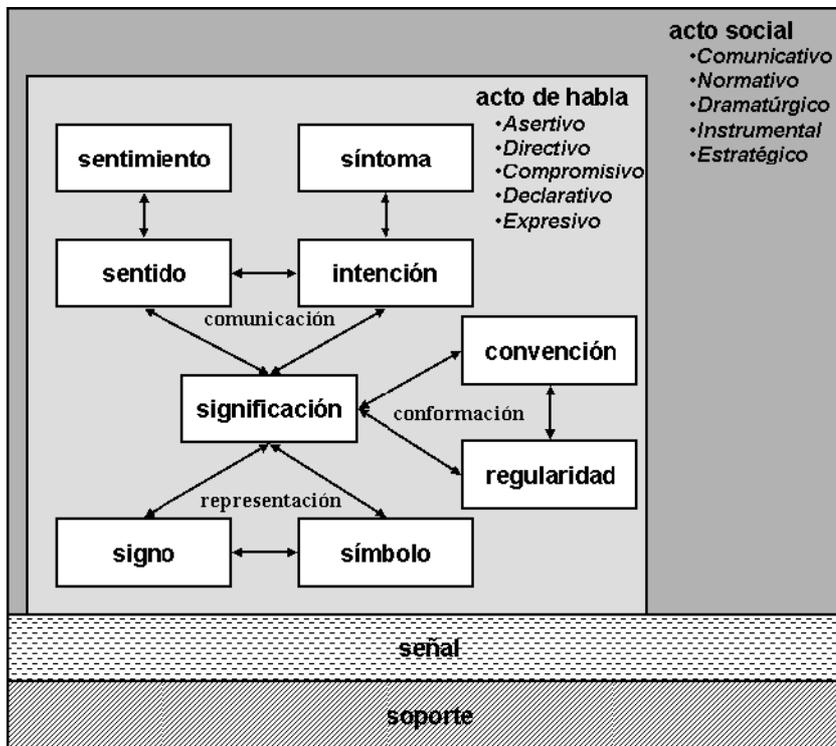


Diagrama 14. Resumen de las trece acepciones de la noción de información.

Limitaciones y contribuciones de este estudio

La primera limitación de este estudio es su incompletitud. Hemos intentado estudiar las acepciones que hemos considerado más relevantes para el dominio de problemas de sistemas de información, que son sistemas formales –mecánicos— para la comunicación humana. Con ello no pretendemos haber agotado todas las acepciones del término ‘información’. Por ejemplo, no hemos estudiado sus acepciones en otras disciplinas como la física o la economía. Para un estudio de éstas véase (Boisot and Canals, 2004).

En segundo lugar, en este capítulo no se ha conseguido crear un modelo de la noción de ‘información’, aunque el diagrama 14 pueda dar esta apariencia. Si se ha presentado gráficamente ha sido más por razones nemotécnicas que por la coherencia de las interrelaciones presentadas.

La posible contribución de este estudio es la presentación de un esbozo de las múltiples facetas complementarias de la noción de información, esbozo que no hemos encontrado en la literatura

consultada. La utilidad de este catálogo de significados del término 'información' la defiende Aristóteles en sus Tópicos con las palabras que hemos incluido al principio de este capítulo, esto es, el conocer las distintas acepciones del término puede ayudarnos a utilizarlo con mayor precisión, aumentando así el rigor de nuestra disciplina '*...for if we know the number of meanings of a term, we shall certainly never be misled by false reasoning.*'

Por otra parte, este estudio muestra las insuficiencias de los modelos conceptuales de datos, que se limitan a especificar y validar únicamente la información como símbolo y significación de los sistemas a construir. Reforzando así nuestra propuesta de utilizar el formulario como objeto idóneo para especificar los diferentes actos de habla que pueden conllevar y para la validación del esquema conceptual por parte de los usuarios.

Además, el formulario es el objeto mediante el que se pueden especificar lo que antiguamente se llamaban circuitos administrativos y hoy en día se llaman *workflows*, que es el nivel de abstracción adecuado para que los directivos puedan valorar los diseños propuestos dentro del ámbito de la acción social, tal como la concibe Habermas.

Capítulo 6: Conclusiones

Según distintos estudios empíricos, una de las principales causas de la alta tasa de fracasos en el desarrollo de proyectos de sistemas de información de empresa es debida a los malentendidos que se producen entre usuarios, analistas y programadores al establecer los requerimientos de los sistemas a construir. En este trabajo se presentan algunas razones que pueden explicar estos malentendidos y se propone un nuevo método para paliar este problema.

Un componente esencial para la especificación de los requerimientos de un sistema de información informatizado es el esquema conceptual. El esquema conceptual de la empresa que se quiere modelar es, básicamente, un vocabulario coherente y compartido por toda la organización, que permite la rápida comunicación para el control y coordinación de los procesos de la empresa.

Según el IEEE (1999), el esquema conceptual proporciona una definición única, integrada y sin sesgo hacia los distintos usos, de los conceptos relevantes de una empresa. La función primordial del esquema conceptual es establecer una definición consistente de los significados de los conceptos y de sus interrelaciones.

El propósito de los métodos de modelado conceptual es la especificación del esquema conceptual. En este trabajo se han presentado lo que a nuestro juicio son dos insuficiencias de los métodos de modelado conceptual existentes y una propuesta que puede atenuar estas limitaciones.

Contribuciones a la discusión del problema

La primera limitación de los métodos de modelado conceptual tradicionales es que sólo modelan los datos pero no las acciones que se hacen sobre los datos tal como las realizan los usuarios. Como se ha indicado en el capítulo 3, esta limitación de los esquemas conceptuales ha sido señalada recientemente como un área de investigación que es necesario desarrollar.

La segunda limitación de los métodos tradicionales reside en la forma de validación del esquema conceptual. Los métodos de modelado conceptual existentes, al no problematizar la obtención de una definición única, integrada y sin sesgo de los conceptos relevantes de la empresa a modelar, o bien ignoran el problema de la validación, o proponen una validación directa por parte del usuario. Nuestra propuesta —tal como se ha explicado en el capítulo 2, en la sección donde se discute el enfoque axiomático, y en el capítulo 3, en donde se hace una valoración de la notación— reclama la necesidad de la validación del esquema conceptual y la plantea a través de los formularios.

Expresado en términos de sistemas axiomáticos, en este trabajo hemos visto que mientras que los métodos de modelado conceptual se basan en una axiomatización euclídea, el que se ha propuesto en esta tesis se basa en la axiomatización hilbertiana, que es la que es asumida actualmente en lógica formal.

La axiomatización euclídea presupone que se puede tener la certeza de la verdad de los axiomas antes de comprobar si los teoremas que de ellos se deducen son verdaderos. Esto es lo que parece desprenderse del influyente informe ISO (Griethuysen ,1982) en donde se define el esquema conceptual como '*A consistent collection of sentences expressing the necessary propositions that hold for a universe of discourse.*' Y en donde precisan el significado de proposición necesaria: '*A proposition asserted to hold for all entity worlds and therefore must be part of all possible proposition worlds.*'

La axiomatización hilbertiana —o expresado con más precisión— el método científico, que utiliza la axiomatización hilbertiana, asume que los axiomas son conjeturas que se consideran ciertas cuando los teoremas que de ellos se deducen son verdaderos, se corresponden con la realidad.

En cada uno de estos dos enfoques subyace una noción de información como representación distinta. Si presuponemos que los conceptos, o términos generales, que se utilizan en una empresa tienen un significado suficientemente preciso y se pueden definir independientemente de su contexto concreto y situación de uso, entonces la validación del esquema conceptual no será demasiado importante, o, en cualquier caso, se podrá validar directamente preguntando a los usuarios.

Por el contrario, si presuponemos que los términos empleados en una empresa en algunos casos tienen un significado que no es fácil de precisar y que además puede variar de un departamento a otro y dentro de un mismo departamento puede depender de la situación, entonces consideraremos que el esquema conceptual no es más que una conjetura y que el significado de los términos se debe validar en cada contexto y situación concreta.

Para intentar valorar las ventajas e inconvenientes de nuestra propuesta frente a otras, fueron revisados, además de los métodos de modelado conceptual, métodos pertenecientes a distintas áreas de investigación tales como: la ingeniería ontológica, la ingeniería de requerimientos, la ingeniería del software, los métodos formales de especificación de software y el método relacional.

En un principio, la comparación pareció imposible —en términos kuhnianos diríamos que son paradigmas inconmensurables— y algunas propuestas nos resultaban incluso ininteligibles. La clave de entendimiento y de comparación fue obtenida al descubrir que la raíz de la discrepancia entre las distintas clases de propuestas estaba en que concebían la noción de información como representación de manera diferente. Manera que se asemeja a las discrepancias que se han mantenido durante toda la historia de la filosofía entre tres formas distintas de entender lo que es un concepto, o universal, o término general, que simplificamos en tres escuelas: la conceptualista, la realista y la nominalista. Esto llevó a revisar las posiciones de diversos filósofos a lo largo de la historia y nos permitió posicionar los distintos métodos de análisis y diseño de sistemas de información que habían sido revisados.

Esta revisión permitió también detectar una confusión entre las nociones de idealismo y nominalismo que introdujeron Burrell y Morgan en su libro *Sociological Paradigms and Organisational Analysis* y que han

heredado todos los autores de sistemas de información que se han basado en dicho texto. Estos resultados han sido presentados en el capítulo 4.

Mientras se efectuaba la revisión en filosofía quedó patente que la noción de información no se agota en su capacidad referencial. Esto se ha presentado en el capítulo 5 como un catálogo de las distintas nociones detectadas. A diferencia del esquema del capítulo 4, en el que las distintas nociones son alternativas (contradictorias), en éste son complementarias.

Contribuciones del método propuesto

En este trabajo se ha presentado una notación y un procedimiento que permite expresar los formularios de empresa como una deducción formal a partir del esquema conceptual. La idea intuitiva es muy sencilla: de cada relación del esquema se pueden obtener distintos formularios base, que se presentan en dos formas básicas, primera forma visual y segunda forma visual. Los formularios más complejos se pueden expresar conectando y superponiendo distintos formularios base.

Se ha propuesto una gramática que permite expresar formalmente los formularios y unas reglas de inferencia que permiten deducirlos formalmente del esquema conceptual. Esto es, se ha intentado definir un cálculo formal de formularios mediante el que se pueden expresar los documentos transaccionales típicos de las empresas y organizaciones definiendo un vocabulario y una gramática que permite expresar los formularios como fórmulas del lenguaje, unos axiomas (que se expresan mediante el esquema conceptual) y unas reglas de inferencia que permiten obtener los formularios válidos, y sólo estos, a partir de los axiomas.

El término ‘válido’ referido a los formularios es bivalente, tiene dos aspectos: el formal y el convencional. Por una parte, en su aspecto formal, en el contexto del cálculo, un formulario válido es aquel que es un teorema, esto es, es aquel que puede ser deducido a partir de los axiomas mediante las reglas de inferencia. En su aspecto convencional, un formulario es ‘válido’ para un usuario si se ajusta a los usos y costumbres establecidas. Si se corresponde con sus expectativas.

El problema de definición de un buen cálculo de formularios es empírico, es un problema de antropología cultural. Un buen cálculo de formularios es aquel que es un modelo de una realidad social; en otras palabras, es aquel que hace coincidir la validez formal con la validez convencional de las fórmulas que expresan los formularios.

Puesto que los formularios son una convención social, las reglas ‘correctas’ de su composición tienen una parte de arbitrariedad, las reglas ‘correctas’ son un uso social asumido implícitamente. El problema empírico es el descubrir en los usos sociales las convenciones implícitas y explicitarlas formalmente.

En este trabajo nos hemos limitado a explicitar las reglas de los formularios más usuales. Todavía queda una importante tarea de estudio empírico de formularios menos usuales. Por esta razón no proponemos nuestro método como una solución final, definitiva, sino como una prueba de concepto de una vía de investigación que, a nuestro juicio, es digna de ser proseguida.

A juzgar por la revisión de la literatura que hemos realizado, esta aproximación al problema de análisis y diseño de sistemas de información mediante la utilización de formularios ha sido muy poco investigada académicamente. Y en estos pocos casos los trabajos consultados se limitan a la parte de elicitación de los requerimientos, pero no al de su especificación y validación. A este respecto no hemos encontrado ningún trabajo publicado.

Sin embargo, el formulario es el objeto que se emplea habitualmente para la comunicación de las operaciones diarias en las empresas y organizaciones, tanto las internas, entre los distintos departamentos, como las externas, en las que se utilizan para comunicarse con otras empresas y organizaciones. Este hecho lo hace especialmente adecuado para que los usuarios puedan validar las especificaciones de los sistemas propuestos.

Además, el método que proponemos puede permitir expresar por medio del formulario otros componentes de información que en los esquemas conceptuales no es posible hacer. Los formularios son los ‘*business acts*’, el equivalente a los ‘*speech acts*’ de las teorías del lenguaje. Forman las unidades mínimas de significado en las que se pueden incorporar los distintos propósitos ilocutivos que se mencionaron en el

capítulo 5. Y esto se puede conseguir, mediante la calculización de los formularios que hemos propuesto, sin renunciar a las ventajas de concisión, simplicidad e integración que proporcionan los métodos de modelado conceptual.

Por último, queremos destacar que el método que proponemos refuta las afirmaciones de algunos autores que, como vimos en el capítulo 1, en la sección ‘Tres facetas del problema’, declaran que es imposible obtener especificaciones que sean inteligibles para los usuarios y, al mismo tiempo, formales. Si bien, es cierto que esto se ha logrado reduciendo el dominio de problemas a los formularios de empresa.

El lenguaje presentado, o mejor, la vía de investigación que proponemos, demuestra a nivel de prueba de concepto que es posible conseguir especificaciones que simultáneamente sean formales y comprensibles por los usuarios que las deben validar.

La comprensibilidad de las especificaciones es debida, por una parte, a su comprensibilidad (concisión), y por otra parte a la familiaridad que tienen los usuarios con los formularios. La familiaridad no la dan las fórmulas sino las pantallas que generan, que es lo que entiende el usuario.

La formalidad de la especificación, debida a la axiomatización de formularios que proponemos, permite que los programadores puedan construir los programas sin necesidad de conocer el significado de los términos del esquema conceptual. Así se pueden evitar los posibles, y probables, malentendidos que se producen entre analistas y programadores.

Los malentendidos entre los prescriptores (usuarios) y los analistas quedan atenuados al hacer que los primeros puedan validar los formularios. Evitando así el difícil problema que se le presenta al analista cuando se le demanda que obtenga la definición del significado de los términos generales del esquema conceptual con precisión y exactitud. Precisión y exactitud que frecuentemente los usuarios son incapaces de proporcionarle.

Sumario de las principales contribuciones

Como resumen de lo que consideramos son las principales aportaciones de esta tesis, en el sentido de que plantea problemas y

propone vías de solución que anteriormente han sido poco estudiados, son los siguientes.

Respecto al lenguaje de especificación propuesto

Se ha presentado un lenguaje de especificación conceptual y formal de sistemas de información basado en los formularios de empresa que:

- Es más completo que los tradicionales de modelado conceptual, al modelar no sólo los datos sino también las acciones que los usuarios realizan sobre los datos.
- Muestra la viabilidad de los formularios para realizar la especificación formal de sistemas de información.
- Muestra la posibilidad de conseguir especificaciones que sean formales y, a la vez, inteligibles por los usuarios que las deben validar.
- Permite una manera de validar los esquemas conceptuales que en determinadas situaciones (cuando los términos generales del vocabulario de la empresa a modelar son imprecisos y ambiguos) es más adecuada que la de los métodos de modelado conceptual tradicionales.

Respecto los problemas de especificación de requerimientos

El propósito último del método que proponemos, al igual que el de los otros métodos de especificación existentes, es conseguir unas especificaciones de los sistemas de software a construir que se correspondan lo más fielmente posible con lo requerido por los usuarios y que además no puedan ser malinterpretadas por los programadores que han de construir el sistema de software que satisfará las necesidades expresadas por los usuarios. Como ya se vio, el malentender los requerimientos es una de las principales causas de la alta tasa de fracasos de los proyectos de sistemas de información.

Evaluar la oportunidad y adecuación de un método para solucionar una determinada clase de problemas, depende de la naturaleza que atribuyamos a los problemas. El problema en el que nos hemos centrado es el de malentender los requerimientos en los proyectos de sistemas de información, que son fallos en la transmisión de la información entre

usuarios y analistas, entre analistas y programadores, y entre los usuarios que utilizarán el sistema de software para comunicarse.

Pero la naturaleza de este problema depende de la presupuesta naturaleza de la información, depende de las características que se atribuyan a la información. En este trabajo hemos reflejado que buena parte de los distintos niveles de ambición y de las estrategias de especificación que tienen los distintos métodos revisados, se pueden explicar en base a las diferentes características que se atribuyen a la información.

Respecto al análisis de las nociones de información

Para justificar la ventaja del método de validación de esquemas conceptuales propuesto frente a otras alternativas, se ha realizado un análisis comparativo de las diferentes nociones de información asumidas en el campo de los sistemas de información. Este análisis, que se ha basado en las diferentes nociones que algunos filósofos han propuesto, ha producido los siguientes resultados:

- Una aclaración de la confusión entre idealismo y nominalismo existente en el área de sistemas de información.
- Un esquema de las distintas y contradictorias nociones de información como representación asumidas en el área de sistemas de información, que permite explicar los diferentes niveles de ambición y estrategias de distintos métodos existentes en la actualidad.
- Un mapa de las distintas y complementarias acepciones del término información, que muestra las carencias de los métodos de modelado conceptual, que reducen la información únicamente a su función referencial.

Limitaciones

Las limitaciones de este trabajo pueden ser agrupadas en tres clases: en cuanto al ámbito de aplicabilidad, respecto al lenguaje de especificación, y por el análisis de la noción de información.

Relativas al ámbito de aplicación

Por su propio objetivo, el método propuesto es sólo aplicable a una parte de la comunicación que se realiza en las empresas y organizaciones, aquella que se realiza mediante documentos muy estructurados, esto es, formularios de empresa.

Relativas al lenguaje de especificación

Capacidad expresiva del lenguaje propuesto

Al igual que ocurre con los otros lenguajes de especificación, la capacidad expresiva del lenguaje propuesto no es tan completa como la de un lenguaje de programación. Los lenguajes de especificación deben encontrar el equilibrio entre la simplicidad, la concisión, la formalidad del lenguaje propuesto y su aplicabilidad. Por ejemplo, si se renuncia a la concisión, cualquier lenguaje de programación es claramente superior a un lenguaje de especificación. Y si además renunciamos a la formalidad, la prosa corriente es evidentemente superior a los anteriores. Aun así, manteniendo el equilibrio indicado, cabe enriquecer la notación que se ha propuesto para que sea capaz de especificar los formularios menos usuales.

Incompletitud de las reglas de inferencia

Como hemos indicado anteriormente, las reglas de inferencia propuestas requieren un considerable estudio empírico de las convenciones sociales que crean la realidad social de los formularios, de modo que se pueda aproximar el conjunto de los formularios formalmente válidos al de los que son convencionalmente válidos.

Relativas al análisis de las distintas nociones de información

Los distintos, y distantes, programas de investigación relacionados con sistemas de información que se han intentado abarcar ha producido que la revisión de la literatura no haya sido exhaustiva. El objetivo del estudio no ha sido una revisión sistemática de la literatura sino una búsqueda de autores del área de sistemas de información que ejemplificaran las distintas posiciones deducidas del ámbito filosófico.

La última, y fundamental, limitación de este trabajo es producto de haber relacionado demasiadas áreas de conocimiento: lógica formal, teoría de lenguajes formales y construcción de compiladores, filosofía (de

varias tradiciones y épocas), ingeniería de requerimientos, modelado conceptual de sistemas de información, ingeniería ontológica y métodos formales de especificación de software. Es obvio que esto tiene sus limitaciones, por eso, las certezas que se pudieran desprender de este escrito deben ser consideradas conjeturas que requieren otros estudios posteriores en un futuro.

Futuras líneas de trabajo

Buena parte de las líneas de trabajo que abre esta tesis se deducen de sus limitaciones. Respecto al cálculo de formularios propuesto, sería conveniente desarrollar la capacidad expresiva de la notación y completar las reglas de inferencia, de modo que la mayor parte de los formularios que se utilizan en las organizaciones fuera expresable y deducible de un esquema conceptual cuya capacidad expresiva también se debería enriquecer. Respecto a la noción de información, es necesario profundizar, aumentar el rigor y ampliar las distintas nociones de información que son relevantes en el análisis y diseño de sistemas de información de empresa; y también el demarcar el ámbito de problemas en el que es adecuada cada una de las nociones.

Una vía de trabajo que se abre está basada en la biunivocidad existente entre las fórmulas y los formularios que generan. El estudio del proceso inverso permitiría la creación de un sistema de especificación visual de sistemas de información en el que las fórmulas no se tendrían que escribir, en el que se podrían especificar los formularios manipulando visualmente las relaciones del esquema conceptual.

Dentro de esta vía de visualización, si nos basamos en el formulario, también parece posible la visualización de los circuitos administrativos (workflows), que reflejan las transformaciones que sufren los documentos a lo largo de los procesos de empresa. Esto permitiría a los directivos y prescriptores una comprensión más profunda de la repercusión de los nuevos sistemas de información que se diseñan. Todo esto se podría conseguir integrando estas nuevas especificaciones a los esquemas conceptuales y sin renunciar al carácter formal y a la concisión de éstos.

Otro aspecto, que sólo se ha señalado en este trabajo y que es susceptible de desarrollo, es el enriquecer las especificaciones de los

formularios incorporando los aspectos ilocutivos de los actos de habla de Searle y las facetas sociales, tal como las plantea Habermas, de las acciones comunicativas que conllevan los circuitos administrativos.

Una última vía de investigación posible es la conversión del lenguaje de especificación en un generador de aplicaciones informáticas de empresa. Pero esto requeriría aumentar su capacidad expresiva con el inconveniente de perder su concisión y simplicidad. Una opción alternativa sería estudiar la posibilidad de embeber el lenguaje de especificación de formularios propuesto en un lenguaje de programación orientado a objetos, como una librería de clases predefinidas, con lo que se podría mantener la concisión y simplicidad del lenguaje pero complementando su capacidad expresiva con la potencia de un lenguaje de programación.

No queremos acabar este trabajo sin señalar dos conclusiones genéricas, que esperamos se puedan inferir de este trabajo. La primera hace referencia a la presunta dicotomía que muchas veces se establece entre teoría y práctica, la segunda se refiere a la utilización de la filosofía en el estudio de problemas prácticos.

A pesar de que la forma en que se ha presentado este trabajo es básicamente teórica, queremos destacar su vocación fundamentalmente práctica. Tal como se ha señalado en la introducción, surgió de la práctica diaria de la programación de aplicaciones informáticas de empresa, como una manera de evitar la monótona repetición de la escritura de multitud de programas formalmente idénticos.

Aunque en algunos ámbitos profesionales los términos teórico y práctico son considerados antitéticos, no hay nada más práctico que una buena teoría. Un buen ejemplo de máxima aplicación práctica de una teoría es la axiomatización de la geometría que realizó Euclides. Independientemente de los defectos y carencias que pueda tener nuestra propuesta de axiomatización de formularios, creemos que en el diseño de sistemas de información transaccionales la vía axiomática merece ser explorada.

En cuanto a la utilización de la filosofía, en este trabajo se han tratado temas aparentemente muy alejados de los problemas de análisis y diseño de sistemas de información. Reivindicamos la necesidad de aclarar y distinguir las distintas nociones de información que reflejan fenómenos

diferentes, para así poder llegar a establecer una, o varias, nociones mínimamente consistentes. Creemos que proponer métodos prácticos que no estén fundamentados en nociones de información suficientemente sólidas y adecuadas al tipo de problema a resolver es, en algunos casos, edificar sobre arena, y puede llevar a hacer fracasar proyectos de empresa, con sus consiguientes e importantes repercusiones sociales y económicas.

Para concluir, con este trabajo no se pretende haber resuelto todos los problemas de validación, sin embargo, representa un nuevo enfoque y proporciona unos instrumentos que producen una aproximación más sistemática a la resolución del grave problema de malentender los requerimientos en el diseño de software de sistemas de información.

Una última referencia a la utilidad. Cuando Simon Stevin introdujo el sistema de numeración decimal en la Holanda del siglo XVI, declaró: *'It is not a great invention, but it is eminently useful to everyone'* (Farrington, 1944; p.309) En nuestro caso, pensamos que la vía de investigación que se propone en este trabajo 'no es una gran invención, pero creemos que puede ser útil'.

Bibliografía

- Adam, N.R. and Gangopadhyay, A. (1997). A Form-Based Natural Language Front-End to a CIM Database. *IEEE Transactions on Knowledge and Data Engineering* (9)2, pp.238-250.
- Agerfalk, P. and Eriksson, O. (2004). Action-oriented conceptual modelling. *European Journal of Information Systems* (13)1, pp.80-92.
- Aho, A.V., Sethi, R., and Ullman, J.D. (1986). *Compilers. Principles, Techniques, and Tools*. Addison-Wesley.
- Alter, S. (2000). Same Words, Different Meanings: Are Basic IS/IT Concepts our Self-Imposed Tower of Babel? *Communications of the Association for Information Systems* (3)10, pp.1-89.
- Alter, S. and Sherer, S.A. (2004). A General, but Readily Adaptable Model of Information Systems Risks. *Communications of the Association for Information Systems* (14)4, pp.1-28.
- Andersson, M. (1994). Extracting an Entity Relationship Schema from a Relational Database through Reverse Engineering. *13th International Conference on Entity-Relationship Approach*, Manchester, England.
- ANSI/X3/SPARC. (1975). Study Group on Data Base Management Systems, Interim Report. *FDT-Bulletin of ACM SIGMOD* (7)2, pp.1-140.
- Aristotle. (350a B.C.). De Interpretatione. In: Barnes, J., Editor. *The Complete Works of Aristotle. The Revised Oxford Translation. Volume One*. Ackrill, J. L., Translator. Princeton University Press 1995.
- Aristotle. (350b B.C.). Metaphysics. In: Barnes, J., Editor. *The Complete Works of Aristotle. The Revised Oxford Translation. Volume Two*. 6th ed. Ross, W. D., Translator. Princeton University Press 1995.

*Propuesta de un método de validación de esquemas conceptuales
y análisis comparativo de la noción de información
en los métodos de desarrollo de sistemas de información*

- Aristotle. (350c B.C.). Posterior Analytics. In: Barnes, J., Editor. *The Complete Works of Aristotle. The Revised Oxford Translation. Volume 1.* 6th ed. Barnes, J., Translator. Princeton University Press 1995.
- Aristotle. (350d B.C.). Topics. In: Barnes, J., Editor. *The Complete Works of Aristotle. The Revised Oxford Translation. Volume One.* 6th ed. Pickard-Cambridge, W. A., Translator. Princeton University Press 1995.
- Aristotle. (350e B.C.). On Interpretation. In: Edghill, E.M., Translator. *eBooks@Adelaide.* University of Adelaide Library Electronic Texts Collection 2004.
- Aristóteles. (350 B.C.). *Retórica.* Alianza Editorial, 1998.
- Ashenhurst, R.L. (1973). On the Problem of Characterizing Information Systems. *DATA BASE* (5)2, pp.148-156.
- Astley, W.G. and Zammuto, R.F. (1992). Organization Science, Managers, and Language Games. *Organization Science* (3)4, pp.443-466.
- Auramäki, E., Lehtinen, E., and Lyytinen, K. (1988). A Speech-Act-Based Office Modeling Approach. *ACM Transactions on Information Systems* (6)2, pp.126-152.
- Auramäki, E. and Lyytinen, K. (1996). On the success of speech acts and negotiating commitments. *International Workshop on Communication Modeling (LAP'96)*, Tilburg, The Netherlands.
- Austin, J.L. (1955). *How to Do Things with Words.* Harvard University Press, 1997.
- Avison, D.E. and Fitzgerald, G. (1995). *Information Systems Development: Methodologies, Techniques and Tools.* McGraw-Hill.
- Bagui, S. and Earp, R. (2003). *Database Design Using Entity-Relationship Diagrams.* Auerbach Publications.
- Bancilhon, F. and Spyratos, N. (1981). Update Semantics of Relational Views. *ACM Transactions on Database Systems* (6)4, pp.557-575.
- Barsalou, T., Siambela, N., Keller, A.M., and Wiederhold, G. (1991). Updating Relational Databases through Object-Based Views. *ACM SIGMOD RECORD* (20)2, pp.248-257.

- Beath, C.M. and Orlikowski, W.J. (1994). The Contradictory Structure of Systems Development Methodologies: Deconstructing the IS-User Relationship in Information Engineering. *Information Systems Research* (5)4, pp.350-377.
- Beauchamp, T.L. (1999). Editor's Introduction. In: Hume, D. *An Enquiry concerning Human Understanding*. Oxford University Press.
- Benbasat, I. and Zmud, R.W. (1999). Empirical Research in Information Systems: The Practice of Relevance. *MIS Quarterly* (23)1, pp.3-16.
- Berkeley, G. (1710). *Principles of Human Knowledge and Three Dialogues*. Oxford University Press, 1999.
- Berners-Lee, T., Hendler, J., and Lassila, O. (2001). The Semantic Web. *Scientific American* (284)5, pp.34-43 .
- Bittner, K. and Spence, I. (2003). *Use Case Modeling*. Addison-Wesley.
- Blackburn, S. and Simmons, K. (1999). *Truth*. Oxford University Press.
- Bodart, F., Patel, A., Sim, M., and Weber, R. (2001). Should Optional Properties Be Used in Conceptual Modelling? A Theory and Three Empirical Tests. *Information Systems Research* (12)4, pp.384-405.
- Boehm, B. (1984). Verifying and Validating Software Requirements and Design Specifications. *IEEE Software* (1)1, pp.75-88.
- Boisot, M. and A. Canals. (2004). Data, Information and Knowledge: Have We Got It Right? Univesitat Oberta de Catalunya, IN3 Internet Interdisciplinary Institute; Working Paper Series WP04-02.
- Boland Jr., R.J. (1985). Phenomenology: A Preferred Approach to Research in Information Systems. In: Mumford, E., Hirschheim, R., Fitzgerald, G., and Wood-Harper, T., Editors. *Research Methods in Information Systems. Proceedings of the IFIP WG8.2 Colloquium*, Manchester, UK. pp.193-201.
- Boland Jr., R.J. (1991). Information Systems Use as a Hermeneutic Process. In: Nissen, H.-E., Klein, H.K., and Hirschheim, R., Editors. *Information Systems Research: Contemporary Approaches & Emergent Traditions. Proceedings of the IFIP TC8/WG8.2 Working Conference*, Copenhagen, Denmark. pp.439-458.
- Boland Jr., R.J. and Day, W. (1982). The Process of System Design: A

*Propuesta de un método de validación de esquemas conceptuales
y análisis comparativo de la noción de información
en los métodos de desarrollo de sistemas de información*

- Phenomenological Approach. In: Ginzberg, M. and Ross, C., Editors. *3rd International Conference on Information Systems*, Ann Arbor, Michigan.
- Booch, G. (1994). *Object-Oriented Analysis and Design*. Addison-Wesley.
- Booch, G., Rumbaugh, J.E., and Jacobson, I. (1999). *The Unified Modeling Language User Guide*. Addison-Wesley, 2000.
- Bowen, J.P. and Hinchey, M.G. (1995). Seven More Myths of Formal Methods. *IEEE Software* (12)4, pp.34-41.
- Brooks, F.P. (1986). *The Mythical Man-Month: Essays on Software Engineering*. Addison-Wesley, 1995.
- Bubenko, J.A. (1977). Validation and Verification Aspects of Information Modeling. *Very Large Data Base Conference*.
- Bunge, M. (1973). *Philosophy of Physics*. Filosofía De La Física. D. Reidel Publishing Company, 1982.
- Burrell, G. and Morgan, G. (1979). *Sociological Paradigms and Organisational Analysis*. Ashgate, 1998.
- Bussler, C., Fensel, D., and Maedche, A. (2002). A Conceptual Architecture for Semantic Web Enabled Web Services. *ACM SIGMOD RECORD* (31)4, pp.24-29.
- Butler, T. (2000). Making Sense of Knowledge: A Constructivist Viewpoint. *Americas Conference on Information Systems*, Long Beach, California. pp.1462-1466
- Carnap, R. (1956). *Meaning and Necessity. A Study in Semantics and Modal Logic*. The University of Chicago Press, 1988.
- Chae, B. and Courtney, J.F. (2000). Hegelian Inquiring Organizations: Guiding Principles, Design Guidelines, and IT Support. *Americas Conference on Information Systems*, Long Beach, California. pp.1497-1502.
- Chaitin, G.J. (1975). Randomness and Mathematical Proof. *Scientific American* (232)5, pp.47-52.
- Chaitin, G.J. (1982). Gödel's Theorem and Information. *International Journal of Theoretical Physics* (22), pp.941-954.

- Chaitin, G.J. (1999). *The Unknowable*. Available at: www.cs.auckland.ac.nz/CDMTCS/chaitin. Last accessed: 2004
- Chalmers, D.J. (1995). The Puzzle of Conscious Experience. *Scientific American* (273)6, pp.80-86.
- Chandrasekaran, B., Josephson, J.R., and Benjamins, R. (1999). What Are Ontologies, and Why Do We Need Them? *IEEE Intelligent Systems* (14)1, pp.20-26.
- Chang, L. and Lin, T. (2000). Political Games of Users and MIS Professionals in the Information Systems Development Process. *Americas Conference on Information Systems*, Long Beach, California. pp.1474-1482.
- Chen, I.-M.A. and McLeod, D. (1989). Derived Data Update in Semantic Databases. *15th VLDB Conference*, Amsterdam. pp.225-235.
- Chen, P.P.-S. (1976). The Entity-Relationship Model --Toward a Unified View of Data. *ACM Transactions on Database Systems* (1)1, pp.9-36.
- Chen, P.P.-S. (1977). The Entity-Relationship Model --A basis for the Enterprise View of Data. *National Computer Conference*, pp.77-84.
- Choobineh, J. and Lo, A.W. (2004). CABSYYDD: Case-Based System for Database Design. *Journal of Management Information Systems* (21)3, pp.281-314.
- Choobineh, J., Mannino, M.V., Nunamaker, J.F.JR. , and Konsynski, B.R. (1988). An Expert Database Design System Based on Analysis of Forms. *IEEE Transactions on Software Engineering* (14)2, pp.242-253.
- Choobineh, J., Mannino, M.V., and Tseng, V.P. (1992). A Form-Based Approach for Database Analysis and Design. *Communications of the ACM* (35)2, pp.108-120.
- Chua, W.F. (1986). Radical Developments in Accounting Thought. *The Accounting Review* (61)4, pp.601-632.
- Churchland, P.M. and Churchland, P.S. (1990). Could a machine think? *Scientific American* (262)1, pp.32-37.
- Coad, P. and Yourdon, E. (1991). *Object-Oriented Analysis*. Prentice-Hall.
- Codd, E.F. (1970). A Relational Model of Data for Large Shared Data Banks.

- Communications of the ACM* (13)6, pp.377-387.
- Codd, Edgar F. (1971). Further Normalization of the Data Base Relational Model. IBM Research Report; RJ909.
- Codd, E.F. (1979). Extending the Database Relational Model to Capture More Meaning. *ACM Transactions on Database Systems* (4)4, pp.397-434.
- Codd, E.F. (1981). Data Models in Database Management. *ACM SIGMOD RECORD* (11)2, pp.112-114.
- Codd, E.F. (1982). Relational Database: A Practical Foundation for Productivity. *Communications of the ACM* (25)2.
- Codd, E.F. (1990). *The Relational Model for Data Base Management: Version 2*. Addison Wesley, 1990.
- Coffa, J.A. (1991). *The Semantic Tradition from Kant to Carnap, To the Vienna Station*. Wessels, L., Editor. Cambridge University Press, 1998.
- Connolly, T. and Begg, C. (2005). *Database Systems. A Practical Approach to Design, Implementation, and Management*. Addison-Wesley, 2005.
- Crawshay-Williams, R. (2003). Bertrand Russell. In: *Encyclopaedia Britannica*. 2003.
- Curtis, B., Krasner, H., and Iscoe, N. (1988). A Field Study of the Software Design Process for Large Systems. *Communications of the ACM* (31)11, pp.1268-1287.
- Damsgaard, J. and Truex, D.P. (2000). Binary trading relations and the limits of EDI standards: the Procrustean bed of standards. *European Journal of Information Systems* (9)3, pp.173-188.
- Date, C.J. (1982). A Formal Definition of the Relational Model. *ACM SIGMOD RECORD* (13)1, pp.18-29.
- Date, C.J. (1983). *An Introduction to Database Systems. Volume II*. 3rd ed. Addison-Wesley, 1983.
- Date, C.J. (1995). Updating Union, Intersection, and Difference Views. In. *Relational Database Writings 1991-1994*. Addison-Wesley 1995.
- Davenport, T.H., Eccles, R.G., and Prusak, L. (1992). Information Politics.

- Sloan Management Review* , pp.53-65.
- Davis, A.M. (1993). *Software Requirements: Objects, Functions, and States*. Prentice-Hall.
- Davis, G.B. and Olson, M. (1985). *Management Information Systems. Conceptual Foundations, Methods and Development*. New York McGraw-Hill.
- Deetz, S. (1996). Describing Differences in Approaches to Organization Science: Rethinking Burrell and Morgan and Their Legacy. *Organization Science* (7)2, pp.191-207.
- Defense Information Systems Agency. Center for Standards. (1996). *Department of Defense Technical Architecture Framework for Information Management. Volume 1: Overview. Version 3.0*.
- Dell, M. and Fredman, C. (1999). *Direct from Dell*. HarperCollins Business.
- Dennett, D.C. (1987). *La actitud intencional*. Editorial Gedisa, 1998.
- Denning, P.J. (1995). Can There Be a Science of Information? *ACM Computing Surveys* (27)1, pp.23-25.
- Dietz, J. (2001). DEMO: Towards a discipline of organisation engineering. *European Journal of Operational Research* (128)2, pp.351-363.
- Dilthey, W. (1868). On Understanding and Hermeneutics: Student Lecture Notes (1867-68). Makkreel, R.A. and Rodi, F., Editors. *Selected Works - Volume IV. Hermeneutics and the Study of History*. Makkreel, R. A., Translator Princeton University Press 1996. pp. 229-234.
- Dilthey, W. (1900). The Rise of Hermeneutics. In: Makkreel, R.A. and Rodi, F., Editors. *Selected Works - Volume IV. Hermeneutics and the Study of History*. Princeton University Press 1996.
- Dilthey, W. (1860). Schleiermacher's Hermeneutical System in Relation to Earlier Protestant Hermeneutics. In: Makkreel, R.A. and Rodi, F., Editors. *Selected Works - Volume IV. Hermeneutics and the Study of History*. Princeton University Press 1996.
- Doan, A., Doy, N.F., and Halevy, A.Y. (2004). Introduction to the Special Issue on Semantic Integration. *ACM SIGMOD RECORD* (33)4, pp.11-13.
- DRAE. *Diccionario de la lengua española*. Espasa Calpe, S.A., 2003.

*Propuesta de un método de validación de esquemas conceptuales
y análisis comparativo de la noción de información
en los métodos de desarrollo de sistemas de información*

- Einstein, A. (1922). *Sidelights on Relativity*. New York Dover Publications, 1983.
- El-Masri, R. and Wiederhold, G. (1979). Data Model Integration Using the Structural Model. *ACM SIGMOD International Conference on Management of Data*, Boston, Massachusetts. pp.191-202.
- Embley, D.W. (1989). NFQL: The Natural Forms Query Language. *ACM Transactions on Database Systems* (14)2, pp.168-211.
- Euclid. (300 B.C.). *The Thirteen Books of the Elements. Vol. 1 (Books I and II)*. Dover Publications, Inc. 1956.
- Euromethod Project. (1996). *Euromethod. Version 1*. www.fast.de/Euromethod
- Falkenberg, E. D., W. Hesse, P. Lindgreen, B. Nilsson, J. L. H. Oei, C. Rolland, R. Stamper, F. J. M. V. Assche, A. A. Verrijn-Stuart, and K. Voss. (1998). A Framework of Information System Concepts (The FRISCO Report). IFIP
- Farrington, B. (1944). *Greek Science. Its Meaning For Us*. Spokesman, 2000.
- Feynman, R.P. (1996). *Feynman Lectures on Computation*. Penguin Books, 1999.
- Flores, F. (1994). *Creando Organizaciones para el Futuro*. Dolmen Ediciones.
- Flores, F., Graves, M., Hartfield, B., and Winograd, T. (1988). Computer Systems and the Design of Organizational Interaction. *ACM Transactions on Information Systems* (6)2, pp.153-172.
- Flores, F. and Winograd, T. (1989). *Understanding Computers and Cognition*. Ablex Publishing Corporation.
- Frege, G. (1918). The Thought: A Logical Inquiry. In: Blackburn, S. and Simmons, K., Editors. *Truth*. Oxford University Press 1999. pp. 85-105.
- Frege, G. (1892). On Sense and Reference. In: Richard, M., Editor. *Meaning*. Blackwell 2003.
- Gadamer, H.-G. (1960). *Wahrheit und Methode*. Verdad y Método I. 7ª ed. Ediciones Sígueme, 1997.
- Gadamer, H.-G. (1986). *Wahrheit und Methode. Ergänzungen - Register*. Verdad y Método II. 3ª ed. Ediciones Sígueme, 1998.

- Gell-Mann, M. (1994). *The Quark and the Jaguar. Adventures in the Simple and the Complex*. Abacus, 1995.
- Gibbs, W.W. (1994). La crisis crónica de la programación. *Investigación y Ciencia*. Noviembre, pp.72-81.
- Giddens, A. (1976). *New Rules of Sociological Method*. Stanford University Press, 1993.
- Giddens, A. (1984). *The Constitution of Society: Outline of the Theory of Structure*. Berkeley, CA University of California Press, 2001.
- Glass, R.L. (1996). Formal Methods are a Surrogate for a More Serious Software Concern. *IEEE Computer* (29)4, p.19.
- Goguen, J.A. (1994). Requirements Engineering as the Reconciliation of Technical and Social Issues. Jirotko, M. and Goguen, J.A., Eds. *Requirements Engineering: Social and Technical Issues*. Academic Press 1994. pp. 165-200.
- Goguen, J.A. (1996). Formality and Informality in Requirements Engineering. *2nd International Conference on Requirements Engineering*, Colorado Springs, Colorado.
- Goguen, J.A. (1997). Towards a Social, Ethical Theory of Information. In: Bowker, G., Gasser, L., Star, L., and Turner, W., Editors. *Social Science Research, Technical Systems and Cooperative Work: Beyond the Great Divide*. Erlbaum 1997. pp. 27-56.
- Goguen, J.A. (2002). Consciousness and the Decline of Cognitivism. *Distributed Collective Practices*, UCSD, San Diego, California.
- Goldkuhl, G. (1996). Generic Business Frameworks and Action Modelling. *International Workshop on Communication Modeling (LAP'96)*, Tilburg, The Netherlands.
- Goldkuhl, G. and Lyytinen, K. (1982). A Language Action View of Information Systems. In: Ginzberg, M. and Ross, C., Editors. *3rd International Conference on Information Systems*. Ann Arbor, Michigan.
- Goodhue, D. and Wybo, M.D. (1992). The impact of data integration on the costs and benefits of information systems. *MIS Quarterly* (16)3, pp.293-311.

- Gopal, A. and Prasad, P. (2000). Understanding GDSS in Symbolic Context: Shifting the Focus from Technology to Interaction. *MIS Quarterly* (24)3, pp.509-546.
- Grice, P. (1989). *Studies in the Way of Words*. Harvard University Press, 1991.
- Griethuysen, J. J. van. (1982). Editor. Concepts and Terminology for the Conceptual Schema and the Information Base. ISO Report, ISO/TC97/SC5/N695.
- Gruber, T.R. (1993). A Translation Approach to Portable Ontology Specifications. *Knowledge Acquisition* (5)2, pp.199-220.
- Gruber, T.R. (1995). Toward Principles for the Design of Ontologies Used for Knowledge Sharing. *International Journal Human-Computer Studies* (43)5, pp.907-928.
- Guarino, N. (1995). Formal Ontology, Conceptual Analysis and Knowledge Representation. *International Journal Human-Computer Studies* (43)5, pp.625-640.
- Guarino, N. (1998). Formal Ontology and Information Systems. *International Conference on Formal Ontology in Information Systems*, Trento, Italy. pp.3-15.
- Guarino, N. and Welty, C. (2000). Ontological Analysis of Taxonomic Relationships. Laender, A. and Storey, V.C., eds. *19th International Conference on Conceptual Modeling*, Salt Lake City, Utah.
- Guarino, N. and Welty, C. (2002). Evaluating Ontological Decisions with OntoClean. *Communications of the ACM* (45)2, pp.61-65.
- Guarino, N. and Welty, C. (2004). An Overview of OntoClean. Staab, S. and Studer, R., Editors. *Handbook on Ontologies*. Springer Verlag 2004. pp. 151-159.
- Guizzardi, G., Herre, H., and Wagner, G. (2002). On the General Ontological Foundations of Conceptual Modeling. Spaccapietra, S., March, S.T., and Kambayashi, Y., eds. *21st International Conference on Conceptual Modeling*. Tampere, Finland. pp.65-78
- Habermas, J. (1981). *The Theory of Communicative Action, Vol.1, Reason and the Rationalization of Society*. McCarthy, T., Translator. Beacon Press, 1984.
- Hamilton, G.A. (1988). *Logic for Mathematicians*. Cambridge University Press

- Paraninfo, 2000.
- Hammer, M. (1990). Reengineering Work: Don't Automate, Obliterate. *Harvard Business Review* (68)4, pp.104-112.
- Hammer, M. and Berkowitz, B. (1980). DIAL: A Programming Language for Data Intensive Applications. *ACM SIGMOD International Conference on Management of Data*, Santa Monica, California. pp.75-92.
- Hammer, M. and Champy, J. (1993). *Reengineering the Corporation: A Manifesto for Business Revolution*. HarperCollins.
- Hammer, M., Howe, W.G., Kruskal, V.J., and Wladawsky, I. (1977). A Very High Level Programming Language for Data Processing Applications. *Communications of the ACM* (20)11, pp.832-840.
- Hawking, S. (1997). The Objections of an Unashamed Reductionist. In: Penrose, R. *The Large, the Small and the Human Mind*. Cambridge University Press 1997.
- Hevner, A.R., March, S.T., Park, J., and Ram, S. (2004). Design Science in Information Systems Research. *MIS Quarterly* (28)1, pp.75-106.
- Hirschheim, R., Iivari, J., and Klein, H.K. (1997). A Comparison of Five Alternative Approaches to Information Systems Development. *Australian Journal of Information Systems* (5)1.
- Hirschheim, R. and Klein, H.K. (1989). Four Paradigms of Information Systems Development. *Communications of the ACM* (32)10, pp.1199-1216.
- Hirschheim, R., Klein, H.K., and Lyytinen, K. (1995). *Information Systems Development and Data Modeling: Conceptual and Philosophical Foundations*. Cambridge University Press, 1995.
- Hume, D. (1740). *A Treatise of Human Nature*. Penguin Books, 1985.
- Hume, D. (1758). *An Enquiry concerning Human Understanding*. Oxford University Press, 1999.
- IEEE. (1990). Standard Glossary of Software Engineering Terminology. Standards Coordinating Committee of the IEEE Computer Society; ANSI/IEEE Std 610.12-1998.
- . (1999). IEEE Standard for Conceptual Modeling Language Syntax and

*Propuesta de un método de validación de esquemas conceptuales
y análisis comparativo de la noción de información
en los métodos de desarrollo de sistemas de información*

- Semantics for IDEF1X97 (IDEFobject). Software Engineering Standards Committee of the IEEE Computer Society; IEEE Std 1320.2-1998.
- Iivari, J. (1986). Dimensions of Information Systems Design: A Framework for a Long-Range Research Program. *Information Systems* (11)2, pp.185-197.
- Iivari, J. (1991). A Paradigmatic Analysis of Contemporary Schools of IS Development. *European Journal of Information Systems* (1)4, pp.249-272.
- Iivari, J. and Hirschheim, R. (1996). Analyzing Information Systems Development: A Comparison and Analysis of Eight IS Development Approaches. *Information Systems* (21)7, pp.551-575.
- Iivari, J., Hirschheim, R., and Klein, H.K. (1998). A Paradigmatic Analysis Contrasting Information Systems Development Approaches and Methodologies. *Information Systems Research* (9)2, pp.164-193.
- Iivari, J., Hirschheim, R., and Klein, H.K. (1999). Beyond Methodologies: Keeping up with Information Systems Development Approaches through Dynamic Classification. *32nd Hawaii International Conference on Systems Sciences*,
- Iivari, J., Hirschheim, R., and Klein, H.K. (2000). A Dynamic Framework for Classifying Information Systems Development Methodologies and Approaches. *Journal of Management Information Systems* (17)3, pp.179-218.
- Jackson, M. (1995). The World and the Machine. *17th International Conference on Software Engineering*, Seattle.
- Jacobson, I., Christerson, M., Jonsson, P., and Övergaard, G. (1992). *Object-Oriented Software Engineering: A Use Case Driven Approach*. Addison-Wesley, 1995.
- Jaspersen, J., Carte, T.A., Saunders, C.S., Butler, B.S., Croes, H., and Zheng, W. (2002). Review: Power and Information Technology Research: A Metatriangulation Review. *MIS Quarterly* (26)4, pp.397-459.
- Jesitus, J. (1997). Broken Promises? FoxMeyer's Project was a Disaster. *Industry Week/IW* (246)20, pp.31-36.
- Johannesson, P. (1995). Representation and Communication -A Speech Act Based Approach to Information Systems Design. *Information Systems* (20)4, pp.291-303.

- Johnson, B.C. (2002). Retail: The Wal-Mart Effect. *McKinsey Quarterly* 1, pp.40-43.
- Kant, I. (1781a). *The Critique of Pure Reason*. Meiklejohn, J.M.D., Translator. Prometheus Books, 1990.
- Kant, I. (1781b). *Crítica de la razón pura*. Editorial Losada, 2003.
- Keil, M., Cule, P.E., Lyytinen, K., and Schmidt, R.C. (1998). A Framework for Identifying Software Project Risks. *Communications of the ACM* (41)11, pp.76-83 .
- Keller, A.M. and Ullman, J.D. (1984). On Complementary and Independent Mappings on Databases. *Proceedings of the 1984 ACM SIGMOD International Conference on Management of Data*. Boston, Massachusetts. pp.143-148
- Kernighan, B.W. and Pike, R. (1987). *El entorno de programación UNIX*. Prentice Hall.
- Kettinger, W.J. and Choong, C.L. (2002). Understanding the IS-User Divide in IT Innovation. *Communications of the ACM* (45)2, pp.79-84.
- Kim, D.H. and Park, S.J. (1997). FORM: A flexible data model for integrated CASE environments. *Data & Knowledge Engineering* (22)2, pp.133-158.
- Kim, Y.-G. and March, S.T. (1995). Comparing Data Modeling Formalisms. *Communications of the ACM* (38)6, pp.103-115.
- Kishore, R., Sharman, R., and Ramesh, R. (2004a). Computational Ontologies and Information Systems: I. Foundations. *Communications of the Association for Information Systems* (14), pp.158-183.
- Kishore, R., Zhang, H., and Ramesh, R. (2004b). A Helix-Spindle Model for Ontological Engineering. *Communications of the ACM* (47)2, pp.69-75.
- Klein, H.K. and Hirschheim, R. (1987). A Comparative Framework of Data Modelling Paradigms and Approaches. *The Computer Journal* (30)1, pp.8-15.
- Klein, H.K. and Myers, M.D. (1999). A Set of Principles for Conducting and Evaluating Interpretive Field Studies in Information Systems. *MIS Quarterly* (23)1, pp.67-94.
- Kling, R. (1980). Social Analyses of Computing: Theoretical Perspectives in Recent Empirical Research. *ACM Computing Surveys* (12)1, pp.61-110.

- Konicki, S. (2002). Now In Bankruptcy, Kmart Struggled With Supply Chain. *InformationWeek* 873, p.26.
- Kotler, P. (1974). *Dirección de mercadotecnia. Análisis, planeación y control*. Editorial Diana.
- Kuhn, T.S. (1962). *The Structure of Scientific Revolutions*. La Estructura De Las Revoluciones Científicas. Contín, A., Traductor. Fondo de Cultura Económica, 1971.
- Lakoff, G. (1987). *Women, Fire, and Dangerous Things. What Categories Reveal about the Mind*. The University of Chicago Press, 1990.
- Lamsweerde, A.v. (2000). Formal Specification: A Roadmap. *International Conference on Software Engineering*, Limerick, Ireland. pp.147-159.
- Latour, B. (1987). *Science in Action*. Harvard University Press, 2002.
- Lechtenbörger, J. (2003). The Impact of the Constant Complement Approach Towards View Updating. *22nd ACM SIGMOD-SIGACT-SIGART Symposium on Principles of Database Systems*, San Diego, CA.
- Lee, A.S. (1994). Electronic Mail as a Medium for Rich Communication: An Empirical Investigation Using Hermeneutic Interpretation. *MIS Quarterly* (18)2, pp.143-157.
- Lehtinen, E. and Lyytinen, K. (1994). Action Based Model of Information System. *Information Systems* (11)4, pp.299-317.
- Leifer, R., Lee, S., and Durgee, J. (1994). Deep Structures: Real Information Requeriments Determination. *Information & Management* (17)5, pp.275-285.
- Levine, J.R., Mason, T., and Brown, D. (1995). *Lex & Yacc*. O'Reilly.
- Lewis, D. (1969). *Convention. A Philosophical Study*. Blackwell, 2002.
- Li, M. and Vitányi, P. (1993). *An Introduction to Kolmogorov Complexity and its Applications*. Springer-Verlag.
- Locke, J. (1690). *An Essay Concerning Human Understanding*. Penguin Books, 2004.
- Loucopoulos, P. and Karakostas, V. (1995). *System Requirements Engineering*. London McGraw-Hill.

- Luqi and Goguen, J.A. (1997). Formal Methods: Promises and Problems. *IEEE Software* (14)1, pp.73-85.
- Lyytinen, K. (1985). Implications of Theories of Language for Information Systems. *MIS Quarterly* (9)1, pp.61-74.
- Lyytinen, K. (2003). Information Systems and Philosophy: The Hopeless Search for Ultimate Foundations. *Americas Conference on Information Systems*, Tampa, Florida. pp.2710.
- Madnick, S. (2001). The Misguided Silver Bullet: What XML will and will NOT do to help Information Integration. *MIT Sloan Working Paper* (No. 4185-01).
- Mahfouz, A. and Paradice, D. (2000). Kantian Inquiring Systems: An Illustration of a Reail Organization. *Americas Conference on Information Systems*, Long Beach, California. pp.1503-1506.
- Mannino, M.V., Choobineh, J., and Hwang, J.J. (1986). Acquisition and Use of Contextual Knowledge in a Form-Driven Database Design Methodology. Spaccapietra, S., Ed. *5th International Conference on Entity-Relationship Approach*, Dijon, France. pp.361-377.
- Markowitz, V.M. and Shoshani, A. (1992). Representing Extended Entity-Relationship Structures in Relational Databases: A Modular Approach. *ACM Transactions on Database Systems* (17)3, pp.423-464.
- Markus, M.L. (1983). Power, Politics, and MIS Implementation. *Communications of the ACM* (26)6, pp.430-444.
- McConnell, S. (2000). What's in a Name? *IEEE Software* (17)5, pp.7-9.
- Medina-Mora, R., Winograd, T., Flores, R., and Flores, F. (1992). The Action Workflow Approach to Workflow Management Technology. *ACM CSCW*, Toronto. pp.281-288.
- Merriam-Webster. (2000). *Merriam-Webster Collegiate Dictionary*. Merriam-Webster Incorporated.
- Mfourga, N. (1997). Extracting Entity-Relationship Schemas from Relational Databases: A Form-Driven Approach. Baxter, I.D., Quilici, A., and Verhoef, C., Eds. *4th Working Conference on Reverse Engineering*, Amsterdam, The Netherlands. pp.184-193.

- Mingers, J. (2004). Re-establishing the Real: Critical Realism and Information Systems. In: Mingers, J. and Willcocks, L., Editors. *Social Theory and Philosophy for Information Systems*. John Wiley & Sons 2004. pp. 372-406.
- . (2005). Information, Knowledge and Truth: A Polyvalent View. Kent Business School; Working Paper No. 77.
- Monod, E. (2002). For a Kantian Foundation of IS Research: Proposals for an Epistemological Pluralism. *Americas Conference on Information Systems*, Dallas, Texas. pp.1751-1759 .
- Monod, E. (2003). A Copernican Revolution in IS: Using Kant's Critique of Pure Reason for Describing Epistemological Trends in IS. *Americas Conference on Information Systems*, Tampa, Florida. pp.2719-2724.
- Monod, E. (2004). Einstein, Heisenberg, Kant: Methodological Distinction and Conditions of Possibilities. *Information and Organization* (14)2, pp.105-121.
- Murray, F. (1989). The organizational politics of information technology. *Technology Analysis and Strategic Management* , pp.285-298.
- Myers, M.D. (2004). Hermeneutics in Information Systems Research. In: Mingers, J. and Willcocks, L., Editors. *Social Theory and Philosophy for Information Systems*. John Wiley & Sons 2004. pp. 103-128.
- Nagel, E. and Newman, J.R. (1958). *Gödel's Proof*. New York University Press, 1986.
- Nannery, M. (2002). Use IT or Lose It. *Chain Store Age* (94)3, pp.12.
- Navathe, S.B. (1992). Evolution of Data Modeling for Databases. *Communications of the ACM* (35)9, pp.112-123.
- Nietzsche, F. (1873). *Sobre Verdad y Mentira en Sentido Extramoral*. Valdés, L.M. and Orduña, T., Traductores. Tecnos, 1990.
- Nietzsche, F. (1872). *Darstellung der antiken Rhetorik*. Escritos Sobre Retórica. Santiago Guervós, L.E.d., traductor. Editorial Trotta, S.A., 2000.
- NIST. (1993). Integration Definition for Information Modeling (IDEF1X). U.S. National Institute of Standards and Technology ; Federal Information Processing Standards Publication 184 (FIPS-184).
- Orlikowski, W.J. and Barley, S.R. (2001). Technology and Institutions: What

- Can Research on Information Technology and Research on Organizations Learn from Each Other? *MIS Quarterly* (25)2, pp.145-165.
- Orlikowski, W.J. and Baroudi, J.J. (1991). Studying Information Technology in Organizations: Research Approaches and Assumptions. *Information Systems Research* (2)1, pp.1-28.
- Osterland, A. (2000). Blaming ERP. *CFO Magazine*. January, 1.
- Peaucelle, J.-L. (1981). *Les systemes d'information, la représentation*. Presses Universitaires de France, 1981.
- Peirce, C.S. (1894). What is a Sign? In. *The Essential Peirce. Selected Philosophical Writings. Volume 2 (1983-1913)*. Indiana University Press 1998.
- Pettigrew, A.M. (1972). Information Control as a Power Resource. *Sociology*(6)2, pp.187-204.
- Popper, K.R. (1963). *Conjectures and Refutations. The Growth of Scientific Knowledge*. 4th ed. Conjeturas y Refutaciones. El Desarrollo Del Conocimiento Científico. Míguez, N., Translator. Paidós Básica, 1994.
- Popper, K.R. (1974). *Unended Quest. An Intellectual Autobiography*. Fontana/Collins, 1977.
- Prahalad, C.K. and Krishnan, M.S. (1999). The New Meaning of Quality in the Information Age. *Harvard Business Review*(77)5. pp 109-118 .
- Putnam, H. (1988). *Representation and Reality*. MIT Press, 1991.
- Putnam, H. (1981). *Reason, Truth and History*. Cambridge University Press, 1997.
- Quine, W.V.O. (1953). *From a Logical Point of View. Nine Logico-Philosophical Essays*. Harvard University Press, 1999.
- Quine, W.V.O. (1960). *Word & Object*. The MIT Press, 1997.
- Quine, W.V.O. (1966). *The Ways of Paradox and other essays*. Harvard University Press, 1997.
- Quine, W.V.O. (1969). *Ontological Relativity & Other Essays*. Columbia University Press.

- Quine, W.V.O. (1981). *Mathematical Logic*. Harvard University Press, 1996.
- Quine, W.V.O. (1990). *Pursuit of Truth*. Harvard University Press, 1996.
- Quine, W.V.O. (1995). *From Stimulus to Science*. Harvard University Press, 1998.
- Quinn, J.B., Baruch, J.J., and Zien, K.A. (1996). Software-Based Innovation. *Sloan Management Review* (37)4, pp.11-24.
- Raghavan, S., G. Zelesnik, and G. Ford. (1994). Lecture Notes on Requirements Elicitation. Carnegie Mellon University; CMU/SEI-94-EM-10.
- Rahm, E. and Berstein, P.A. (2001). A survey of approaches to automatic schema matching. *VLDB Journal* (10)4, pp.334-350.
- Reale, G. and Antiseri, D. (1983). *Historia del pensamiento filosófico y científico III. Del romanticismo hasta hoy*. Herder, 2002.
- Richard, M. (2003). *Meaning*. Blackwell.
- Ridjanovic, D., Jarvenpaa, S.L., Mantha, R., Ram, S., and Rosenthal, A. (1987). ER Versus Relational: What are the Differences? March, S.T. *6th International Conference on Entity-Relationship Approach*, New York.
- Rorty, R. (1967). *The Linguistic Turn: Recent Essays in Philosophical Method*. El Giro Lingüístico. Bello, G., Traductor. The University of Chicago Press, 1998.
- Rosemann, M., Vessey, I., Weber, R., and Wyssusek, B. (2004). On the Applicability of the Bunge-Wand-Weber Ontology to Enterprise Systems Requirements. *15th Australasian Conference on Information Systems*, Hobart, Tasmania.
- Rouff, C. (1996). Formal Specification of User Interfaces. *SIGCHI Bulletin* (28)3.
- Rowe, L.A. (1985). Fill-in-the-Form Programming. *Very Large Data Base Conference*, Stockholm, Sweden.
- Rowe, L.A. and Shoens, K.A. (1979). Data Abstraction, Views and Updates in RIGEL. *ACM SIGMOD International Conference on Management of Data*, Boston, Massachusetts.

- Rowe, L.A. and Shoens, K.A. (1982). A Form Application Development System. *ACM SIGMOD Conference*, Orlando, Florida. pp.28-39.
- Rowe, L.A. and Shoens, K.A. (1983). Programming Language Constructs for Screen Definition. *IEEE Transactions on Software Engineering* (9)1, pp.31-39.
- Rumbaugh, J.E., Blaha, M.R., Premerlani, W.J., Eddy, F., and Lorenzen, W. (1991). *Object-Oriented Modeling and Design*. Prentice-Hall .
- Russell, B. (1961). *The Basic Writings of Bertrand Russell 1903 1959*. Bertrand Russell Escritos Básicos I. George Allen & Unwin Ltd., 1985.
- Russell, B. (1950). *An Inquiry into Meaning and Truth*. Unwin Paperbacks, 1980.
- Russell, B. (1912). *The Problems of Philosophy*. Oxford University Press, 1998.
- Russell, B. (1908). William James's Conception of Truth. In: Blackburn, S. and Simmons, K., Editors. *Truth*. Oxford University Press 1999 . pp. 69-82.
- Saiedian, H. (1996). An Invitation to Formal Methods. *IEEE Computer* (29)4, pp.16-17.
- Santesmases Mestre, M. (1996). *Marketing. Conceptos y estrategias*. Ediciones Pirámide.
- Saunders, C.S. and Scamell, R.W. (1986). Organizational Power and the Information Services Department: A Reexamination. *Communications of the ACM* (29)2, pp.142-147.
- Saussure, F.d. (1915). *Course in General Linguistics*. McGraw_Hill Education, 1966.
- Scharer, L. (1981). Pinpointing Requirements. In: Thayer, R.H. and Dorfman, M., Editors. *Software Requirements Engineering*, 2nd ed. IEEE Computer Society Press 2000.
- Scheer, A.-W. and Hars, A. (1992). Extending Data Modeling to Cover the Whole Enterprise. *Communications of the ACM* (35)9, pp.166-172.
- Schein, E.H. (1996). Three Cultures of Management: The Key to Organizational Learning. *Sloan Management Review* (38)1, pp.9-20.
- Schein, E.H. (1992) Management and Information Technology. Two Subcultures in Collision? In: *Organizational Culture and Leadership*. 2 ed.

- Jossey-Bass, 1997.
- Schein, E.H. (1992). *Organizational Culture and Leadership*. Jossey-Bass Publishers, 1997.
- Schleiermacher, F. (1828). *Hermeneutics and Criticism, And Other Writings*. Bowie, A., Translator and Editor. Cambridge University Press, 1998.
- Schmidt, R.C., Lyytinen, K., Keil, M., and Cule, P.E. (2001). Identifying Software Project Risks: An International Delphi Study. *Journal of Management Information Systems* (17)4, pp.5-36.
- Schöning, U. (1989). *Logic for Computer Scientists*. Birkhäuser.
- Schoop, M. (2001). An Introduction to the Language-Action Perspective. *ACM SIGGROUP Bulletin* (22)2.
- Scott, J.E. The FoxMeyer Drugs' Bankruptcy: Was it a Failure of ERP? *Proceedings of The Association for Information Systems Fifth Americas Conference on Information Systems*, Milwaukee.
- Scott, J.E. and Vessey, I. (2002). Managing Risks in Enterprise Systems Implementations. *Communications of the ACM* (45)4, pp.74-81.
- Searle, J.R. (1969). *Speech Acts. An Essay in the Philosophy of Language*. Cambridge University Press, 1997.
- Searle, J.R. (1979). *Expression and meaning. Studies in the theory of speech acts*. Cambridge University Press, 1997.
- Searle, J.R. (1990). Is the brain's mind a computer program? *Scientific American* (262)1, pp.26-31.
- Searle, J.R. (1992). Rationality and Realism: What is at Stake? *Daedalus* (122)4, pp.55-84.
- Searle, J.R. (1995). *The Construction of Social Reality*. The Free Press, 1995.
- Sesé, F., Wareham, J., and Bonet, E. (2006). Words and Objects in Information Systems Development: Six Paradigms of Information as Representation. *14th European Conference on Information Systems*, Göteborg, Sweden.
- Shanks, G., Tansley, E., and Weber, R. (2003). Using Ontology to Validate

- Conceptual Models. *Communications of the ACM* (46)10, pp.85-89.
- Shannon, C.E. (1962). Computers and Automation. Progress and Promise in the Twentieth Century. In: Sloane, N.J.A. and Wyner, A.D., Editors. *Claude Shannon Collected Papers*. IEEE Press 1993.
- Shannon, C.E. (1950). The Lattice Theory of Information . In: Sloane, N.J.A. and Wyner, A.D., Editors. *Claude Shannon Collected Papers*. IEEE Press 1993.
- Shannon, C.E. (1948). A Mathematical Theory of Communication. In: Sloane, N.J.A. and Wyner, A.D., Editors. *Claude Shannon Collected Papers*. IEEE Press 1993. pp. 5-83.
- Shannon, C.E. and Weaver, W. (1963). *The mathematical theory of communication*. University of Illinois Press.
- Shapiro, S. (2005). Categories, Structures, and the Frege-Hilbert Controversy: The Status of Meta-mathematics. *Philosophia Mathematica* (13)1, pp.61-77.
- Sharman, R., Kishore, R., and Ramesh, R. (2004). Computational Ontologies and Information Systems: II. Formal Specification. *Communications of the Association for Information Systems* (14), pp.184-205.
- Sherer, S.A. and Alter, S. (2004). Information System Risks and Risk Factors: Are They Mostly about Information Systems? *Communications of the Association for Information Systems* (14), pp.29-64.
- Simon, H.A. (1996). *The Sciences of the Artificial*. The MIT Press.
- Smith, A.D. (2003). *Husserl and the Cartesian Meditations*. Routledge.
- Smith, B. and Welty, C. (2001). FOIS Introduction. Ontology: Towards a New Synthesis. *2nd International Conference on Formal Ontology in Information Systems*, Ogunquit, Maine, USA. pp.iii-ix.
- Smith, J.M. and Smith, D.C.P. (1977). Database Abstractions: Aggregation and Generalization. *ACM Transactions on Database Systems* (2)2, pp.105-133 .
- Sokal, A. and Bricmont, J. (1998). *Intellectual Impostures*. Imposturas Intelectuales. Guix Vilaplana, J.C., Traductor. Ediciones Paidós, 1999.
- Sowa, J.F. (1984). *Conceptual Structures: Information Processing in Mind and Machine*. Addison-Wesley.

*Propuesta de un método de validación de esquemas conceptuales
y análisis comparativo de la noción de información
en los métodos de desarrollo de sistemas de información*

- Spyns, P., Meersman, R., and Jarrar, M. (2002). Data Modelling versus Ontology Engineering. *ACM SIGMOD RECORD* (31)4, pp.12-17.
- Stamper, R. (1987). Semantics. Boland Jr., R.J. and Hirschheim, R., Editors. *Critical Issues in Information Systems Research*. John Wiley & Sons Ltd. 1987. pp. pp.43-78.
- Stamper, R. (1991). The Semiotic Framework for Information Systems Research. In: Nissen, H.-E., Klein, H.K., and Hirschheim, R., Editors. *Information Systems Research: Contemporary Approaches & Emergent Traditions Proceedings of the IFIP TC8/WG8.2 Working Conference*, Copenhagen, Denmark. pp.515-527.
- Stamper, R. (1998). A dissenting position. In: Falkenberg, E.D., Hesse, W., Lindgreen, P., Nilsson, B., Oei, J.L.H., Rolland, C., Stamper, R., Assche, F.J.M.V., Verrijn-Stuart, A.A., and Voss, K., Authors. *A Framework of Information System Concepts (The FRISCO Report)*. IFIP 1998. pp. 191-196.
- Stamper, R., Liu, K., Kolkman, M., Klarenberg, P., Slooten, F.V., Ades, Y., and Slooten, C.V. (1991). From Database to Normbase. *International Journal of Information Management* (11), pp.67-84.
- Stamper, R. and Liu, K. (1994). Organisational Dynamics, Social Norms and Information Systems. *27th Hawaii International Conference on System Sciences*,
- Stamper, R., Liu, K., Hafkamp, M., and Ades, Y. (1997). Signs Plus Norms -- One Paradigm for Organisational Semiotics. *1st International Workshop on Computational Semiotics*, Paris.
- Standish Group. The CHAOS Report (1994) [Web Page]. 1995; Accessed 2004 Sep. Available at:
http://standishgroup.com/sample_research/chaos_1994_1.php.
- Stegmüller, W. (1973). *Probleme und Resultate der Wissenschaftstheorie und Analytischen Philosophie. Band II: Theorie und Erfahrung. (Zweiter Halbband: "Theorienstrukturen und Theoriendynamik")*. Estructura y Dinámica De Teorías. Segundo Tomo De Teoría y Experiencia. Moulines, C.U., Traductor. Ariel, 1983.
- Tarski, A. (1944). The Semantic Conception of Truth and the Foundations of Semantics. In: Blackburn, S. and Simmons, K., Editors. *Truth*. Oxford University Press 1999. pp. 115-143.
- Te'eni, D. (2001). Review: A Cognitive-Affective Model of Organizational

- Communication for Designing IT. *MIS Quarterly* (25)2, pp.251-312.
- Teorey, T.J., Yang, D., and Fry, J.P. (1986). A Logical Design Methodology for Relational Databases Using the Extended Entity-Relationship Model. *ACM Computing Surveys* (18)2, pp.197-222.
- Truex, D.P. and Baskerville, R.L. (1998). Deep structure or emergence theory: contrasting theoretical foundations for information systems development. *Information Systems Journal* (8)2, pp.99-118.
- Tseng, V.P. and Mannino, M.V. (1988). Inferring Database Requirements from Examples in Forms. Batini, C., Ed. *7th International Conference on Entity-Relationship Approach*, Roma. pp.391-405.
- Tsichritzis, D. (1982). Form Management. *Communications of the ACM* (25)7, pp.453-478.
- UML. OMG Unified Modeling Language Specification. Version 1.5 [Web Page]. 2003 Mar; Accessed 2004. Available at: www.uml.org.
- Verrijn-Stuart, A. A. Editor. *A Framework of Information System Concepts. The Revised FRISCO Report (Draft)*. (2001). IFIP
- Walsham, G. (1995). The Emergence of Interpretativism in IS Research. *Information Systems Research* (6)4, pp.376-394.
- Walton, S. and Huey, J. (1992). *Sam Walton. Made in America*. Bantam Book.
- Wand, Y. (1989). A Proposal for a Formal Model of Objects. In: Kim, W. and Lochovsky, F.H., Editors. *Object-Oriented Concepts, Databases, and Applications*. ACM Pres 1989.
- Wand, Y. *Ontological Modelling of Information Systems Concepts*. (1997).
- Wand, Y., Monarchi, D.E., Parsons, J., and Woo, C.C. (1995). Theoretical Foundations for Conceptual Modelling in Information Systems Development. *Decision Support Systems* (15), pp.285-304.
- Wand, Y., Storey, V.C., and Weber, R. (1999). An Ontological Analysis of the Relationship Construct in Conceptual Modeling. *ACM Transactions on Database Systems* (24)4, pp.494-528.
- Wand, Y. and Wang, R.Y. (1996). Anchoring Data Quality Dimensions in Ontological Foundations. *Communications of the ACM* (39)11, pp.87-95.

- Wand, Y. and Weber, R. (1988). An Ontological Analysis of some Fundamental Information Systems Concepts. *9th International Conference on Information Systems*, Minneapolis, Minnesota.
- Wand, Y. and Weber, R. (2002). Research Commentary: Information Systems and Conceptual Modeling --A Research Agenda. *Information Systems Research* (13)4, pp.377-388.
- Weber, R. (1996). Are Attributes Entities? A Study of Database Designers' Memory Structures. *Information Systems Research* (7)2, pp.137-162.
- Weber, R. (2003a). Editor's Comments: Still Desperately Seeking the IT Artifact. *MIS Quarterly* (27)2, pp.iii-xi.
- Weber, R. (2003b). Editor's Comments: Theoretically Speaking. *MIS Quarterly* (27)3, pp.iii-xii.
- Whitehead, A.N. (1929). *Process and Reality*. The Free Press, 1985.
- Wilshire, B.W. Metaphysics. *Encyclopaedia Britannica*. 2002.
- Wing, J.M. (1990). A Specifier's Introduction to Formal Methods. *IEEE Computer* (23)9, pp.8-24.
- Winograd, T. (1987). A Language/Action Perspective on the Design of Cooperative Work. *Human-Computer Interaction* (3)1, pp.3-30.
- Winograd, T. and Flores, F. (1986). *Understanding Computers and Cognition*. Ablex Publishing Corporation, 1989.
- Winslett, M. (2004). Peter Chen Speaks Out. *ACM SIGMOD RECORD* (33)1, pp.110-118.
- Wittgenstein, L. (1953). *Philosophical Investigations*. Prentice Hall, 1958.
- Woodcock, J. and Davies, J. (1996). *Using Z, Specification, Refinement, and Proof*. Prentice Hall, 1996 .
- Wu, J.-H., Doong, H.-S., Lee, C.-C., Hsia, T.-C., and Liang, T.-P. (2004). A Methodology for Designing Form-Based Decision Support Systems. *Decision Support Systems* (36)3, pp.313-335.
- Yao, S.B., Hevner, A.R., Shi, Z., and Luo, D. (1984). FORMANAGER: An Office Forms Management System. *ACM Transactions on Information Systems*

(2)3, pp.235-262.

Yourdon, E. (1989). *Modern Structured Analysis*. Prentice-Hall.

Yule, G. (1996). *Pragmatics*. Oxford University Press.

Zave, P. (1996). Formal Methods are Research, not Development. *IEEE Computer* (29)4, pp.26-27.

Zave, P. and Jackson, M. (1997). Four Dark Corners of Requirements Engineering. *ACM Transactions on Software Engineering and Methodology* (6)1, pp.1-30.

Zhirnov, V.V. and Herr, D.J.C. (2001). New Frontiers: Self-Assembly and Nanoelectronics. *IEEE Computer* (34) 1, pp.34-43.

Zúñiga, G.L. (2001). Formal Ontology in Information Systems. *International Conference on Formal Ontology in Information Systems*, Maine, USA. pp.187-197.