

Universidad de Granada
Departamento de Lenguajes y Sistemas Informáticos



Tesis Doctoral

**SISTEMAS DE RECOMENDACIÓN SEMÁNTICOS
PARA LA COMPARTICIÓN DE CONOCIMIENTO Y
LA EXPLOTACIÓN DE TESAUROS: Un enfoque
práctico en el ámbito de los sistemas nutricionales.**

Presentado por
Vanesa Espín Martín

Dirigido por
Dra. María Visitación Hurtado Torres
Dr. Manuel Noguera García

Granada, Mayo de 2016

Editor: Universidad de Granada. Tesis Doctorales

Autora: Vanesa Espín Martín

ISBN: 978-84-9125-959-6

URI: <http://hdl.handle.net/10481/44033>

Índice de Contenidos

PREFACE	8
Motivations	9
Objectives.....	10
Document Structure	11
Chapter References	12
CAPÍTULO 1	
SISTEMAS DE RECOMENDACIÓN	14
1.1. Introducción	14
1.2. Definiciones de Sistema de Recomendación.....	15
1.3. Estructura general de los Sistemas de Recomendación.....	15
1.4. Clasificación de los Sistemas de Recomendación.....	16
1.4.1 Sistemas de Recomendación de Filtro Colaborativo	18
1.4.2 Sistemas de Recomendación Basados en Contenido.....	21
1.4.3 Sistemas de Recomendación Basados en Conocimiento.....	25
1.4.4 Sistemas Híbridos.....	26
1.5. Limitaciones de los SR tradicionales y nuevas tendencias	28
Referencias del capítulo	29
CAPÍTULO 2	
WEB SEMÁNTICA, TESAUROS Y ONTOLOGÍAS	32
2. 1. Introducción la Web Semántica y su Arquitectura.....	32
2. 2. Tesauros y Ontologías.....	35
2.2.1 Definiciones y Clasificación de Ontologías	36
2.2.2 Componentes de una Ontología	40
2. 3. Lenguajes para la Definición de Tesauros y Ontologías	40
2.3.1 RDF y RDF-Schema	41
2.3.2 Limitaciones del poder expresivo de RDF y RDF Schema	45
2.3.3. OWL	46
2.3.4 OWL2.....	51
2.3.5 SKOS.....	54
2. 4. Proceso de Desarrollo de Ontologías	55
2. 5. Herramientas para la edición y manejo de ontologías.....	57
2.5.1 Protégé.....	57

2.5.2 Jena.....	58
2.5.3 OWL API.....	59
2.5.4 Herramientas de Gestión de XML.....	59
2. 6. Lógica Descriptiva.....	61
2.6.1 Decidibilidad y complejidad computacional en DL.....	61
2.6.2 Lenguajes de descripción.....	62
2.6.3 Familias <i>AL</i> relacionadas con versiones de OWL DL.....	64
2. 7. Reglas y SWRL.....	65
2. 8. El razonamiento ontológico.....	69
3.8.1 FaCT++.....	69
3.8.2 Hermit.....	70
3.8.3 Pellet.....	70
Referencias del capítulo.....	70
CAPÍTULO 3	
SISTEMAS DE RECOMENDACIÓN SEMÁNTICOS.....	73
3. 1. Introducción.....	73
3. 2. Características de los Sistemas de Recomendación Semánticos.....	74
3.2.1 Tipo de base de conocimiento.....	74
3.2.2 Modelado del usuario y perfiles de usuario.....	75
3.2.3 Estrategias de matching ítem-usuario.....	76
3. 3. Ejemplos de SRS.....	79
Referencias del capítulo.....	80
CAPÍTULO 4	
SISTEMAS DE RECOMENDACIÓN EN EL DOMINIO NUTRICIONAL.....	82
4. 1. Introducción.....	82
4. 2. Desafíos en las recomendaciones nutricionales.....	82
4. 3. Fuentes de información nutricional.....	83
4. 4. Trabajo Relacionado.....	84
4.4.1. Perfil Nutricional del Usuario.....	84
4.4.2. Estrategias de Recomendación.....	85
4.4.3. Monitorización.....	86
4.4.4. Análisis de preferencias y personalización de las recomendaciones.....	86
4. 5. Otros Sistemas de Recomendación Nutricionales.....	87
Referencias del capítulo.....	89

CAPÍTULO 5

NutElCare: SISTEMA DE RECOMENDACIÓN SEMÁNTICO NUTRICIONAL.....	92
5.1. Introducción	92
5.2. Funcionamiento del Sistema	93
5.3. Arquitectura del Sistema	94
5.3.1 Principios de usabilidad y accesibilidad del sistema.....	96
Referencias del capítulo	98

CAPÍTULO 6

ADQUISICIÓN Y REPRESENTACIÓN DEL CONOCIMIENTO DESDE EL ENFOQUE SEMÁNTICO	100
6. 1. Introducción	100
6. 2. Adquisición de Conocimiento Nutricional.....	102
6.2.1 Características Nutricionales de los Alimentos.....	102
6.2.2 Tipos de Dietas para Personas Mayores.....	103
6.2.3 Recomendaciones Nutricionales para Personas Mayores	103
6.2.4 Restricciones Alimenticias.....	104
6. 3. Representación del Conocimiento Nutricional.....	104
6.3.1 Creación de la Ontología de Dominio. Enfoque Top-Down.....	105
6.3.2 Creación de la Ontología de Dominio. Enfoque Bottom-up.	121
6.3.3 Representación Ontológica de Dietas.....	123
6. 4. Adquisición de Conocimiento para la Representación del Perfil de Usuario.....	128
6. 5. Representación del Perfil de Usuario	131
6.5.1 Reutilización de Modelos de Usuario Ontológicos	131
6.5.2 Desarrollo de la Ontología de Perfil de Usuario	132
6. 6. Enlazado de la Ontología Nutricional y el Perfil de Usuario	134
6. 7. Razonamiento mediante Reglas SWRL	135
Referencias del capítulo	137

CAPÍTULO 7

EXPLOTACIÓN DEL CONOCIMIENTO SEMÁNTICO Y PRODUCCIÓN DE RECOMENDACIONES	140
7.1. Introducción	140
7.2. Explotación del Conocimiento Semántico	141
7.2.1 Inicialización de OWL API.....	141
7.2.2 Razonamiento en NutElCare.....	142
7.3. Formalización de la Estrategia de Recomendación.....	146

7.4.	Instanciación del Perfil de Usuario	148	
7.5.	Recomendación Basada en Conocimiento. Selección del Modelo de Dieta.	149	
7.6.	Recomendación basada en Contenido. Obtención de Alternativas.	151	
7.6.1	Análisis de Similitud Semántica desde la Perspectiva del Sistema	152	
7.6.2	Cálculo de Similitud Semántica en NutElCare.....	153	
7.7.	Aplicación de la Estrategia de Recomendación.	158	
	Referencias del capítulo	164	
CHAPTER 8			
CONCLUSIONS & ONGOING WORK			166
8.1.	Conclusions	166	
8.2.	Ongoing Work.....	170	
8.2.1	Intra-System Enhancements	170	
8.2.2	Knowledge Sharing	170	
	Chapter References	171	
ANEXO A			
EJEMPLOS DE MODELOS DE DIETAS PARA MAYORES.....			173
ANEXO B			
RESTRICCIONES Y RECOMENDACIONES NUTRICIONALES PARA MAYORES			175
ANEXO C			
ESTUDIO DE INTOLERANCIA ALIMENTICIA A-100			180
ANEXO D			
CUESTIONARIOS DE SCREENING			181
Referencias Bibliográficas de los Anexos.....			183
Bibliografía			184

Índice de Figuras

Figura 1.1. Técnicas de recomendación y sus fuentes de conocimiento. Fuente: (Burke, 2007). 17	
Figura 1.2. Arquitectura de un SR Basado en Contenido. Fuente: (Lops et al., 2011).	22
Figura 1.3. Conexión entre las fuentes de conocimiento y los tipos de recomendación. Fuente: (Burke, 2007).	27
Figura 2.1. Pila de Tecnologías de la Web Semántica. Fuente: W3C.....	33
Figura 2.2. Clasificación de ontologías según su grado de generalidad. Fuente: (Guarino, 1998).	37
Figura 2.3. Espectro de ontologías. Fuente: (Lassila and Mcguinness, 2001).....	39
Figura 2.4. Sentencia expresada en forma de grafo RDF. Fuente: W3C.	41
Figura 2.5. Grafo RDF. Fuente: W3C.....	42
Figura 2.6. Vocabulario OWL Lite, OWL DL, OWL Full. Fuente: (W3C, 2004).	47
Figura 2.7. Relaciones de subclase entre OWL y RDF/RDFS. Fuente: (Antoniou and Harmelen, 2003)	48
Figura 2.8. Posibles sintaxis de OWL2. Fuente: W3C.....	52
Figura 2.9. Sublenguajes OWL1 vs OWL2 . Fuente: (Hoekstra, 2009).	53
Figura 2.10. Arquitectura de Jena. Fuente: Apache Jena.....	58
Figura 2.11. Equivalencia entre sintaxis DL y semántica <i>FOL</i> . Fuente: (Grosz et al., 2003). ..	64
Figura 2.12. Componentes principales de Pellet. Fuente: (Sirin et al., 2007).....	70
Figura 5.1. Arquitectura de NutEiCare. Fuente (Espín et al., 2016).	95
Figura 5.2. Ejemplo de uso de pictogramas ARASAAC en NutEiCare. Fuente: (Espín et al., 2016).	98
Figura 6.1. Extracto de tabla de composición nutricional.	102
Figura 6.2. Principales nutrientes por grupos de alimentos. Fuente: (Pinilla and de Torres Aured, 2007).	102
Figura 6.3. Grupos básicos de alimentos a consumir cada día y su función en el organismo... 104	
Figura 6.4. Extracto de la ontología PIPS Food desplegada en Protégé.	107
Figura 6.5. Segunda versión del concepto Productos de AGROVOC desplegado en Protégé. 113	
Figura 6.6. Concepto de Dominio AGROVOC y sus subclases directas en Protégé.	113
Figura 6.7. AGROVOC desplegada en su primer nivel.	114
Figura 6.8. Representación gráfica de Productos AGROVOC. Subclase Alimentos desplegada.	115
Figura 6.9. Ejemplo de eliminación de clases en Protégé.	116
Figure 6.10. Nueva jerarquía de la clase “Alimentos” en la ontología nutricional.	118
Figura 6.11. Extracto de las propiedades de tipo de datos de la ontología nutricional.	119

Figura 6.12. Algunas Propiedades de Objeto de la ontología nutricional.	120
Figura 6.13 Representación de las propiedades nutricionales de la instancia “Lemon”.	121
Figura 6.14. Restricciones de propiedad sobre “Recommended intake” para delimitar las raciones semanales recomendadas.	122
Figura 6.15. Modelo gráfico del XML Schema de una Dieta mediante XSDDiagram.	123
Figura 6.16. Definición XML Schema del elemento “Meal” de una dieta.	124
Figura 6.17. Dieta anotada mediante XML siguiendo el XML Schema definido.	125
Figura 6.18. Enriquecimiento de la ontología nutricional a partir del modelo sintáctico de las dietas del repositorio XML. Fuente: (Espín et al., 2015).	125
Figura 6.19. Página JSP que realiza la transformación XSLT.	126
Figura 6.20. Extracto de la hoja de estilos XSL para la definición del mapeo de reglas.	127
Figura 6.21. Resultado de aplicación de la transformación XSLT sobre el nodo “Meal” a clase OWL.	127
Figura 6.22. Resultado del enriquecimiento de la ontología mediante la transformación XSLT del modelo sintáctico XML de las dietas.	128
Figura 6.23. Extensión de la dimensión básica del perfil de usuario GUMO con requerimientos nutricionales.	133
Figura 6.24. Dimensión de contexto de GUMO reutilizada en el perfil de usuario.	133
Figura 7.1 Inicialización de OWL API con la ontología nutricional.	142
Figura 7.2 Carga del razonador Pellet como Listener de cambios en la ontología.	142
Figura 7.3. Inferencia del estado nutricional del usuario mostrado en Protégé.	143
Figura 7.4. Propiedades de objeto inferidas de la instancia “Gluten”.	144
Figura 7.5. Propiedades del plato “spaguettiBolognaise”.	145
Figura 7.6. Descripción de la instancia “spaguetti”.	145
Figura 7.7. Explicación de inconsistencia al recomendar a Mary un alimento con gluten.	146
Figura 7.8 Diagrama de flujo de la estrategia de recomendación de NutElCare.	147
Figura 7.9. Estructura taxonómica de tipos de dietas y elementos considerados en el enfoque basado en restricciones.	151
Figura 7.10. Ejemplo de distancia taxonómica entre dos instancias <i>i</i> y <i>j</i>	156
Figura 7.11. Propiedades instanciadas del usuario del sistema para la recomendación de dietas.	159
Figura 7.12. Ejemplo de clasificación de dietas según restricciones de usuario con valores de pesos sobre las restricciones.	159
Figura 7.13. Clasificación de las dietas #Diet11 y #Diet39.	160
Figura 7.14. Selección de la clase “Fresh Meat” como nodo de parada y sus clases descendientes.	161

Figura 7.15. Instancias adyacentes a “Chicken thigh” y su distancia taxonómica.....	162
Figura 7.16. Cálculo de pesos en propiedades LP entre “Chicken thigh” y su adyacente “Pork shoulder”.....	162
Figura 7.17. Cálculo de pesos en propiedades TP entre “Chicken thigh” y su adyacente “Pork shoulder”.....	163
Figura 7.18. Diez primeras instancias más similares a “Chicken thigh” previamente al uso del factor de interés.....	163
Figura 7.19. Diez primeras instancias más similares a “Chicken thigh” ajustadas al gusto del usuario, al cual se le devuelven las Top K más similares, siendo K=5.....	164
Figura D.1. MNA-SF en español. Fuente: (Vellas et al., 2006) y (Rubenstein et al., 2001).....	181
Figura D.2. Ejemplo de Cuestionario de Frecuencia de Consumo Alimenticio CFCA. Fuente: (Trinidad Rodríguez et al., 2008).....	182

Índice de Tablas

Tabla 1.1. Ejemplo de Matriz de Ratings.....	19
Tabla 6.1. Reglas de mapeo entre los nodos del modelo de dieta XSD definido a conceptos OWL.....	126
Tabla 6.2. Valoración del Estado Nutricional según IMC. Fuente: (SENPE and SEGG, 2007).	129

PREFACE

Proper nutrition is a key factor for human health but it becomes essential for more vulnerable sectors of population as children, elderly or people with some diseases or disabilities. Nutrition experts continue working on the design of healthy diet plans that meet the requirements of these sectors. Focusing on the elderly population, there is a strong evidence that a healthy eating along with adequate physical exercise may improve their quality of life and foster longevity (World Health Organization, 2002a).

The large volume of data available online contributes to the enormous potential of the Web, exposing, however, some weaknesses. There is a wealth of information on nutrition and healthy diets spread throughout the Web, such as health magazines, nutritional databases, forums, etc. Part of this information has been drawn up by experts upon well-founded scientific basis so they can be helpful for users as long as it stems from reliable sources. However, in many cases, deciding whether a source is reliable or not, is not a trivial task. In addition, this wealth of information may be difficult to assimilate by non-expert users, due to, among other factors, the complexity of managing this large volume of data to suit one's specific needs, the disparity in the sources' reliability or the completeness of the information described.

Recommender Systems (RS) are counselling tools that generate recommendations on a particular object of study using different techniques to try to satisfy the needs and preferences of users (Herrera-Viedma et al., 2004). These systems are able to partially solve the problems cited by collecting expert nutritional information and processing it to offer personalized recommendations to the users.

An essential factor in these systems is to represent the information in a way that enables processes and applications to retrieve such information efficiently for reuse, and also facilitates communication between system agents and even among external agents, encouraging collaboration among them and thus increasing the quality of recommendations. However, the heterogeneity of information representation in traditional RS has been detrimental to these features.

In the context of the Semantic Web, several technologies have been developed to represent and share knowledge by using a common vocabulary, so its use has begun to be extended to all areas where it is necessary to dispose of formal representations for describing and reasoning about a domain of discourse.

A solution proposal in recent years to solve problems of heterogeneity, communication and reuse of information in classical RS is the application of Semantic Web Technologies in development, such as ontologies (Middleton et al., 2004). Ontologies are known to be large sources of knowledge. This is largely due to their ability to reason and the semantics which they supply; they provide formal, uniform, reusable and sharable representations on a domain and have been applied to different Recommender Systems to reduce heterogeneity and improve content retrieval.

OWL (Web Ontology Language), now evolved to its OWL 2 version (Grau et al., 2008), is the most widespread standard language for definition and management of ontologies, because it has expressive power to add semantics that allows reasoning about them thanks to its equivalence to Description Logics (DL).

Motivations

One of the major drawbacks that have suffered classic Recommender Systems from, is the problem of heterogeneity of representation, leading to communication problems among the actors involved in the system and among agents and users, implying also that the information can not be reused by other agents, processes or applications (Lops et al., 2011), (Peis, 2008).

In recent years, several authors have applied technologies of the Semantic Web to give partial solutions to some of the problems associated with information representation in Recommender Systems (Farsani and Nematbakhsh, 2006), (Blanco-Fernández et al., 2008b) and (Cantador et al., 2011), achieving significant improvements in the quality of recommendations.

Although some proposals related to nutritional RS can be found in the literature, most of them have been focused on obtaining recipes that appeal to users (Svensson et al., 2000), (Lawrence et al., 2001), (Sobecki et al., 2006), (Snae and Bruckner, 2008), (Zhang et al., 2008) and significantly fewer are oriented health care, although in recent years considerable efforts on the inclusion of health requirements in such systems are being made (Freyne et al., 2011), (van Pinxteren et al., 2011), (El-Dosuky et al., 2012). The fact that targeted users of this type of systems belong to more vulnerable sectors of the population, nutritionally speaking, only is taken into account an even smaller number of them. Some examples are found in (Aberg, 2006) and (Arwan et al., 2013) intended to elderly and diabetic users, respectively.

The expected growth in the aging of the world population, ubiquity and affordability of smart devices, and the control over the costs of health care lead to new scientific challenges to be addressed in order to provide new life opportunities for the aging population (Benghazi et al., 2012). The scarcity of nutritional systems oriented to elderly people has motivated this work focusing on these users, since they constitute a sector of the population for whom eating properly is a critical factor.

Provided that a lack of motivation is a major obstacle in any health therapy (Becker and Maiman, 1980), for the effective operation of a health-related RS it is crucial the consideration of the *user compliance*, which consisting on constantly keep the motivation of the user in using the system. To this end, the main challenge to face in the nutritional domain is that the recommendations fit both needs, and preferences of users (Suksom et al., 2010), because if the recommendations are not consistent with their needs could damage their health and, if not desirable, the user will end up not following them and eventually stop using the system (Mika, 2011). To do this, the system must be able to perform regular monitoring of users' consumption, learning their taste preferences and habits, detecting at the same time possible nutritional deficiencies and try to correct them (De Pessemier et al., 2013).

The user model must collect all this information about the use the system along with its nutritional profile. At the same time, the recommendations should be offered in a flexible and efficient use of the system and must be easy and enjoyable for users.

Currently, there are no proposals that meet all these requirements; therefore, this thesis aims to provide a solution as complete as possible that addresses each of the aforementioned challenges.

Objectives

The aim of this paper is to propose a Recommender System capable of efficiently storing either human (as nutritionists or geriatricians) or computerized data from various expert sources (such as nutritional databases, thesauri, information systems or recommendations from WHO¹ or SENPE²). Thus, it is possible to dispose of a reliable and as complete as possible nutritional information.

The RS must manage all the information stored to provide dietary recommendations tailored to the needs of each type of user and customized according to their tastes and specific requirements. From this information, the system must be able to offer appropriate nutritional advice and recommendations automatically, in agile and efficient manner. These tips could be recipes or healthy diet plans for a period of time as required by the characteristics of a specific person or group of people.

Being a health-related system, it must be able to supply current and periodic information on their use, i.e., whether the user follows the recommendations, and the influence of those recommendations in their nutritional status. Therefore, the system must be complemented by a monitoring component that allows being aware of its influence on the users' nutritional state. The inclusion of learning techniques of the user behaviour in the system will improve the quality of recommendations by learning from their tastes and will encourage the implementation of retrospective methods which analyse the impact of the use of the system in the user's health. This information could be used by the system, by the user, or caregiver or dietician, so it permits the detection of possible deficiencies, reporting on them and even try to correct them.

One of the primary goals of this thesis is to enhance the representation and management of information by applying Semantic Web Technologies on the system development resulting in a Semantic Recommender System (SRS) which provides nutritional recommendations and healthy diets to elderly users. These recommendations are always guided by the nutritional properties of food and expert recommendations, which will be represented in the knowledge base of the system through a nutritional ontology represented in OWL language.

To accomplish this proposal the following specific objectives have raised:

1. Study of Recommender Systems and semantic representation techniques of information, along with their application to such systems.
2. Analysis and state of the art of RS in the nutritional field.
3. Development of an ontology that integrates nutritional information from expert sources, such as thesauri and nutritional databases, and the material provided by nutritionists and geriatricians who collaborate in this work.
4. Study of users modelling techniques along with the user behaviour and their representation in semantic profiles using ontologies within the scope of Semantic Recommender Systems.
5. Analysis of requirements and definition of the user model and semantic profiles from the point of view of nutritional systems. Establishment of an ontology which allows semantic representation of different user profiles, and relevant relationships between these profiles.

¹ World Health Organisation

² Sociedad Española de Nutrición Parenteral y Enteral (Spanish Society of Parenteral and Enteral Nutrition)

6. Definition of the requirements to be met by a food recommendation in terms of the information provided and its representation. An analysis of strategies and algorithms for obtaining recommendations and their implementation in such nutritional SRS in order to meet the requirements defined will be performed.
7. Study and application of learning techniques to analyse and learn the interests and degree of compliance of users with the system (*user compliance*) from their behaviour, and by using retrospective methods to reinforce user modelling and, hence, the quality of the recommendations.
8. To validate the applicability of the techniques and methods proposed, a SRS which integrates the nutritional information contained in an ontology will be built. From this information, the system must be able to offer nutritional recommendations fitting the needs and tastes of a particular person or group of persons automatically, nimbly and efficiently.

Document Structure

This report reflects the work performed to obtain the title of Doctor in the Official PhD on Information and Communication Technologies issued by the University of Granada. This thesis is also eligible for International Mention.

The document is divided into the following chapters:

- Chapter 1 presents an introduction to Recommender Systems. Its origins, evolution, definitions and structure, providing an study and examples of its classification together with the advantages and drawbacks of each type of RS and classic RS in general.
- Chapter 2 details the Semantic Web and its main technologies, focusing on thesauri and ontologies. It includes a deep introduction to OWL language and its equivalence to Description Logics, analysing the key benefits on using it as language for ontologies definition. Also, other semantic technologies are explained along with the different reasoners which permit the inference of new knowledge from the information contained in an ontology.
- In Chapter 3, it is presented a study of the motivations and application of Semantic Web Technologies in the development of RS. Several proposals found in the literature have been shown and the main techniques used have been analysed, being the most important the semantic similarity calculation between ontology concepts.
- Chapter 4 provides a revision of the main challenges to address in the development of Nutritional RS, and how these challenges have been tackled by other researchers. We present a state of the art on Nutritional Recommender Systems identifying several enhancements to apply in our proposal.
- Chapter 5 gives an overall view of the RS developed during this thesis, NutElCare. It shows its main motivations and present the software architecture proposed to cope the requirements of a Nutritional Semantic RS intended to elderly people.
- Chapter 6 presents the formalisation of the NutElCare conceptual framework by its specification in a *domain ontology* represented as an OWL 2 ontology. The process of the ontology development from a well known thesaurus on nutrition along with the information provided by the nutrition and geriatric experts who collaborate in this work is explained. Also, it is presented the ontological modelling of the user profile through

the reuse of an existent user model general ontology. The ontologies introduced make up the knowledge base of the system.

- In Chapter 7, the system knowledge base is exploded in order to provide nutritional recommendations to final users. To this end, a novel hybrid recommendation strategy oriented to nutrition is proposed together with a semantic similarity approach for the food items contained in the ontology. During this chapter, some examples of reasoning processes involved in the recommendations ending with an example of a complete recommendation process which serves to validate our proposal.
- Finally, Chapter 8 discusses the main conclusions and remarks some potential work lines of research related to this thesis.

Chapter References

- ABERG, J. 2006. Dealing with Malnutrition: A Meal Planning System for Elderly. *AAAI Spring Symposium: Argumentation for Consumers of Healthcare*.
- ARWAN, A., PRIYAMBADHA, B., SARNO, R., SIDIQ, M. & KRISTIANTO, H. 2013. Ontology and semantic matching for diabetic food recommendations. *Information Technology and Electrical Engineering (ICITEE)*. IEEE.
- BECKER, M. H. & MAIMAN, L. A. 1980. Strategies for enhancing patient compliance. *Journal of community health*, 6, 113-135.
- BENGHAZI, K., HURTADO, M. V., HORNOS, M. J., RODRÍGUEZ, M. L., RODRÍGUEZ-DOMÍNGUEZ, C., PELEGRINA, A. B. & RODRÍGUEZ-FÓRTIZ, M. J. 2012. Enabling correct design and formal analysis of Ambient Assisted Living systems. *Journal of Systems and Software*, 85, 498-510.
- BLANCO-FERNÁNDEZ, Y., PAZOS-ARIAS, J. J., GIL-SOLLA, A., RAMOS-CABRER, M., LÓPEZ-NORES, M., GARCÍA-DUQUE, J., FERNÁNDEZ-VILAS, A. & DÍAZ-REDONDO, R. P. 2008. Exploiting synergies between semantic reasoning and personalization strategies in intelligent recommender systems: A case study. *Journal of Systems and Software*, 81, 2371-2385.
- CANTADOR, I., CASTELLS, P. & BELLOGÍN, A. 2011. An enhanced semantic layer for hybrid recommender systems: Application to news recommendation. *International Journal on Semantic Web and Information Systems*, 7, 44-78.
- DE PESSEMIER, T., DOOMS, S. & MARTENS, L. A food recommender for patients in a care facility. Proceedings of the 7th ACM conference on Recommender systems, 2013. ACM, 209-212.
- EL-DOSUKY, M. A., RASHAD, M. Z., HAMZA, T. T. & EL-BASSIOUNY, A. H. 2012. Food Recommendation Using Ontology and Heuristics. *Communications in Computer and Information Science*, 322, 423-429.
- FARSANI, H. K. & NEMATBAKHS, M. 2006. A semantic recommendation procedure for electronic product catalog. *International Journal of Applied Mathematics and Computer Sciences*, 3, 86-91.
- FREYNE, J., BERKOVSKY, S. & SMITH, G. 2011. Recipe recommendation: Accuracy and reasoning. *User Modeling, Adaption and Personalization*. Springer Berlin Heidelberg.
- GRAU, B. C., HORROCKS, I., MOTIK, B., PARSIA, B., PATEL-SCHNEIDER, P. & SATTLER, U. 2008. OWL 2: The next step for OWL. *Web Semantics: Science, Services and Agents on the World Wide Web*, 6, 309-322.
- HERRERA-VIEDMA, E., PORCEL, C. & HIDALGO, L. 2004. Sistemas de recomendaciones: herramientas para el filtrado de información en Internet. *Hipertext.net*, 2.
- LAWRENCE, R. D., ALMASI, G. S., KOTLYAR, V., VIVEROS, M. & DURÍ, S. S. 2001. Personalization of supermarket product recommendations. *Data Mining and Knowledge Discovery*, 5, 11-32.

- LOPS, P., GEMMIS, M. D. & SEMERARO, G. 2011. Content-based recommender systems: State of the art and trends. *Recommender systems handbook*. Springer US.
- MIDDLETON, S. E., SHADBOLT, N. R. & DE ROURE, D. C. 2004. Ontological user profiling in recommender systems. *ACM Transactions on Information Systems (TOIS)*, 22, 54-88.
- MIKA, S. 2011. Challenges for Nutrition Recommender Systems. *Proceedings of the 2nd Workshop on Context Aware Intel. Assistance*. Berlin, Germany.
- PEIS, E., DEL CASTILLO, J. M., & DELGADO-LÓPEZ, J. A. 2008. Semantic recommender systems. Analysis of the state of the topic. *Hipertext.net*, 6, 1-5.
- SNAE, C. & BRUCKNER, M. FOODS: a food-oriented ontology-driven system. 2nd IEEE International Conference on Digital Ecosystems and Technologies, 2008. IEEE, 168-176.
- SOBECKI, J., BABIAK, E. & ŚLANINA, M. 2006. Application of hybrid recommendation in web-based cooking assistant. *Knowledge-Based Intelligent Information and Engineering Systems*. Heidelberg: Springer Berlin.
- SUKSOM, N., BURANARACH, M., THEIN, Y. M., SUPNITHI, T. & NETISOPAKUL, P. A Knowledge-based Framework for Development of Personalized Food Recommender System. *Proceedings of the 5th Int. Conf. on Knowledge, Information and Creativity Support Systems*, 2010.
- SVENSSON, M., LAAKSOLAHTI, J., HÖÖK, K. & WAERN, A. 2000. A recipe based on-line food store. *Proceedings of the 5th international conference on Intelligent user interfaces*. ACM.
- VAN PINXTEREN, Y., GELEIJNSE, G. & KAM, P. 2011. Deriving a recipe similarity measure for recommending healthful meals. *Proceedings of the 16th international conference on Intelligent user interfaces*. ACM.
- WORLD HEALTH ORGANIZATION 2002. *Keep fit for life: meeting the nutritional needs of older persons*. Geneva, World Health Organization.
- ZHANG, Q., HU, R., MAC NAMEE, B. & DELANY, S. J. 2008. Back to the future: Knowledge light case base cookery. Technical report, Dublin Institute of Technology.

CAPÍTULO 1

SISTEMAS DE RECOMENDACIÓN

Las personas en su vida diaria reciben recomendaciones que pueden venir de diversas fuentes. Decidir en qué fuentes confiar y qué recomendaciones seguir es una tarea intrínseca al ser humano. Si se traslada esto al ámbito de los Sistemas de Información donde, a partir de una gran cantidad de datos, un sistema sea capaz de ofrecer recomendaciones personalizadas a los usuarios para ayudarles en sus procesos de decisión, surge un nuevo tipo de sistemas, conocidos como Sistemas de Recomendación. En este capítulo se pretende dar una idea global acerca de qué son los Sistemas de Recomendación y cómo han evolucionado desde su origen en la década de los 90 hasta la actualidad. Se explicarán los distintos tipos de Sistemas de Recomendación, así como algunos ejemplos existentes. Se analizará su estructura y algunas de las técnicas más utilizadas para producir sus resultados. Finalmente, se pondrán de relieve las deficiencias y carencias más importantes que los Sistemas de Recomendación tradicionales presentan y se expondrán varias de las soluciones propuestas a esos problemas.

1.1. Introducción

Las personas en su vida diaria reciben recomendaciones desde diversas fuentes: mediante el boca a boca, a través de cartas de recomendación, revistas especializadas, críticas de discos, películas, libros, etc. Normalmente, la decisión de confiar en esas recomendaciones se basa en la experiencia de la persona sobre el tema en cuestión o en la confianza de la fuente que realiza la recomendación. En ciertos momentos, la cantidad de opciones donde elegir es o bien tan elevada que una persona puede verse incapaz de recoger y resumir toda la información para decidir, o bien sus conocimientos sobre la materia pueden ser insuficientes como para seguir una recomendación acertada.

Dentro del ámbito de la computación, podría pensarse intuitivamente en un problema de recuperación de información tradicional, por ejemplo, almacenar la información en una base de datos donde el usuario haga consultas y obtenga la información que desea. Pero esto no es lo que persiguen las recomendaciones que nos interesan, por varios motivos, entre ellos, los usuarios no siempre son capaces de expresar lo que desean y, además, estas recomendaciones no tienen en cuenta al usuario que realiza las consultas, puesto que devolverán la misma respuesta independientemente de la persona que realizó la petición.

Estos aspectos se trasladan a los Sistemas de Información donde los Sistemas de Recomendación (SR) surgen de la necesidad de ofrecer recomendaciones personalizadas a los usuarios. Estos sistemas ayudan a los usuarios a encontrar la información que desean, o en la que podrían estar interesados, mediante recomendaciones. El uso de SR se ha incrementado fuertemente en las dos últimas décadas, sobre todo debido a su potencial comercial, y su éxito está principalmente ligado a que la forma de realizar las recomendaciones es muy similar a la manera en que lo harían unas personas a otras, tratando de simular este comportamiento.

1.2. Definiciones de Sistema de Recomendación

No existe una única definición en la literatura para describir a los SR. Se puede encontrar un amplio abanico de ellas que inciden en distintos aspectos de los mismos, por ejemplo:

- Resnick y Varian (Resnick and Varian, 1997) definieron originariamente los SR como un sistema en el cual *“people provide recommendations as inputs, which the system then aggregates and directs to appropriate recipients”*. Ahora esta definición resulta algo incompleta debido a la evolución de estos sistemas.
- Wang (Wang, 1998) aporta la siguiente definición: *“There is a collection of objects stored in a computer system, and each of them is characterized by a set of attributes. The system’s task is to select some objects according to the user’s requirement”*. Pero como bien indica Wang, esta definición es demasiado simple ya que no aporta la connotación de “recomendación” que lo hacen diferir de cualquier tipo de sistema experto.
- Sarwar (Sarwar et al., 2000) indica: *“Recommender systems apply knowledge discovery techniques to the problem of making product recommendations during a live customer interaction”*. Esta definición está más bien orientada al comercio electrónico, pero ya empieza a hablarse de descubrimiento de conocimiento.
- Burke (Burke, 2000) los define: *“Recommender systems provide advice to users about items they might wish to purchase or examine”*. Esta es una de las definiciones que más se aproximan a nuestra noción de Sistemas de Recomendación, en especial por la connotación que *“might wish”* (podría desear) aporta.
- Tiempo después Burke, (Burke, 2002), amplía su propia definición: *“System that produces individualized recommendations as output or has the effect of guiding the user in a personalized way to interesting or useful objects in a large space of possible options”*.
- Massa y Avesani (Massa and Avesani, 2004) aportan otra definición: *“Recommender Systems allow people to find the resources they need by making use of the experiences and opinions of their nearest neighbours”*.
- Melville y Sindhvani (Melville and Sindhvani, 2010) más recientemente indican: *“The goal of a Recommender System is to generate meaningful recommendations to a collection of users for items or products that might interest them”*.

Aunque algunas de las definiciones citadas incluyen el componente colaborativo, sin embargo, como se verá a continuación, éste no tiene por qué existir. Por lo tanto, las recomendaciones en los SR están basadas, o bien en las preferencias de los usuarios acerca de los ítems a recomendar, ignorando el componente colaborativo, o bien en el contexto social de los usuarios, donde las opiniones de la comunidad son relevantes. Una forma muy utilizada es combinar ambos tipos.

1.3. Estructura general de los Sistemas de Recomendación

Aunque el diseño de los motores de recomendación depende del **dominio** y de las características particulares de los **datos** disponibles (Melville and Sindhvani, 2010), pueden definirse los elementos generales que forman la estructura de los SR.

Las **entidades principales** que intervienen en un SR son los *usuarios* y los *ítems*. El usuario es la persona que utiliza el sistema de recomendación, ya sea para recibir

recomendaciones sobre nuevos ítems que aún no conoce o para aportarlas sobre ítems que ya conoce (Vozalis and Margaritis, 2003). Los ítems pueden entenderse como los elementos potencialmente recomendables.

Las **entradas** del sistema pueden ser de varios tipos, las más usuales son:

- Valoraciones de los usuarios (conocidas como *ratings*) que pueden ser explícitas, cuando los usuarios indican sus intereses previamente o aportan *ratings* sobre los ítems (p.e. puntuaciones numéricas o “me gusta / no me gusta”) o implícitas, también llamadas no-intrusivas, cuando se estudia el comportamiento del usuario de forma transparente según reaccione ante las recomendaciones (p.e. historiales de visitas o de selección previa).
- Datos demográficos, que recogen los datos genéricos de los usuarios, como edad, género, educación, nacionalidad, etc. suelen introducirse por ellos mismos de forma explícita.
- Datos de contexto, que consideran el contexto del usuario en el momento de recibir las recomendaciones, utilizado generalmente en entornos móviles.
- Datos de contenido, basados en análisis textual de los documentos de los ítems ya valorados por el usuario o en las características descriptivas de los ítems.

De esas entradas dependerán después los distintos tipos de algoritmos de recomendación o viceversa.

Las **salidas** del sistema pueden tomar una de las siguientes formas:

- Una *recomendación*, que es una lista formada por los K ítems que se consideran más adecuados para el usuario.
- Una *predicción*, que representa una opinión anticipada de un usuario sobre un ítem, es decir, se predice cual será el grado de satisfacción de un usuario con un ítem determinado. Esta forma de salida de los SRs también se conoce como *Individual Scoring*.

1.4. Clasificación de los Sistemas de Recomendación

Tradicionalmente, los SR se clasifican en base al **paradigma** utilizado para la selección de elementos o de filtrado de información, de acuerdo con la manera de realizar las recomendaciones:

- *Basados en contenido (BC)*. Generan recomendaciones equiparando las preferencias del usuario con las características utilizadas en la representación de los ítems ignorando la información relativa de otros usuarios. En otras palabras, se le recomendará al usuario un ítem similar al que él mismo haya elegido anteriormente o bien, haciendo un matching de sus preferencias con los ítems. Por ejemplo, si un usuario es afín a los discos de Soul, el sistema centrará sus recomendaciones en cualquier disco etiquetado con la categoría “Soul”.
- *De filtrado colaborativo (FC)*. También conocidos como sistemas de recomendación sociales. Basan sus recomendaciones en preferencias de la comunidad (valoraciones de usuarios o historial de consumo) ignorando los atributos del ítem y del usuario (por ejemplo, las consideraciones demográficas). Son un tipo de sistemas de recomendación para los que se crea un conjunto de usuarios que tienen características o gustos similares

entre sí. Si a un conjunto del grupo de usuarios les gusta un determinado ítem, es de suponer que ese mismo ítem guste al resto de usuarios de ese grupo; dicho de otro modo, se le recomendará al usuario un ítem que a las personas con los mismos gustos y preferencias, les haya gustado en el pasado.

Los investigadores Resnick y Varian (Resnick and Varian, 1997) proponen otra clasificación según los distintos criterios a considerar a la hora de realizar recomendaciones, criterios a los que llaman **dimensiones**:

- Representación y contenido de las evaluaciones: los contenidos de una evaluación pueden venir dados por un único bit (recomendado o no) o por comentarios de texto sin estructurar.
- Forma de introducción de las evaluaciones: de forma explícita o implícita.
- Identificación de la fuente: las recomendaciones pueden realizarse de forma anónima, identificando la fuente, o bien usando un pseudónimo.
- Forma de agregar las evaluaciones: cómo se van a agregar las evaluaciones sobre los ítems para generar las recomendaciones.
- Presentación de las recomendaciones: algunas recomendaciones pueden ser descartadas, las escogidas se podrían presentar, por ejemplo, ordenadas de acuerdo a las evaluaciones numéricas mostrando incluso dicha evaluación al usuario y algún tipo de justificación (*como a usted le gustó X, entonces se le recomienda Y,Z*). Las justificaciones en estos sistemas se suelen denominar “explicaciones” (Tintarev and Masthoff, 2007).

Por otro lado, Burke, en (Burke, 2002) y (Burke, 2007), propone una clasificación según las **técnicas** de recomendación considerando: (i) los datos básicos, o la base de conocimiento subyacente, (ii) los datos de entrada, o información que el usuario debe comunicar al sistema para generar recomendaciones y (iii) un algoritmo que combina los datos básicos con los datos de entrada para proponer sugerencias. En la figura 1.1 se muestran dichas técnicas de recomendación y sus fuentes de conocimiento.

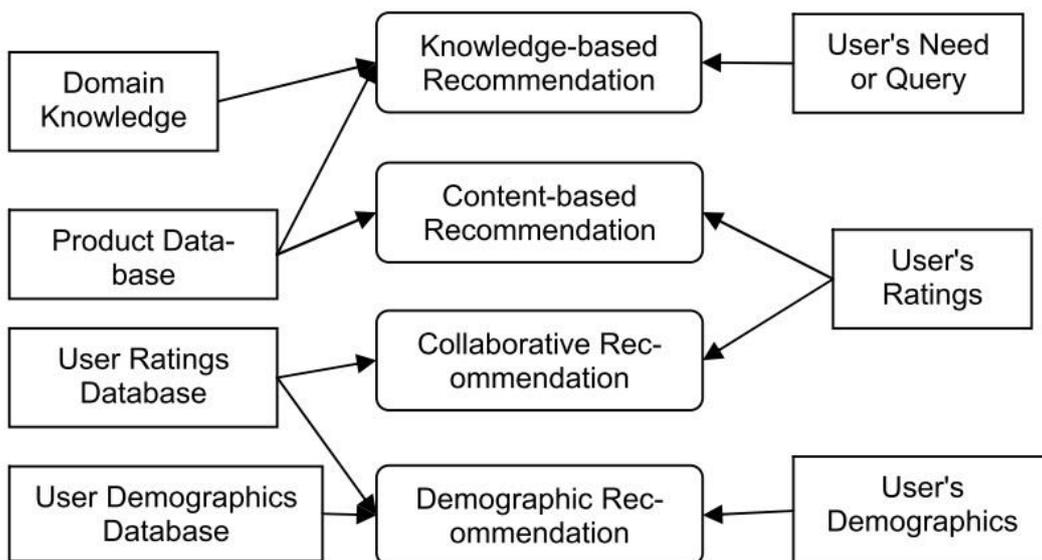


Figura 1.1. Técnicas de recomendación y sus fuentes de conocimiento. Fuente: (Burke, 2007).

Cada una de esas técnicas de recomendación viene definida como:

- *Basada en el conocimiento*, que dispone de información acerca de cómo un ítem satisface una necesidad del usuario y establece relación entre una necesidad y una recomendación. Un SR basado en conocimiento sugiere ítems basándose en inferencias sobre las necesidades y preferencias de los usuarios. Algunos sistemas han usado el razonamiento basado en casos para la recomendación.
- *Basada en el contenido*, es una técnica surgida a partir del clustering. Los objetos de interés se definen mediante sus características comunes. El sistema aprende el perfil del usuario basándose en las características de los ítems que él mismo ha valorado y ofrece recomendaciones de acuerdo con su perfil. Esta técnica y la colaborativa crean un modelo a largo plazo, que van actualizando a medida que llegan evidencias.
- *Colaborativa*, acumula valoraciones (ratings) de usuarios sobre los ítems, identifica similitudes entre usuarios en base a ellas y genera nuevas recomendaciones comparando dichas similitudes. Las valoraciones pueden recogerse de varias maneras: votaciones numéricas en un intervalo de valores, me gusta / no me gusta, comentarios sobre el ítem, etc. Su fuerza reside en que son completamente independientes de la representación de los objetos a recomendar.
- *Demográfica*, que clasifica a los usuarios en grupos y hace recomendaciones de acuerdo con el grupo. Pretenden categorizar al usuario basándose en atributos personales basados en clases demográficas.

Se puede diferenciar también entre SR centralizados (cuando las descripciones de productos y perfiles de usuarios se encuentran almacenados en un servidor centralizado) y SR no centralizados (normalmente, las descripciones se encuentran distribuidas a través de la Web).

A continuación se profundizará en los tipos de sistemas más comunes según la técnica de recomendación utilizada.

1.4.1 Sistemas de Recomendación de Filtro Colaborativo

En un SR de Filtro Colaborativo, o FC, se recomiendan ítems que a otros usuarios con gustos similares les han gustado anteriormente. Por tanto, para obtener sus recomendaciones, estos sistemas no hacen análisis alguno de los ítems, sino que se basan solamente en los términos de similitud entre los usuarios de la comunidad (Basu et al., 1998). Se crea un conjunto de usuarios que tienen características o gustos similares entre sí. Si a un conjunto del grupo de usuarios les gusta un determinado ítem, es de suponer que ese mismo ítem guste al resto de usuarios de ese grupo, como se indicó anteriormente.

También se les conoce como SR sociales (Burke, 2002).

Por lo general, siguen la siguiente estrategia:

- Los usuarios aportan valoraciones sobre ítems.
- Se analizan las valoraciones realizadas por los usuarios para establecer grupos de usuarios con preferencias similares, conocidos como “grupos de vecinos”.
- Se realizan las recomendaciones al usuario activo teniendo en cuenta sus vecinos más cercanos y sus correspondientes valoraciones.

Funcionamiento de los SR de Filtro Colaborativo

Sea m el número de usuarios $U=\{u_1, u_2, \dots, u_m\}$ y n el número de ítems $I=\{i_1, i_2, \dots, i_n\}$. Cada usuario u_i (donde $i=\{1, 2, \dots, m\}$) tiene una lista de ítems I_{u_i} para la cual ya ha expresado su opinión. Notar que I_{u_i} está contenido en I , $I_{u_i} \subset I$, siendo I_{u_i} el conjunto nulo (*null set*) cuando no se requieren dichas preferencias para todos los ítems.

Las opiniones de los usuarios normalmente vienen recogidas como *rating scores*. Un *rating* del usuario u_i sobre el ítem i_j se denota como $r_{i,j}$ donde cada *rating* es un número, o el símbolo ϕ para *no rating*. Todos estos ratings se recogen en una matriz $m \times n$ denotada como R , donde las filas contienen los ratings de un usuario sobre los diferentes ítems y las columnas tienen todos los ratings de un ítem por los diferentes usuarios. Los algoritmos se distinguen dependiendo de si van a operar con los usuarios (filas) o con los ítems (columnas) (Vozalis and Margaritis, 2003).

En la tabla 1.1 se muestra un ejemplo de una matriz R donde se almacenan los ratings de los distintos usuarios sobre los diversos ítems. El símbolo ϕ denota al conjunto nulo, es decir, no existe recomendación de esos usuarios sobre ese ítem.

Tabla 1.1. Ejemplo de Matriz de Ratings.

R	Uvas	Peras	Manzanas	Fresas	Plátanos
Juan	3	2	ϕ	8	1
María	ϕ	4	5	1	ϕ
Pedro	ϕ	6	2	ϕ	7

Por lo tanto, según se hagan los cálculos sobre los usuarios o sobre los ítems, se distinguen dos familias de algoritmos para el filtro colaborativo:

1. *Métodos basados en memoria (o en usuario)*. Son los algoritmos que trabajan sobre las filas de la matriz de ratings R , es decir, operan con los ratings hechos por el usuario actual, u_a , sobre los distintos ítems. La formación del grupo de vecinos se implementa en dos pasos: primero se calcula la similaridad³ entre todos los usuarios de la matriz R mediante métricas de proximidad y después se genera la vecindad del usuario activo para seleccionar aquellos usuarios más próximos a él.
2. *Métodos basados en modelo (o en ítem)*. En esta técnica, se observa al conjunto de ítems, denotado por I_{u_a} , que el usuario u_a ha valorado y se calcula la similaridad al ítem objetivo i_j para seleccionar los k ítems más parecidos (i_1, i_2, \dots, i_k) según sus similaridades correspondientes ($s_{i_1}, s_{i_2}, \dots, s_{i_k}$). Las predicciones pueden calcularse gracias a una media con pesos de los ratings del usuario activo sobre esos ítems similares. Se llaman también de modelo, porque en muchas ocasiones se utilizan modelos probabilísticos de aprendizaje (p.e. Redes Bayesianas, Clustering, Redes Neuronales, Lógica Difusa, etc.).

³ La palabra similaridad no está recogida en la RAE, en su lugar aparece la palabra similitud, pero aquí se emplea para aportarle una connotación matemática.

Campos de aplicación y ejemplos

Los SRs colaborativos se usan sobre todo en los entornos sociales, donde existe una comunidad de usuarios subyacente.

En redes sociales como Facebook⁴ la relación de amistad entre usuarios es tomada en cuenta para la generación de las recomendaciones porque además añade el valor de la confianza sobre quién ha valorado los ítems anteriormente. En general, en este tipo de redes se valoran además intereses generales, datos demográficos, etc.

También se usan servicios dependientes del contexto (localización temporal, espacial, acción realizada en ese instante de tiempo, etc.) donde lo importante es el contexto actual del usuario. En sistemas basados en la localización espacial, en entornos móviles y en el campo de la computación ubicua, donde la ubicación del usuario es importante, este tipo de recomendaciones son muy utilizadas. Por ejemplo, si un usuario está en el barrio del *Zaidín de Granada*, se le puede recomendar restaurantes que otros usuarios con gustos similares han valorado positivamente en el mismo barrio.

En sitios web de comercio electrónico, es donde este tipo de sistemas están más presentes. Quizás el más destacado es Amazon⁵, donde los usuarios pueden comprar y vender diferentes tipos de artículos. Amazon usa ambas técnicas de filtrado colaborativo y tiene en cuenta tanto el historial de compras y de visitas del usuario, como sus intereses implícitos o explícitos, para generar relaciones de vecindad (Linden et al., 2003), por ejemplo: “*usuarios que compraron el ítem X (comprado o visitado por el usuario actual), también compraron los ítems W; X; Y; Z*”. Amazon utiliza la recomendación para personalizar el sitio web de cada usuario con sus intereses, potenciando nuevas acciones de negocio. El gran número de ítems y de usuarios que maneja Amazon en su sistema, motivó la creación de su propio algoritmo *item-to-item* de filtro colaborativo para producir una alta calidad en las recomendaciones en tiempo real bajo ese entorno desafiante. En este algoritmo se construye una tabla de ítems similares offline de forma iterativa, que será la base de las recomendaciones. Hace matching de los ítems comprados y valorados por los usuarios con ítems similares combinándolos en una lista de recomendaciones. Los comentarios de los usuarios también son de interés a la hora de calcular las recomendaciones. Otros sitios de comercio electrónico que usan recomendaciones de filtro colaborativo son eBay⁶ o iTunes⁷.

Las recomendaciones FC son también muy típicas en sitios web que tratan temas específicos como música, libros o películas sin necesidad de un componente comercial. Ejemplos de estos sitios son Last.fm⁸, que es una estación de radio online capaz de recomendar canciones y grupos musicales, o MovieLens⁹, desarrollado en la Universidad de Minnesota, que usa el filtro colaborativo para recomendar películas agrupando a usuarios con gustos similares según sus valoraciones sobre las películas e intentando predecir la opinión individual del usuario sobre películas que aún no ha visto (Miller et al., 2003).

⁴ <http://www.facebook.com>

⁵ <http://www.amazon.com>

⁶ <http://www.ebay.com/>

⁷ <http://www.apple.com/itunes/>

⁸ <http://www.last.fm>

⁹ <https://movielens.org/>

Ventajas y desventajas

Los sistemas de recomendación de filtrado colaborativo tradicionales (o colaborativos puros) tienen algunas limitaciones:

- **Problema del nuevo usuario.** Cuando llega un nuevo usuario al sistema, su perfil no está lo suficientemente completo para que se pueda encontrar sus vecinos cercanos, por lo tanto, no se le pueden hacer recomendaciones. Existen varias mejoras para este problema, por ejemplo, mostrarle los ítems mejor valorados y que el usuario a su vez los valore.
- **Arranque en frío o cold start.** Este problema tiene varias implicaciones, y se refiere a la consideración de que sólo pueden recomendarse ítems que ya hayan sido valorados. En la etapa inicial de la utilización del sistema, resulta que la mayor parte de los ítems no han sido calificados todavía, por lo tanto no se pueden usar en los cálculos de la similitud. Otro momento en que surge este problema es con la aparición de un nuevo ítem en el sistema, cuando aún no se sabe suficiente de él como para clasificarlo, o las valoraciones de usuarios con gustos selectos que pueden perjudicar a otros, sobre todo a los más eclécticos.
- **La oveja negra o green sheep.** Se refiere a usuarios cuyos gustos no son usuales, ya que no están de acuerdo ni en desacuerdo con ningún grupo de personas de forma consistente. Estos usuarios no recibirán recomendaciones precisas en la mayoría de los casos.
- **Explicación de las recomendaciones.** En este tipo de sistemas, es más complicado que el sistema sea capaz de explicarle al usuario por qué se le está recomendando ese ítem. El usuario podría llegar a considerar las recomendaciones como mero spam.
- **Dispersión.** En ocasiones sucede que existen muchos ítems e usuarios, pero sólo son valorados unos pocos ítems, es decir, los usuarios generan pocos ratings y la matriz de recomendaciones tiene demasiados valores nulos; esto conlleva a problemas a la hora de generar los valores de proximidad y, de nuevo, las recomendaciones son débiles.
- **Confianza.** Los usuarios suelen considerar las recomendaciones que provienen de fuentes de confianza, por lo que el sistema debe poner medios para que las fuentes de las valoraciones se consideren fiables. Otro problema, son los usuarios malintencionados o los que valoran sus propios ítems por interés. Una de las soluciones propuestas a este problema es introducir valores de confianza en el sistema (O'Donovan and Smyth, 2005).
- **Calidad de las recomendaciones.** Este es un factor muy importante, y está ligado a casi todos los anteriores, sobre todo a la confianza. Una recomendación de un ítem a un usuario que a éste no le gusta, se considera un falso positivo. Si el número de falsos positivos es elevado, el usuario podría molestarse y dejar de confiar en el sistema y, por tanto, dejará de utilizarlo.

1.4.2 Sistemas de Recomendación Basados en Contenido

El enfoque de Sistemas de Recomendación Basados en Contenido (BC) tiene sus raíces en la comunidad de recuperación información (IR), y emplea muchas de sus mismas técnicas (Balabanovic and Shoham, 1997). Aunque fueron los más utilizados inicialmente, los sistemas de Filtro Colaborativo pronto tomaron más fuerza y pasaron a ser más comunes alcanzando su apogeo con el auge de la Web 2.0.

Los Sistemas de Recomendación BC recomiendan un ítem a un usuario basándose en la descripción del ítem y un perfil de los intereses del usuario (Pazzani and Billsus, 2007). En estos sistemas, se estudian las características de los ítems para que compartan una descripción común, por otro lado se generan perfiles de usuario a partir de los intereses, cualidades y actividades pasadas con esos ítems en el pasado por el usuario, es decir, se realiza un análisis de los ítems que él mismo ha visitado, seleccionado, rechazado o comprado anteriormente. Los ítems que se pueden recomendar al usuario se describen mediante un conjunto de características denominadas propiedades o atributos (Lops et al., 2011). En los BC puros (los que se tratan en este apartado), no existe el componente colaborativo, el sistema caracteriza cada patrón sin tener que hacer matching con los intereses de los otros; los ítems se recomiendan en base a la información del ítem en sí, en lugar de las preferencias de otros usuarios, lo que permite ofrecer a los usuarios explicaciones sobre las recomendaciones y aumentar la confianza del usuario en las mismas (Mooney and Roy, 2000).

Normalmente, el perfil del usuario se crea y modifica dinámicamente durante la vida del sistema en respuesta al feedback de la deseabilidad de los ítems que se le han propuesto al usuario (Pazzani and Billsus, 2007). El proceso básico del sistema BC consiste en hacer matching entre los atributos de un perfil de usuario con los atributos de un objeto de contenido (ítem) para recomendar al usuario nuevos ítems interesantes (Lops et al., 2011).

Funcionamiento de los SR Basados en Contenido

La arquitectura típica de un Sistema de Recomendación BC puede definirse como se muestra en la figura 1.2.

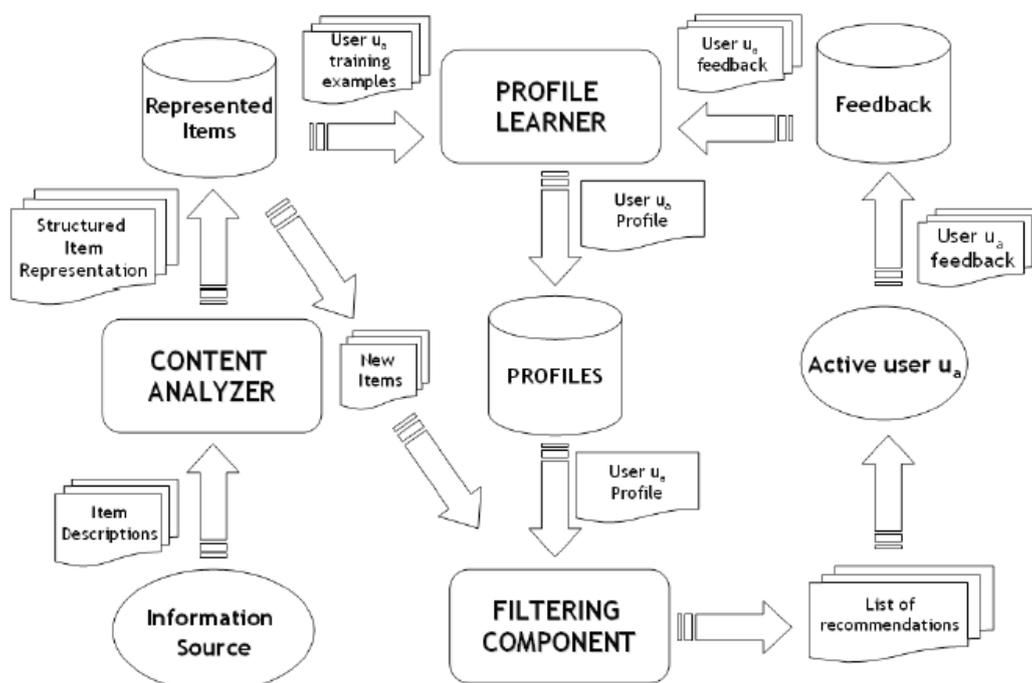


Figura 1.2. Arquitectura de un SR Basado en Contenido. Fuente: (Lops et al., 2011)

Los componentes principales son:

- **Analizador de contenido.** Pre-procesa los ítems / documentos para adecuar su contenido y estructura de cara a los siguientes pasos. Suele emplear técnicas de recuperación de

información, ya que en la mayoría de los casos se extrae la información a partir del texto. Esta información se almacena en el repositorio Ítems Representados. Como ésta será la entrada a los demás componentes del sistema, es importante decidir de qué forma van a representarse los ítems dentro del mismo.

- **Gestor de perfiles.** Recoge los datos representativos de las preferencias y trata de generalizar estos datos para construir el perfil del usuario. Normalmente se emplean técnicas de aprendizaje como estrategia de generalización. Para construir el perfil del usuario actual u_a , se define un conjunto de entrenamiento TR_a para u_a que consiste en un conjunto de pares (I_k, r_k) donde r_k es el rating del ítem I_k por u_a . A partir de ese conjunto se genera un modelo predictivo que será el perfil del usuario y se almacenará en el repositorio *Profiles*. Las técnicas de aprendizaje para la generación de perfiles más utilizadas son:
 - *Modelo espacio-vectorial* donde los perfiles y los ítems se representan como vectores de términos y se realizan cálculos de similaridad. Es la técnica más simple y es similar a la empleada por los sistemas FC.
 - *Naïve Bayes*. Especialmente utilizada en los dominios textuales. El modelo estima la probabilidad a posteriori, $P(C|d)$ de que el documento d pertenezca a una clase C . Para clasificar un documento d , se elige la clase con mayor probabilidad:

$$C = \operatorname{argmax}_{C_j} \frac{P(C_j) P(d | C_j)}{P(d)}$$
 - *Vecinos más cercanos*. A la hora de clasificar un nuevo ítem, el algoritmo utiliza una función de similaridad para compararlo con sus k vecinos más cercanos. Dependiendo del tipo de datos usados en la representación de los ítems, se usará un método u otro de esta función. Si los datos son estructurados, normalmente se utiliza una distancia euclídea y cuando se trata de clasificación de texto se emplea más la medida de similaridad del coseno (Pazzani and Billsus, 2007).
 - *Feedback relevante*. Técnica que ayuda a los usuarios a refinar consultas incrementalmente basándose en los resultados de búsqueda previos. Los usuarios van proporcionando feedback a las decisiones de los ítems recuperados según su grado de satisfacción de los mismos. Su adaptación a la recuperación de texto se hace mediante el algoritmo de Rocchio (Rocchio, 1971) que opera en el espacio de vectores.
 - *Árboles y reglas*. Emplean técnicas de árboles de decisión y reglas para generar las recomendaciones. Son muy útiles cuando se requiere una explicación del proceso de la recomendación de una lista de ítems.
- **Componente de filtrado.** Este módulo explota el perfil del usuario para sugerir los ítems relevantes que encajen con el perfil mediante una lista de ítems recomendados I_r . Las reacciones de los usuarios hacia los ítems recomendados se almacenan en el repositorio *Feedback*, esas reacciones pueden ser introducidas explícitamente (p.e. “me gusta / no me gusta”, ratings, comentarios de texto) por el usuario o bien inferirse implícitamente por el sistema a partir de las actividades del usuario.

Campos de aplicación y ejemplos

El campo de aplicación donde más se han utilizado este tipo de sistemas, sobretodo en sus orígenes, es el ámbito del filtrado y recuperación de información a partir de documentos de

texto. Se ha empleado en recomendaciones de páginas web como en Amalthea (Moukas et al., 1998), Personal WebWatcher (Mladenic, 1996) o ArgueNet (Chesnevar and Maguitman, 2004).

Algunos ejemplos de SR basados en contenido son: LIBRA (Mooney and Roy, 2000) que recomienda libros a usuarios explotando las descripciones de los libros en el sitio web de Amazon y usa algoritmos probabilísticos Naïve Bayes para la gestión de perfiles de usuarios. ACR News (Mobasher et al., 2000) donde se recomiendan noticias en Usenet a partir de su contenido. INTIMATE (Mak et al., 2003) recomienda películas usando técnicas de categorización para aprender de las sinopsis de las películas del Internet Movie Database IMDB¹⁰.

También existen sistemas basados en contenido para recomendar música a los usuarios. Uno de ellos es Pandora¹¹, donde los usuarios construyen su “estación” de radio online basándose en intereses musicales indicando canciones o artistas que le gustan. A partir de estas preferencias, Pandora reproduce canciones similares que al usuario le podrían gustar y el usuario puede hacer feedbacks sobre esas canciones indicando si le gustan o no (“pulgar arriba / pulgar abajo”).

Un sistema algo diferente, ya que introduce la noción de grupos de usuarios y tiene en cuenta la localización de los mismos, es Pocket Restaurant Finder (McCarthy, 2002), que sugiere alternativas a un grupo de personas teniendo en cuenta las preferencias de sus miembros. Está diseñado para utilizarse en contextos localizados, por ejemplo, cuando un grupo de asistentes a una conferencia intentan decidir un restaurante para cenar. Las recomendaciones del sistema, por lo tanto, se basarán en la localización física del grupo de usuarios y en los gustos culinarios de sus miembros.

Ventajas y desventajas

El paradigma basado en contenido tiene algunas ventajas respecto a los colaborativos:

- **Independencia del usuario.** Lo importante son las valoraciones del propio usuario y no las de los demás, lo cual es apropiado para cierto tipo de sistemas donde influyen determinado tipo de gustos, como el sabor, por ejemplo, si al usuario actual le gusta el pepino y a otro usuario que le gusta el pepino también le gusta el pimiento no implica que al usuario actual le tenga que gustar el pimiento.
- **Transparencia.** Se puede aportar explicaciones a las recomendaciones explicando el proceso seguido, o las características del contenido que han llevado a ese ítem a la lista de recomendaciones. Además de aportar mayor confianza al usuario, le ayudan en la decisión de seguir la recomendación o no.
- **Nuevos ítems.** Se puede recomendar ítems que no hayan sido valorados anteriormente por usuario alguno, por lo que el problema del arranque en frío se amortigua pero, como se mostrará a continuación, sigue existiendo.
- **Menos intrusivos.** Permite construir recomendaciones de forma menos intrusiva que los sistemas de filtro colaborativo, observando el comportamiento del usuario para construir su perfil sin molestarle para que esté continuamente valorando recomendaciones de forma explícita (Middleton et al., 2004).

¹⁰ IMDB <http://www.imdb.com>

¹¹ PandoraInternetRadioandMusicGenomeProject.<http://www.pandora.com>

Aún así los SR basados en contenido tienen varios inconvenientes:

- **Contenido del análisis limitado.** Sus técnicas tienen un límite natural asociado al número de características que se le asocian y algunas de ellas pueden quedar fuera del análisis y no tenerlas en cuenta. Además, generalmente, es necesario conocimiento experto del dominio capaz de obtener y clasificar dichas características.
- **Nuevo usuario.** En este caso, está relacionado con el problema del arranque en frío. Cuando llega un usuario nuevo al sistema, aún no se cuenta con información relativa a los ítems que ha visitado en el pasado, por lo que inicialmente, estas características no pueden tenerse en cuenta para las recomendaciones. De hecho, se necesita que el usuario haya aportado varias valoraciones en el sistema para que se le pueda empezar a recomendar ítems de forma más precisa y acertada.
- **Sobre-especialización o serendipia.** Al usuario se le van a recomendar cosas según lo que haya visitado y valorado en el pasado, pero esto puede llevar a que nuevos ítems, que podrían gustarle potencialmente, nunca se le muestren porque no ha visitado objetos similares, es decir, se mueve en un rango de ítems con características parecidas y los demás pueden quedar ocultos.

1.4.3 Sistemas de Recomendación Basados en Conocimiento

Estos sistemas utilizan el conocimiento acerca de los usuarios y de los productos aplicando un enfoque basado en el conocimiento para la generación de una recomendación, razonando sobre qué productos cumplen con los requisitos de los usuarios (Burke, 2000). Este razonamiento utiliza técnicas de inferencia previamente conocidas, como el razonamiento basado en casos (Bridge et al., 2005), o el basado en restricciones (Felfernig et al., 2006). La base de conocimiento de estos sistemas suele ser extensa puesto que no sólo se almacena conocimiento sobre usuarios y productos sino también sobre sus dominios.

Funcionamiento de los SR Basados en Conocimiento

Los SR basados en Conocimiento, explotan el conocimiento intensivo sobre el dominio del producto y del usuario al hacer recomendaciones. A diferencia del filtro colaborativo y basado en contenido, este paradigma usa representaciones explícitas acerca del conocimiento de los ítems y los usuarios que se obtiene durante la fase de elicitación de requisitos (Felfernig et al., 2006).

El razonamiento basado en casos recupera casos pasados similares ya resueltos y los reutiliza para la solución del problema actual. En los SR puede traducirse como la búsqueda de ítems similares a los que resultaron más satisfactorios al usuario en el pasado (Lorenzi and Ricci, 2005).

En el enfoque basado en restricciones se define explícitamente un conjunto de reglas de recomendación que mapean los requerimientos del usuario a las características de los ítems (Ge et al., 2010) mediante implicaciones lógicas del tipo: “Si el usuario requiere X, el ítem a recomendar debe tener la característica X’”. Se definen diferentes restricciones sobre la base de conocimiento (Felfernig et al., 2006), por ejemplo:

- Restricciones de compatibilidad: son restricciones sobre las posibles combinaciones de los requerimientos de usuario.

- Restricciones de filtro: delimitan los valores que puede tomar una propiedad de un ítem según un requisito del usuario.
- Pesos para indicar la importancia de cada requerimiento del usuario.

Se recomiendan los ítems que ofrecen mayor grado de *satisfacibilidad* de restricciones.

Campos de aplicación y ejemplos

Resulta complicado encontrar sistemas de recomendación clasificados exclusivamente como basados en conocimiento. En su lugar, la mayoría de los sistemas que usan técnicas de recomendación basadas en conocimiento suelen emplear otro tipo de técnicas de recomendación, como la colaborativa o la basada en contenido, dando lugar a los conocidos como sistemas híbridos (apartado 1.4.4).

Ventajas y desventajas

Las representaciones explícitas previas acerca de los ítems y los usuarios permiten evitar algunos de los problemas previamente citados como el del nuevo ítem y el del nuevo usuario (Felfernig et al., 2006). Otra ventaja es que no requieren de las valoraciones del usuario para su funcionamiento (Lorenzi and Ricci, 2005). Sin embargo, el uso exclusivo de técnicas de recomendación basadas en conocimiento tradicionales suele derivar en recomendaciones incompletas, por lo que dichas técnicas suelen combinarse con otras en los denominados Sistemas Híbridos explicados en el siguiente apartado.

1.4.4 Sistemas Híbridos

Los SR citados previamente utilizan técnicas bien estudiadas, como las colaborativas, las basadas en contenido y la basadas en conocimiento. Cada una de ellas tiene sus propias ventajas y sus debilidades. Su uso es el más extendido dentro del ámbito de los sistemas de recomendación. Esto es debido a que las carencias de uno son contrarrestadas con características del otro y los beneficios de ambos potencian el funcionamiento del sistema y, por ende, la calidad de las recomendaciones (Vozalis and Margaritis, 2003) y (Burke, 2000).

En la figura 1.3 se muestran los diferentes tipos de recomendación que se utilizan según la fuente de conocimiento. Puede apreciarse que el componente social no está relacionado con las fuentes basadas en conocimiento, así como las opiniones de los usuarios sobre los ítems o su comportamiento.

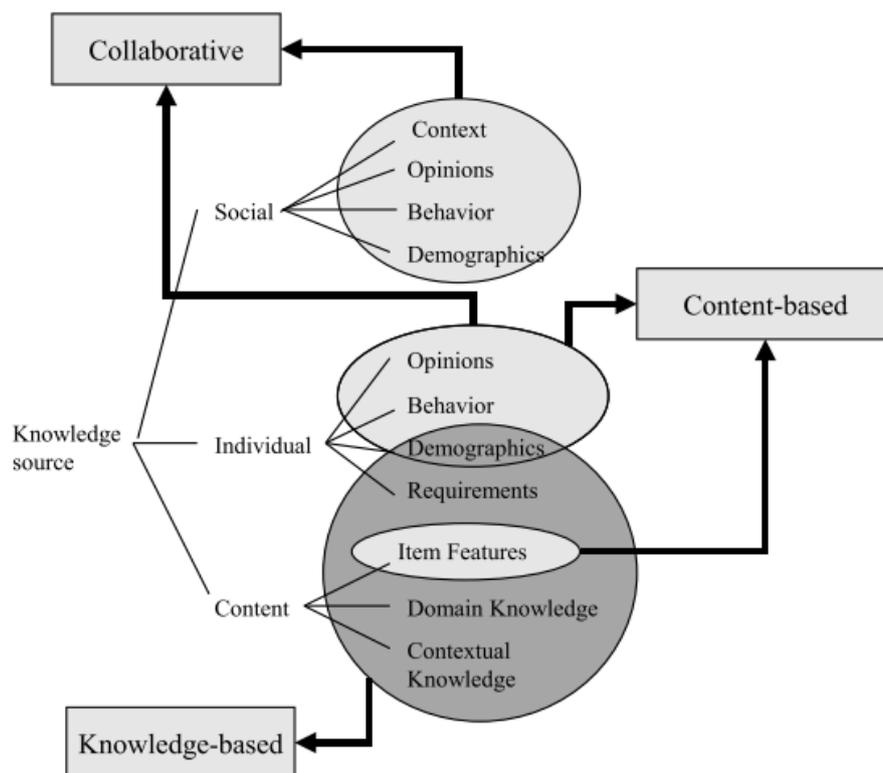


Figura 1.3. Conexión entre las fuentes de conocimiento y los tipos de recomendación. Fuente: (Burke, 2007).

Funcionamiento de los SR Híbridos

Burke (Burke, 2002) distingue varios métodos de hibridación para combinar unas técnicas con otras dentro de un mismo sistema:

- Ponderación (*weighted*). Un SR híbrido ponderado es aquel donde las puntuaciones de un ítem recomendado se calculan a partir de los resultados de todas las técnicas presentes en el sistema. Es decir, se combinan las puntuaciones de varias técnicas de recomendación para obtener una única recomendación.
- Intercambio (*switching*). El sistema intercambia de una técnica a otra dependiendo de la situación del sistema en ese momento.
- Mezcla (*mixed*). Se presentan a la vez distintas recomendaciones de diferentes sistemas de recomendación.
- Combinación de características (*feature combination*). En un único algoritmo de recomendación, se integran características de diferentes fuentes de recomendación.
- Cascada (*cascade*). Un sistema de recomendación refina las recomendaciones dadas por otro.
- Aumento de características (*feature augmentation*). Se usa la salida de una técnica como entrada de otra técnica.
- Meta-nivel (*meta-level*). El modelo aprendido por un sistema es usado como entrada de otro sistema.

Campos de aplicación y ejemplos

FAB (Balabanovic and Shoham, 1997) es uno de los primeros sistemas de recomendación híbridos y uno de los más referenciados. Es una implementación distribuida de un SR híbrido formando parte del Proyecto de Biblioteca Digital de la Universidad de Stanford. El proceso de recomendación se divide en dos etapas: colección de ítems para formar una base de datos manejable y la selección de esos ítems para usuarios particulares. Los ítems a recomendar son páginas web de posible interés. En la etapa de recolección de ítems, los agentes recolectores, recopilan páginas de la web según un topic. Basándose en los perfiles de usuario, el agente recolector de búsqueda selecciona aquellos que se correspondan con los temas del perfil del usuario. El agente recolector de índice construye consultas para los motores de búsqueda web previamente indexados. Considera la construcción de perfiles precisos, como una tarea imprescindible para un funcionamiento adecuado del sistema, afirmando que el éxito de FAB depende de la habilidad de los perfiles aprendidos para representar los intereses actuales de los usuarios. Perfiles precisos, permiten al componente basado en contenido asegurarse de que las recomendaciones son apropiadas y al componente colaborativo asegurarse de que usuarios con perfiles similares son, efectivamente, similares.

Basu y colaboradores (Basu et al., 1998) proponen un sistema de recomendación híbrido de películas cuya base de conocimiento es la base de datos de películas del Internet Movie Database IMDB. Formalizan el problema de la recomendación de películas como un problema de aprendizaje tal que, para un usuario y una película, se produzca una salida entre los valores le ha gustado o no le ha gustado. Las recomendaciones tratan de predecir esos valores. Para lograrlo, cuentan con valoraciones de usuarios sobre películas (del 1 al 10) y utilizan las características de las películas (género, actores, director, país de origen, etc.) para emplear las técnicas basadas en contenido. Los factores híbridos del sistema surgen al definir nuevas características colaborativas que están influenciadas por el contenido.

1.5. Limitaciones de los SR tradicionales y nuevas tendencias

En cada dominio se presentan diferentes problemas a los que hay que dar diferentes soluciones. Además, la capacidad de evolución se ha convertido en necesaria, lo que ha llevado a la diversificación de estos sistemas (Peis, 2008). Las limitaciones mencionadas no quedaban resueltas completamente, sino que se podían paliar con el uso de sistemas híbridos, que trataban de combinar las diferentes técnicas para contrarrestar las deficiencias y complementar sus ventajas.

Analizando el conjunto de los sistemas de recomendación estudiado anteriormente, se ha comprobado la importancia de los perfiles de usuario para poder ofrecer recomendaciones que se adapten a los gustos o necesidades de los usuarios. Estos perfiles son intrínsecos a cada sistema y se empiezan a construir a partir de la primera interacción del usuario, por lo que necesita que éste lo use varias veces para que se pueda poblar el perfil. Esto limita la comprensión del usuario por parte del sistema. Un perfil de usuario potente, debería ser persistente y mejor aún, común a las diversas aplicaciones o agentes en los que el usuario esté involucrado. Cuanto más se conozca al usuario, se consigue un mayor grado de personalización, y, por lo tanto, mejor satisfacción en su interacción con los diversos sistemas, lo que conllevará a mejores recomendaciones.

Otro aspecto importante a mejorar, es la capacidad de representación de los ítems de forma que puedan ser obtenidos y comprendidos por otros sistemas, aplicaciones o agentes, sobretodo de dominios comunes. La mayoría de los sistemas de recomendación tradicionales utilizan bases de datos independientes, donde almacenan las propiedades de los ítems que se van a recomendar. Aparece así el problema de la heterogeneidad de los datos y de la representación de la información. Esta representación no permite la comunicación eficiente entre sistemas, por lo que se dificulta la interoperabilidad entre los distintos agentes que podrían trabajar juntos para ofrecer mejores recomendaciones en dominios similares impidiendo, por tanto, la reutilización de la información por otros procesos o aplicaciones.

Para solventar los problemas de heterogeneidad de los datos y comunicación entre aplicaciones, hay que solucionar algunas de las limitaciones de semántica que surgen, como la sinonimia o la polisemia. Por ejemplo, a la hora de realizar matchings entre los ítems y los usuarios se podrían pasar por alto ítems potencialmente interesantes, pero cuyo nombre de cierta/s característica/s no coincide con la/s búsqueda/s, aunque sean la/s misma/s.

Es debido a este tipo de cuestiones por lo que surge la necesidad de nuevas propuestas para el desarrollo de sistemas de recomendación que vayan más allá y que traten de resolver estas limitaciones. Una de las propuestas con más fuerza en los últimos años son los sistemas de recomendación semánticos, que emplean la tecnología de la Web Semántica (Berners-Lee et al., 2001) como base de su desarrollo. En el capítulo siguiente se introducirán los fundamentos de la Web Semántica y sus tecnologías para dar paso en el siguiente capítulo a un análisis de los SR semánticos y cómo emplean dichas tecnologías para resolver, entre otras, las limitaciones de heterogeneidad e interoperabilidad de los sistemas de recomendación desarrollados con anterioridad, aumentando la calidad de las recomendaciones.

Referencias del capítulo

- BALABANOVIC, M. & SHOHAM, Y. 1997. Fab: content-based, collaborative recommendation. *Communications in the ACM*, 40, 66–72.
- BASU, C., HIRSH, H. & COHEN, W. Recommendation as Classification: Using Social and Content-Based Information in Recommendation. Fifteenth National Conference on Artificial Intelligence, 1998. AAAI Press, 714–720.
- BERNERS-LEE, T., HENDLER, J. & LASSILA, O. 2001. The semantic web. *Scientific american*, 284, 28-37.
- BRIDGE, D., GÖKER, M. H., MCGINTY, L. & SMYTH, B. 2005. Case-based recommender systems. *The Knowledge Engineering Review*, 20, 315-320.
- BURKE, R. 2000. Knowledge-based Recommender Systems. In: KENT, A. (ed.) *Encyclopedia of Library and Information Systems*. New York: Marcel Dekker.
- BURKE, R. 2002. Hybrid Recommender Systems: Survey and Experiments. *User Modeling and User-Adapted Interaction*, 12, 331–370.
- BURKE, R. 2007. Hybrid web recommender system. *The adaptive web*. Springer Berlin Heidelberg.
- CHESNEVAR, C. I. & MAGUITMAN, A. G. ArgueNet: An Argument-Based Recommender System for Solving Web Search Queries. 2nd IEEE Intelligent Systems Conference, IS-2004, 2004. 282–287.
- FELFERNIG, A., FRIEDRICH, G., JANNACH, D. & ZANKER, M. 2006. An integrated environment for the development of knowledge-based recommender applications. *International Journal of Electronic Commerce*, 22, 11-34.

- GE, M., DELGADO-BATTENFELD, C. & JANNACH, D. Beyond accuracy: evaluating recommender systems by coverage and serendipity. Proceedings of the fourth ACM conference on Recommender systems 2010. ACM, 257-160.
- LINDEN, G., SMITH, B. & YORK, J. 2003. Amazon.com recommendations: item-to-item collaborative filtering. *Internet Computing, IEEE*, 7, 76-80.
- LOPS, P., GEMMIS, M. D. & SEMERARO, G. 2011. Content-based recommender systems: State of the art and trends. *Recommender systems handbook*. Springer US.
- LORENZI, F. & RICCI, F. 2005. Case-based Recommender Systems: a unifying view. *Intelligent Techniques for Web Personalization*. Springer Berlin Heidelberg., 89-113.
- MAK, H., KOPRINSKA, I. & POON, J. INTIMATE: A web-based movie recommender using text categorization. IEEE/WIC International Conference on Web Intelligence, 2003. 602-605.
- MASSA, P. & AVESANI, P. Trust-aware Collaborative Filtering for Recommender Systems. Federated Int. Conference On The Move to Meaningful Internet: CoopIS, DOA, ODBASE., 2004. 492-508.
- MCCARTHY, J. F. Pocket Restaurant Finder: A situated recommender systems for groups. Workshop on Mobile Ad-Hoc Communication at the 2002 ACM Conference, 2002. Human Factors in Computer Systems.
- MELVILLE, P. & SINDHWANI, V. 2010. Recommender systems. *Encyclopedia of machine learning*. Springer US.
- MIDDLETON, S. E., SHADBOLT, N. R. & DE ROURE, D. C. 2004. Ontological user profiling in recommender systems. *ACM Transactions on Information Systems (TOIS)*, 22, 54-88.
- MILLER, B. N., ALBERT, I., LAM, S. K., KONSTAN, J. A. & RIEDL, J. MovieLens unplugged: experiences with an occasionally connected recommender system. 8th international conference on Intelligent user interfaces, IUI '03, 2003 New York, USA. ACM, 263-266.
- MLADENIC, D. 1996. Personal WebWatcher: design and implementation. Department of Intelligent Systems, J.Stefan Institute, Slovenia.
- MOBASHER, B., COOLEY, R. & SRIVASTAVA, J. 2000. Automatic personalization based on Web usage mining. *Communications in ACM*, 43, 142-151.
- MOONEY, R. J. & ROY, L. Content-based book recommending using learning for text categorization. Fifth ACM conference on Digital libraries, DL '00, 2000 New York, USA. ACM, 195-204.
- MOUKAS, A., MOUKAS, R. & MAES, P. 1998. Amalthea: An Evolving Multi-Agent Information Filtering and Discovery System for the WWW. *Autonomous Agents and Multi-Agent Systems*, 59-88.
- O'DONOVAN, J. & SMYTH, B. Trust in Recommender Systems. 10th international conference on Intelligent user interfaces, IUI '0, 2005 New York, USA. ACM, 167-174.
- PAZZANI, M. & BILLSUS, D. 2007. Content-based recommendation systems. *The Adaptive Web*. Lecture Notes in Computer Science, Springer Berlin, Heidelberg.
- PEIS, E., DEL CASTILLO, J. M., & DELGADO-LÓPEZ, J. A. 2008. Semantic recommender systems. Analysis of the state of the topic. *Hipertext.net*, 6, 1-5.
- RESNICK, P. & VARIAN, H. R. 1997. Recommender systems. . *Communications of the ACM*, 40, 56-58.
- ROCCHIO, J. 1971. Relevance Feedback in Information Retrieval. *The SMART System: Experiments in Automatic Document Processing*. Prentice Hall.
- SARWAR, B., KARYPIS, G., KONSTAN, J. & RIEDL, J. Application of dimensionality reduction in recommender system-a case study Proceedings of the 6th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining, Workshop on Web Mining for E-Commerce -- Challenges and Opportunities 2000.

- TINTAREV, N. & MASTHOFF, J. A survey of explanations in recommender systems. 23rd International Conference on Data Engineering Workshop, 2007. IEEE, 801-810.
- VOZALIS, E. & MARGARITIS, K. G. 2003. Analysis of recommender systems algorithms. *The 6th Hellenic European Conference on Computer Mathematics & its Applications*, 732-745.
- WANG, P. 1998. Why recommendation is special? *Papers from the 1998 Workshop on Recommender Systems, part of the 15th National Conference on Artificial Intelligence*.

CAPÍTULO 2

WEB SEMÁNTICA, TESAUROS Y ONTOLOGÍAS

La Web ha cambiado profundamente la forma en la que nos comunicamos, trabajamos, nos relacionamos e incluso hacemos negocios. Tenemos acceso a millones de recursos, independientemente de nuestra situación geográfica e idioma y podemos realizar transacciones económicas a través de Internet. La comunicación prácticamente con todo el mundo en cualquier momento y a bajo coste es posible hoy en día. Sin embargo, al mismo tiempo, estos factores que han propiciado el éxito de la Web, también han originado sus principales problemas: la sobrecarga de información y la heterogeneidad de las fuentes de información. La Web Semántica surge para intentar solventar estas limitaciones, con la idea básica de que es necesario un cambio en el enfoque de la Web dotándola de más significado, de modo que las aplicaciones sean capaces de “comprender” y comunicarse entre sí. Las ontologías son consideradas el corazón de la Web Semántica, ya que son capaces de aportar el significado. En este capítulo se introducirán los principios de la Web Semántica y las ontologías, su evolución, arquitectura y tecnologías, y se verá cómo la definición de ontologías mediante el lenguaje OWL permite el razonamiento sobre su contenido para descubrir nuevo conocimiento.

2. 1. Introducción la Web Semántica y su Arquitectura

La continua y veloz evolución de la Web ha llegado a cambiar muchos de los hábitos de las personas en poco tiempo. Es posible realizar gran parte de nuestras compras o ventas sin salir de casa, a través de Internet. Podemos comunicarnos casi instantáneamente con otros sin apenas esfuerzo ni gasto económico. La mayor parte de la información que “existe” en el mundo, también “existe” en la Web y es, en gran medida, fácilmente accesible. Pero todos estos factores, a la vez que han impulsado el uso de Internet entre cada vez más gente y que han propiciado su éxito, también han puesto de manifiesto sus principales limitaciones:

- **La sobrecarga** de información: automáticamente nos lleva a pensar en los inconvenientes a la hora de decidir cuál es la información que nos interesa y si la fuente o la misma información puede ser considerada como de confianza.
- **La heterogeneidad** de las fuentes de información: conlleva consiguientes problemas de interoperabilidad y reutilización.

La Web Semántica (Berners-Lee et al., 2001) ayuda a resolver estos dos importantes problemas permitiendo a los usuarios delegar tareas en el software. Gracias a la semántica en la Web, el software es capaz de procesar su contenido, razonar con este, combinarlo y realizar deducciones lógicas para resolver automáticamente problemas cotidianos.

La Web Semántica es una extensión de la Web tradicional que permite a las máquinas interpretar el significado de los datos, en base a anotaciones semánticas basadas en descripciones ontológicas. Fue ideada por Tim Berners Lee, considerado el padre de la Web (creador de HTML, HTTP y URL), e impulsado por el organismo internacional de estandarización de tecnologías Web, W3C, que él mismo dirige. Al dotar a la Web de más significado y, por lo tanto, de más semántica, se pueden obtener soluciones a problemas

habituales en la búsqueda de información gracias a la utilización de una infraestructura común, mediante la cual, es posible compartir, procesar y transferir información de forma sencilla. Esta Web extendida y basada en el significado, se apoya en lenguajes universales que resuelven los problemas ocasionados por una Web carente de semántica en la que, en ocasiones, el acceso a la información se convierte en una tarea difícil y frustrante. La Web Semántica como infraestructura basada en metadatos (datos sobre los datos) aporta un camino para razonar en la Web, extendiendo así sus capacidades.

Los tesauros y las ontologías forman parte de la Web Semántica. Son utilizados para aportar significado a los datos y favorecen el uso y gestión de sus metadatos.

Arquitectura de la Web Semántica

Desde sus comienzos, se ha estado desarrollando una arquitectura en capas que ha sufrido diversas variaciones a partir de la original propuesta por Tim Berners-Lee. Esta arquitectura a menudo se representa mediante un diagrama al que se conoce como la Pila de Tecnologías de la Web Semántica (Semantic Web Stack) e identifica el conjunto de tecnologías, herramientas y normas básicas que apoyan la visión de una Web dotada de significado. En la figura 2.1 se muestra la última versión (2007) de la Pila de Tecnologías¹².

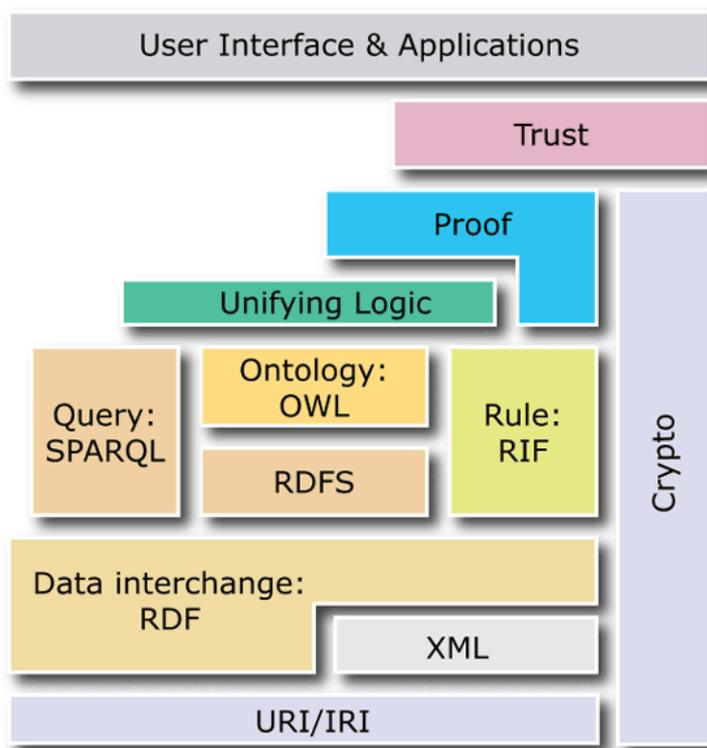


Figura 2.1. Pila de Tecnologías de la Web Semántica. Fuente: W3C (<http://www.w3.org/2001/sw/>).

A la hora de definir dichas capas se suele distinguir como bloques diferenciados entre:

1. **Tecnologías Hipertexto:** las capas inferiores contienen tecnologías conocidas de hipertexto en la web, sin necesidad de cambios proporcionan la base para la Web Semántica.

¹² Semantic Web Stack (Layer Cake) <http://www.w3.org/2007/03/layerCake.png>.

- IRI (*Internationalized Resource Identifier*), es una extensión de la URI (*Uniform Resource Identifier*) y proporciona medios para la identificación exclusiva de los recursos de la Web Semántica, que necesita usar el identificador único de los recursos para permitir su manipulación en capas superiores.
 - XML es un lenguaje de marcado que permite la creación de documentos compuestos de datos estructurados. La Web Semántica da significado (semántica) a los datos estructurados.
2. **Tecnologías de Web Semántica Estándar:** estas capas son creadas exclusivamente para la manipulación de datos en la Web Semántica.
- RDF (W3C, 2014a) es un modelo de datos que proporciona información descriptiva simple sobre los recursos que se encuentran en la Web. Se describen los recursos mediante la creación de declaraciones en forma de los llamados triples RDF. Esto permite representar la información sobre los recursos en forma de gráfico.
 - RDF-Schema ó RDFS (W3C, 2014b) proporciona un vocabulario básico para RDF permitiendo crear jerarquías de clases y propiedades.
 - OWL (OWL Working Group, 2012) es un lenguaje para desarrollar temas o vocabularios específicos en los que asociar esos recursos. Extiende a RDFS añadiendo construcciones más avanzadas para describir la semántica de las sentencias o declaraciones RDF. Está basada en la lógica descriptiva y acerca el poder del razonamiento a la Web Semántica. Se utiliza para definir ontologías estructuradas que pueden ser utilizadas a través de diferentes sistemas.
 - SPARQL (W3C, 2012b) es lenguaje de consulta sobre RDF; puede usarse para consultar cualquier dato basado en RDF (por tanto, también las declaraciones que contengan RDFS y OWL). Este lenguaje de consultas se emplea para recuperar información en las aplicaciones de Web Semántica.
3. **Tecnologías de Web Semántica No Realizadas:** estas capas contienen tecnologías que aún no están estandarizadas o que contienen sólo ideas de lo que debería implementarse.
- RIF (W3C, 2013): aporta el soporte a las reglas. RIF (Rule Interchange Format) es un lenguaje XML para expresar reglas que los ordenadores pueden ejecutar. SWRL es un lenguaje de reglas que fue diseñado como extensión de OWL y permite escribir relaciones que no pueden ser descritas directamente mediante la lógica de descripción de OWL. RIF fue diseñado como un formato para el intercambio de reglas entre los sistemas de reglas, por ejemplo los sistemas que implementan SWRL.
 - Crypto es la capa de criptografía. Hay que verificar que las declaraciones provienen de fuentes confiables. En la actualidad, el W3C ha desarrollado una API Javascript para llevar a cabo operaciones de criptografía básicas, así como una API para que las aplicaciones gestionen el material necesario para la generación de llaves necesarias para dichas operaciones. Aunque esta capa se encuentra en estado avanzado aún se considera “no estable” (W3C, 2014c).
 - Trust es la capa de confianza. Conlleva la verificación de que las declaraciones vienen de fuentes de confianza. El término empleado en este ámbito es el de *Provenance* que se refiere a las fuentes de información, como entidades y procesos que forman parte de la producción o envío de *artefactos* (representaciones de datos semánticos). El uso de metadatos es una parte importante de esta capa.
 - La interfaz de usuario permite a los humanos usar las aplicaciones semánticas.

Para obtener una adecuada definición de los datos, la Web Semántica utiliza esencialmente RDF, SPARQL y OWL, mecanismos que ayudan a convertir la Web en una infraestructura global en la que es posible compartir y reutilizar datos y documentos.

2. 2. Tesauros y Ontologías

Un **Tesauro** es una lista de palabras o términos controlados para representar conceptos. Comprende una lista de términos preferidos ordenados de forma alfabética, temática y jerárquica, una lista de sinónimos de dichos términos indicando cual es su término preferido y las definiciones de dichos términos. Además de la relación de sinonimia, los tesauros suelen contener otras relaciones de equivalencia como la homonimia, antonimia y polisemia, por lo que pueden verse como vocabularios controlados en un dominio. Una de las finalidades más extendidas en el ámbito computacional es la de indexar documentos y recuperar información en el área de los Sistemas de Información (Gilchrist, 2003).

La ISO 2788 proporciona las siguientes propiedades para describir términos (Garshol, 2004):

- BT (Broader Term): indica su término superior en la jerarquía, por lo general será un concepto menos específico. También es posible el uso de NT (Narrower Term) para indicar la relación inversa.
- SN (Scope Note): es una cadena que indica el significado del término dentro del tesauro.
- USE: se utiliza para indicar cuál es el término preferido en lugar de éste. Implica que ambos conceptos son sinónimos.
- TT (Top Term): indica su ancestro más superior en la jerarquía. Esta información aunque es redundante puede resultar útil.
- RT (Related Term): indica una relación con otro término diferente a las anteriores.

Una **Ontología** es una herramienta que permite organizar, reutilizar y explotar el conocimiento. Aunque la historia de la ontología se remonta a los tiempos de la filosofía aristotélica en su intento de clasificar todo lo existente en el universo, su significado ha ido cambiando a lo largo del tiempo y en la actualidad está ganando un rol específico en los dominios de la representación y la Ingeniería del Conocimiento. En el sentido filosófico, podríamos referirnos a una ontología como un sistema particular de categorías para una cierta visión del mundo independientemente del lenguaje, por otro lado, en el campo de la Inteligencia Artificial, y de la informática en general, una ontología se refiere a un artefacto de ingeniería formado por un vocabulario específico para describir una cierta realidad más una serie de suposiciones explícitas aclarando el significado pretendido de las palabras del vocabulario (Guarino, 1998). Las motivaciones para el desarrollo de ontologías, son muy variadas, pero, sobre todo dentro de la comunidad investigadora y científica, se podrían recoger las siguientes (Noy and McGuinness, 2001):

- **Compartir una comprensión común de la estructura de información entre gente o agentes.** Por ejemplo, supongamos que varios sitios Web contienen información médica o que prestan servicios de comercio electrónico médicos. Si estos sitios web comparten y publican la misma ontología subyacente de los términos que usan, entonces los agentes pueden extraer y agregar información a dichos sitios. Además, los agentes pueden usar esta

información agregada para responder a consultas de los usuarios o como datos de entrada a otras aplicaciones.

- **Permitir la reutilización del dominio de conocimiento.** Si un grupo de personas desarrollan una ontología sobre un dominio específico, otros pueden usarla en su actividad sobre el mismo dominio o incluso en otros dominios. Esta es una de las motivaciones más fuertes en el desarrollo de ontologías y donde se han hecho muchos esfuerzos de investigación y recomendaciones.
- **Hacer explícitas las asunciones sobre el dominio.** Permite la independencia de los lenguajes de programación y que los cambios en el conocimiento puedan también realizarse en la ontología. Además, las especificaciones explícitas de conocimiento del dominio son útiles para los usuarios nuevos que tienen que aprender el significado de los términos en un dominio.
- **Separar entre el conocimiento del dominio y el conocimiento operacional.** Permite al conocimiento operacional alimentarse del conocimiento ontológico de un dominio.
- **Analizar el conocimiento del dominio.** El análisis formal de los términos de una ontología es extremadamente valioso cuando se trata de reutilizar ontologías existentes y ampliarlas.

La principal diferencia entre tesauros y ontologías radica en su finalidad. Mientras que la finalidad de un tesoro es la de indexar documentos y recuperar información, la de una ontología es la de compartir conocimiento consensuado y razonar sobre ese conocimiento para obtener nuevo conocimiento.

2.2.1 Definiciones y Clasificación de Ontologías

El término *Ontología* comenzó a usarse en el campo de la ingeniería del conocimiento en los años 80, con el propósito de combinar la representación del conocimiento con las capacidades de razonamiento proporcionadas por el uso de distintos tipos de lógicas matemáticas, además de potenciar la necesidad de compartir y reutilizar el conocimiento. Aunque sus inicios están ligados al campo de la Inteligencia Artificial, actualmente las ontologías se emplean en gran cantidad de campos distintos: Web Semántica, modelado de sistemas, comercio electrónico, etc. (McGuinness, 2003), y se utiliza en multitud de disciplinas como la medicina, la educación o el derecho. Consultando el Diccionario de la Real Academia de la Lengua Española se encuentra la siguiente definición de ontología: *1. f. Parte de la metafísica que trata del ser en general y de sus propiedades trascendentales*. Definición que no aporta mucho al tema que nos concierne.

En la literatura se pueden encontrar otras definiciones que sin perder su acepción filosófica pueden aplicarse también al campo de la informática como afirma Guarino (Guarino, 1998) al describirla como un *sistema particular de categorías que refleja cierta visión del mundo*. A continuación se exponen algunas de las definiciones más utilizadas en la literatura:

- Una de las definiciones más citadas en este campo es la de Gruber (Gruber, 1995): *An Ontology is an explicit specification of a conceptualization*. Afirmando que en la Inteligencia Artificial, lo que existe es lo que puede ser representado. Entiende como conceptualización: los objetos, conceptos y otras entidades que se asumen como existentes en algún área de interés y las relaciones entre ellos. La conceptualización es una forma abstracta y simplificada de ver el mundo que se quiere representar con algún propósito.
- La definición anterior es extendida por Borst (Borst, 1997): *An Ontology is a formal specification of a shared conceptualization*.

- Studer y colaboradores (Studer et al., 1998), la amplían nuevamente utilizando las dos anteriores: *An Ontology is a formal, explicit specification of a shared conceptualization*. Entendiendo como conceptualización un modelo abstracto de algún fenómeno del mundo teniendo identificados los conceptos relevantes de dicho fenómeno: formal, porque la ontología debería ser legible por máquinas y compartida, puesto que no se recoge el conocimiento privado de un individuo sino el consensuado por un grupo de individuos.
- Chandrasekaran y colaboradores (Chandrasekaran et al., 1999), las definen como: *Ontologies are content theories about the sorts of objects, properties of objects, and relations between objects that are possible in a specified domain of knowledge*.
- Otra definición del mismo estilo es la dada por Grüninger y Fox (Grüninger et al., 2000): *An ontology is a formal description of entities and their properties, relationships, constraints, behaviours*.
- McGuinness y Noy (Noy and McGuinness, 2001) ofrecen una definición mucho más específica y concreta: *An ontology is a formal explicit description of concepts in a domain of discourse (classes -sometimes called concepts-), properties of each concept describing various features and attributes of the concept (slots -sometimes called roles or properties-), and restrictions on slots (facets -sometimes called role restriction-)*.

Desde el punto de vista de gestión del conocimiento, existen diferentes clasificaciones para las ontologías. A continuación se presentan algunas de las más representativas:

Según el grado de generalidad

Aunque muchas de las definiciones existentes afirman que los conceptos descritos en una ontología pertenecen a un dominio específico, Guarino (Guarino, 1998) propone la siguiente clasificación dependiendo de su grado de generalidad (ver figura 2.2., las flechas representan especialización):

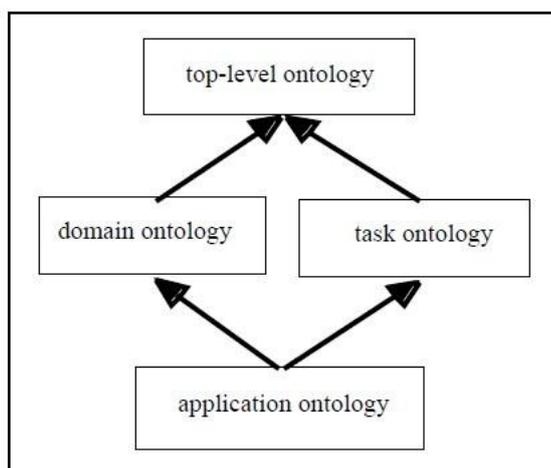


Figura 2.2. Clasificación de ontologías según su grado de generalidad. Fuente: (Guarino, 1998)

- Top-level ontologies (ontologías de alto nivel): describen conceptos genéricos como espacio, tiempo, materia, objeto, evento, acción, etc., los cuales son independientes del dominio o problema particular. Afirma que parece razonable, al menos en teoría, tener ontologías de alto nivel unificadas para grandes comunidades de usuarios.

- Domain ontologies and task ontologies (ontologías de dominio y de tarea): describen un dominio genérico o una tarea genérica especializando los términos introducidos en la ontología de alto nivel.
- Application ontologies (ontologías de aplicación): describen los conceptos dependiendo de un dominio o tarea particular para una aplicación específica.

Según su nivel de formalidad

Se ha comentado anteriormente, en las definiciones de ontología, que la mayoría afirman que debe ser formal, pero Gruber no descarta las ontologías no formales (Gruber, 1995) proponiendo otra clasificación:

- Una ontología informal podría ser especificada mediante un catálogo de tipos que o bien no están definidos o se usa sólo lenguaje natural en su definición.
- Una ontología formal se representa mediante una colección de nombres para conceptos y tipos de relaciones organizadas en un orden parcial por la relación tipo-subtipo. Estas ontologías se pueden clasificar también según la manera en que los subtipos se distinguen de los supertipos:
 - ✓ Ontología axiomatizada: distingue subtipos mediante axiomas y definiciones en lenguaje formal, como la lógica o alguna notación orientada a computadora tal que pueda ser traducida a lógica.
 - ✓ Ontología basada en prototipos: distingue subtipos mediante una comparación con un miembro típico o prototipo para cada subtipo.

Las ontologías grandes, a menudo, emplean una mezcla de métodos definicionales: axiomas y definiciones formales se usan por ejemplo para términos en matemáticas, físicas e ingenierías, y prototipos se usan para plantas, animales e ítems domésticos comunes.

Atendiendo a la conceptualización

Por otro lado, Van Heist et al (van Heijst et al., 1997), proponen una clasificación según dos dimensiones:

- Según cantidad y tipo de la estructura de la conceptualización:
 - ✓ Ontologías terminológicas, como los lexicones, especifican los términos que se usan para representar conocimiento en un dominio de discurso.
 - ✓ Ontologías de información, que representan la estructura de registros de las bases de datos.
 - ✓ Ontologías de modelado de conocimiento, que especifican conceptualizaciones del conocimiento. Suelen ser más ricas estructuralmente que las de información y, a menudo, se desarrollan conforme el uso particular del conocimiento que están describiendo.
- Según el sujeto de la conceptualización:
 - ✓ Ontologías de aplicación: contienen todas las definiciones que se necesitan para modelar el conocimiento requerido en una aplicación particular.
 - ✓ Ontologías de dominio: expresan conceptualizaciones que son dirigidas a dominios particulares.

- ✓ Ontologías genéricas: similares a las de dominio, pero los conceptos que definen se consideran genéricos en varios campos.
- ✓ Ontologías de representación: proporcionan un marco de trabajo representacional para definir otras ontologías.

Según el nivel de detalle en su especificación

Lassila y McGuinness proponen una clasificación de las ontologías en función de la expresividad y riqueza de su estructura interna (Lassila and McGuinness, 2001). Atendiendo a esta clasificación, los distintos tipos de ontologías pueden visualizarse dispuestos en un espectro o rango lineal como el representado en la figura 2.3.

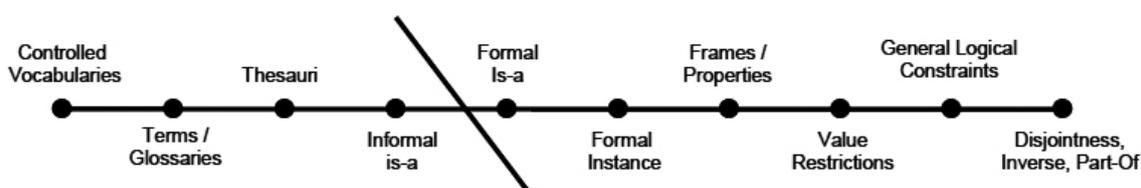


Figura 2.3. Espectro de ontologías. Fuente: (Lassila and McGuinness, 2001)

Por lo que, las ontologías, se clasificarían según definan:

- **Vocabularios controlados.** Mediante una lista finita de términos, como puede ser un catálogo.
- **Glosarios.** Mediante una lista de términos junto con su significado. El significado de los términos se especifica normalmente mediante sentencias en lenguaje natural.
- **Tesauros.** Proveen información adicional acerca de las relaciones entre términos, como las de sinonimia. Se introduce la desambiguación.
- **Taxonomías informales.** Esta categoría se introdujo para hacer referencia a ciertas ontologías que se pueden encontrar en la Web, y en las que las relaciones es-un (*is-a*) no son definidas en sentido estricto.
- **Taxonomías formales.** En este caso, si A es una superclase de B, entonces cualquier instancia de B es necesariamente una instancia de A. Se utilizan para explotar las propiedades de herencia.
- **Instancias formales.** Se incluyen también relaciones entre conceptos para describir un dominio concreto. Diferencian entre los esquemas anteriores que sólo incluyen nombres de clase y los que contienen identidad formal del contenido.
- **Marcos (frames).** Se añaden definiciones formales de propiedades sobre instancias. Estas propiedades participan en la herencia y en otras relaciones semánticas.
- **Restricciones de valor.** Permite restricciones sobre qué valores pueden tomar las propiedades.
- **Restricciones lógicas con carácter general.** Amplían la categoría anterior permitiendo a los valores de una propiedad estar basados en ecuaciones matemáticas usando valores de otras propiedades. Esto se consigue mediante la especificación de sentencias lógicas para determinar el valor de una propiedad.
- **Restricciones lógicas avanzadas.** Amplía la categoría anterior permitiendo que el valor de una propiedad pueda estar basado en las restricciones de la lógica de primer orden.

Conforme se avanza en la línea del espectro, se va avanzando a su vez en expresividad y poder semántico.

2.2.2 Componentes de una Ontología

Según Gómez-Pérez y colaboradores (Gómez-Pérez et al., 2007) una ontología está formada por los siguientes componentes (similares a los previamente introducidos en (Gruber., 1993) y (Noy and McGuinness, 2001)):

- **Clases** (o conceptos): son las ideas básicas que se intentan formalizar. Están organizadas en taxonomías a las que se les puede aplicar mecanismos de herencia¹³.
- **Relaciones**: explicando cómo los conceptos se relacionan entre sí. Existen relaciones tanto de especificación-generalización como de otro tipo.

$$R : C_1 \times C_2 \times \dots \times C_{n-1} \times C_n.$$

Por ejemplo: `SubClase : Ax B` o `CompuestoPor : F1 × F2`. Normalmente las relaciones en una ontología son binarias.

- **Funciones**: son casos especiales de relaciones donde se identifican elementos mediante el cálculo de una función.

$$F : C_1 \times C_2 \times \dots \times C_{n-1} \rightarrow C_n.$$

Por ejemplo:

`MadreDe : Persona → Mujer,`

`PrecioCocheUsado : ModeloxAnioxKilometros → Precio.`

- **Axiomas**: son teoremas que se declaran sobre relaciones que deben cumplir los elementos de la ontología. Por ejemplo: “Si A y B son de la clase C, entonces A no es subclase de B”. Los axiomas formales sirven para modelar sentencias que son siempre ciertas. Normalmente se usan para representar conocimiento que no puede ser formalmente definido por los componentes descritos anteriormente. Además, también se usan para verificar la consistencia de la propia ontología.
- **Instancias**: representan los objetos determinados de un concepto. Son los elementos o individuos de la ontología. Por ejemplo: `Manzana`. Acompañando a esos componentes pueden aparecer otros:
- **Constantes**: permiten representar valores primitivos como cadenas de caracteres o valores numéricos. Por ejemplo: “verde” ó 3,1416.
- **Atributos** (o propiedades): características o propiedades de las clases u objetos. Por ejemplo: `DNI` o `Apellidos`.
- **Restricciones**: descripciones sobre qué debe ser cierto para que un axioma sea cierto. Por ejemplo: para que un objeto sea un coche, su número de ruedas ha de ser “4”.

2.3. Lenguajes para la Definición de Tesauros y Ontologías

Con la aparición de la Web Semántica se crearon distintos lenguajes: RDF, RDFS y DAML+OIL (W3C, 2001) (el cual derivó finalmente en OWL). Más adelante, en 2003, surgió SKOS (Simple Knowledge Organization System) (Miles et al., 2005) empleado para representar la estructura básica y contenido de vocabularios controlados, el cual es el estándar más utilizado

¹³ A veces el término de ontología es confundido con el de taxonomía. La taxonomía se refiere a la clasificación jerárquica de los conceptos que forman la ontología.

para la definición de tesauros, mientras que OWL se ha convertido en el más utilizado en el ámbito de las ontologías.

Como OWL y SKOS están contruidos sobre RDF y RDF-Schema, se presenta previamente una pequeña introducción a los mismos.

2.3.1 RDF y RDF-Schema

RDF (Resource Description Framework) es un modelo de datos simple, recomendación del W3C, para intercambiar datos en la Web. Está destinado especialmente a la representación de metadatos sobre recursos Web, como el título, autor y fecha de modificación de una página Web, los derechos de autor e información de licencia de un documento Web, o el calendario de disponibilidad de algún recurso compartido. Sin embargo, mediante la generalización del concepto de un “recurso de la Web”, RDF también se puede utilizar para representar la información acerca de las cosas que se pueden identificar en la web.

RDF se basó en la idea de expresar declaraciones simples sobre recursos, donde cada declaración consiste en un sujeto, un predicado y un objeto. Supongamos que la sentencia:

“<http://www.example.org/index.html> has a creator whose value is [John Smith](http://www.example.org/staff_id/85740)”,

puede ser representada por una sentencia RDF donde en lugar de las palabras en lenguaje natural se utilizan IRIs, que además de identificar unívocamente a los elementos web, pueden ser interpretados por las máquinas:

- sujeto: <http://www.example.org/index.html>
- predicado: <http://purl.org/dc/elements/1.1/creator>
- objeto: http://www.example.org/staff_id/85740

Se puede representar esta sentencia mediante un grafo RDF como se muestra en la figura 2.4.

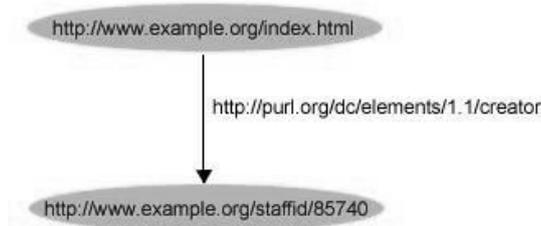


Figura 2.4. Sentencia expresada en forma de grafo RDF. Fuente: W3C.

Otra forma alternativa de escribir las declaraciones es mediante los llamados triples o tripletas. En la notación de triples, cada declaración en el gráfico se escribe como un triple simple de sujeto, predicado y objeto, en ese orden:

```
<http://www.example.org/index.html><http://purl.org/dc/elements/1.1/creator><http://www.example.org/staff\_id/85740>
```

Si se afirma además sobre el mismo sitio web que:

“<http://www.example.org/index.html> has a creation-date whose value is August16, 1999”

“<http://www.example.org/index.html> has a language whose value is English”

Se obtiene el grafo RDF de la figura 2.5. (usando IRIs apropiadas para nombrar las propiedades `creation-date` y `language`, distinguiendo los objetos literales de los objetos recursos que se identifican con nodos por lo que también se identifican con IRI) y los siguientes triples RDF:

```
<http://www.example.org/index.html><http://purl.org/dc/elements/1,1/creator>
<http://www.example.org/staff_id/85740>
<http://www.example.org/index.html><http://www.example.org/terms/creation-date>"August16, 1999".
<http://www.example.org/index.html><http://purl.org/dc/elements/1,1/language>"en".
```

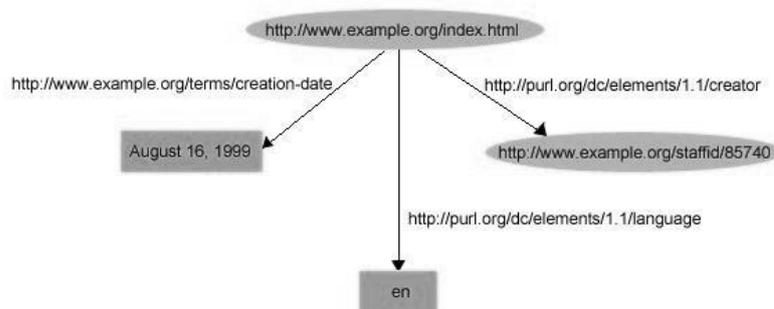


Figura 2.5. Grafo RDF. Fuente: W3C.

Para abreviar las notaciones de los espacios de nombres de las URIs se utilizan prefijos que, además, ayudan a discernir con mayor claridad el tipo de recurso. Por ejemplo, para describir el recurso web `<http://www.example.org/index.html>` que identifica a la organización “example” se usa el prefijo `ex`: y para el conjunto de triples anterior, los prefijos:

- `dc`: IRI del espacio de nombres Dublin Core³
- `http://purl.org/dc/elements/1,1/exterms`: IRI del espacio de nombres `http://www.example.org/terms/` (para los términos usados por la organización “example”),
- `exstaff`: IRI del espacio de nombres `http://www.example.org/staff_id:` (para los identificadores de los trabajadores de la organización “example”).

Los triples RDF anteriores podrían escribirse como:

- `ex:index.html` `dc:creator` `exstaff` : 85740
- `ex:index.html` `exterms:creation-date` “August16, 1999”
- `ex:index.html` `dc:language` “en”

Además de la notación en grafo y la notación en triples, RDF se puede expresar mediante la notación RDF/XML. Supongamos que queremos expresar el ejemplo anterior con esta notación:

```
<?xml version =“1.0”?>
<rdf:RDF xmlns:rdf =“http://www.w3.org /1999/02/22-rdf-syntax-ns#”
```

```

xmlns:dc="http://purl.org/dc/elements/1.1/"
xmlns:exterms="http://www.example.org/terms/">
<rdf:Description rdf:about="http://www.example.org/index.html">
  <exterms:creation-date>August 16,1999</exterms:creation-date>
</rdf:Description>
<rdf:Description rdf:about="http://www.example.org/index.html">
  <dc:language>en</dc:language>
</rdf:Description>
<rdf:Description rdf:about="http://www.example.org/index.html">
  <dc:creator rdf:resource="http://www.example.org/staffid/85740"/>
</rdf:Description>
</rdf:RDF>

```

Y de forma abreviada se podría expresar como:

```

<?xml version ="1.0"?>
<rdf:RDF xmlns:rdf="http://www.w3.org /1999/02/22-rdf-syntax-ns#"
  xmlns:dc="http://purl.org/dc/elements/1.1/"
  xmlns:exterms="http://www.example.org/terms/">
<rdf:Description rdf:about="http://www.example.org/index.html">
  <exterms:creation-date
    rdf:datatype="http://www.w3.org/2001/XMLSchema#date">1999-08-
    16</exterms:creation-date>
  <dc:language>en</dc:language>
  <dc:creator rdf:resource="http://www.example.org/staffid/85740"/>
</rdf:Description>
</rdf:RDF>

```

En este ejemplo, se ha modificado la forma en que se muestra la fecha de creación usando el tipo de datos `date` recogido en el XML Schema del W3C. Ahora, la fecha es un `Typed Literal` (literal tipificado). Para mejorar la legibilidad (y evitar el uso de `rdf:datatype`) se pueden usar entidades XML, `ENTITY`, que proporcionan una nueva abreviatura.

Resumiendo, RDF proporciona una manera de expresar declaraciones simples sobre recursos, utilizando las propiedades con nombre y los valores y utilizando sus conceptos básicos: recursos, propiedades, sentencias (sujeto-predicado-objeto), sentencias sobre sentencias, literales y contenedores (`bag`, `sequence` y `alternative` para describir distintos tipos de conjuntos). Sin embargo, las comunidades de usuarios de RDF también necesitan la capacidad de definir los vocabularios (términos) que se proponen utilizar en dichas sentencias, en particular, para indicar que se está describiendo tipos específicos o clases de recursos, y utilizar las propiedades específicas en la descripción de esos recursos. Por ejemplo, la compañía *example.com* querría describir clases como `exterms:Tent` y usar propiedades como `exterms:model`, `exterms:weightInKg` y `exterms:packedSize` para representarlas. RDF por sí mismo no proporciona medios para definir esas clases y propiedades. Hace falta un vocabulario que las describa tal que RDF pueda usar extensiones proporcionadas por dicho vocabulario, esto se podrá hacer gracias a RDF Schema (RDFS).

RDF Schema no proporciona un vocabulario para clases específicas de la aplicación como `exterms:Tent` ni para propiedades como `exterms:weightInKg`, sino que aporta las facilidades necesarias para describir tales clases y propiedades e indica qué clases y propiedades deben usarse juntas (por ejemplo, decir que la propiedad `exterms:weightInKg` debe usarse para describir a la clase `exterms:Tent`). En otras palabras, RDF Schema proporciona un sistema de tipos para RDF que es similar en algunos aspectos a los de los lenguajes de programación orientados a objetos, como Java. Por ejemplo, RDF Schema permite que los recursos sean definidos como instancias de una o más clases, y que las clases se organicen en modo jerárquico. Sin embargo, las clases y propiedades RDF también son muy distintas en algunos aspectos de los tipos de los lenguajes de programación.

Los conceptos básicos de RDFS son:

- Clase (`rdfs:Class`)
- Tipo (`rdfs:type`). Toda clase tiene un tipo que es `rdfs:Class`. Por ejemplo:


```
ex:MotorVehicle rdf:type rdfs:Class.
```
- Todas las propiedades en RDF se describen como instancias de la clase `rdf:Property`. Así que una nueva propiedad, como `exterms:weightInKg`, se escribe mediante la declaración RDF:


```
exterms:weightInKg rdf:type rdf:Property.
```
- Dominio, rango y subPropiedad (`rdfs:domain`, `rdfs:range`, `rdfs:subPropertyOf`). En RDFS, las propiedades se describen con la clase RDF `rdf:Property` y las propiedades de RDFS `rdfs:domain`, `rdfs:range`, y `rdfs:subPropertyOf`. La propiedad `rdfs:range` se usa para indicar que los valores de una propiedad particular son instancias de una clase designada. Por ejemplo, la organización “*example.org*” quiere indicar que la propiedad `ex:author` tiene instancias de `ex:Person`:


```
ex:Person          rdf:type          rdfs:Class.
ex:author          rdf:type          rdf:Property.
ex:author          rdfs:range        ex:Person.
```
- indicando que las declaraciones RDF que usen la propiedad `ex:author` tendrán instancias de `ex:Book` como objeto. La propiedad `rdfs:domain` se usa para indicar que una propiedad particular se aplica a clases específicas. Por ejemplo, si “*example.org*” quisiera indicar que la propiedad `ex:author` se aplica a las instancias de la clase `ex:Book`:


```
ex:Book           rdf:type          rdfs:Class.
ex:author         rdf:type          rdf:Property.
ex:author         rdfs:domain       ex:Book.
```
- indicando que las declaraciones RDF que usen la propiedad `ex:author` tendrán instancias de `ex:Book` como sujeto.
- SubClase (`rdfs:subClassOf`): utilizado para indicar que una clase es subclase de otra. Un ejemplo en la notación de un documento RDFS:


```
<rdfs:Class rdf:ID="Van">
  <rdfs:subClassOf rdf:resource="#MotorVehicle"/>
```

```
</rdfs:Class>
```

Además de los conceptos principales mostrados, existen otros conceptos en RDFS como `label`, `comment`, `seeAlso`, `isDefinedBy`, listas o contenedores,... etc.

2.3.2 Limitaciones del poder expresivo de RDF y RDF Schema

RDF y RDFS permiten la representación de algún conocimiento ontológico. Las primitivas de modelado principales conciernen a la organización de vocabularios en jerarquías con tipos: relaciones de subclase y subpropiedad, restricciones de rango y dominio e instancias de clases (Antoniou and Harmelen, 2003).

Por ejemplo, en RDF es posible (Horrocks et al., 2003):

- declarar clases como `Pais`, `Persona`, `Estudiante` y `Canadiense`
- afirmar que `Estudiante` es subclase de `Persona`
- afirmar que `Canada` es instancia de `Pais`
- declarar `Nacionalidad` como una propiedad relacionada con la clase `Persona` (su dominio) y `Pais` (su rango)
- afirmar que `edad` es una propiedad con `Persona` como dominio y `integer` como su rango
- afirmar que `Peter` es una instancia de la clase `Canadiense`, y que su `edad` tiene valor 48.

Sin embargo, en RDF/RDFS no es posible declarar (Antoniou and Harmelen, 2003) entre otras cosas:

- **Ámbito local de las propiedades:** `rdfs:range`. Por ejemplo no es posible afirmar que las vacas comen sólo plantas, mientras que otros animales podrían comer carne también.
- **Clases disjuntas:** por ejemplo que `Femenino` y `Masculino` son clases disjuntas.
- **Combinaciones booleanas de clases:** no es posible construir clases nuevas combinando otras mediante unión, intersección o complemento. Por ejemplo, en RDFS no se puede definir la clase `Persona` como la unión disjunta de las clases `Femenino` y `Masculino`.
- **Restricciones de cardinalidad:** por ejemplo que una persona tiene exactamente dos padres, que un curso debe tener al menos un profesor.
- **Algunas características especiales:** a veces es útil indicar que una propiedad es transitiva (como “mayor que”), única (como “es madre de”) o la inversa de otra propiedad (como “come” y “es comido por”).

Los principales requerimientos de los lenguajes de ontologías son (Antoniou and Harmelen, 2003): (a) una sintaxis bien definida, (b) una semántica bien definida, (c) soporte para el razonamiento eficiente, (d) suficiente poder expresivo, y (e) conveniencia de expresión.

Por lo tanto, se necesita un lenguaje de ontologías que sea más rico que RDFS y que permita suficiente poder expresivo para favorecer el razonamiento. Con este propósito, el W3C desarrolla OWL (Web Ontology Language), como recomendación para el desarrollo de ontologías en la Web Semántica.

2.3.3. OWL

OWL se ha convertido en uno de los lenguajes más populares en el campo de las ontologías, incluso, cada vez se utiliza más fuera del ámbito de la Web Semántica (McGuinness and Harmelen, 2004). En el apartado 2.3.2. se han expuesto las motivaciones para el desarrollo de este lenguaje.

OWL proporciona tres sublenguajes (también llamados *especies*) que son incrementalmente expresivos y diseñados para comunidades específicas de implementadores y usuarios:

- OWL Lite: da soporte a aquellos usuarios cuyas necesidades principales son la jerarquía de clasificación y restricciones simples. Por ejemplo, sólo permite valores 0 - 1 para cardinalidad.
- OWL DL: da soporte a aquellos usuarios que quieren la máxima expresividad, pero cumpliendo la completitud computacional (se garantiza que todas las conclusiones son computables) y decibilidad (todos los cálculos terminarán en tiempo finito). Incluye todas las construcciones del lenguaje OWL, aunque pueden usarse sólo bajo ciertas restricciones (por ejemplo, mientras que una clase puede ser subclase de varias clases, una clase no puede ser instancia de otra). El nombre OWL DL se debe a su correspondencia con la Lógica Descriptiva (ver Apartado 2.6 sobre Lógica Descriptiva y sus equivalencias con OWL DL).
- OWL Full: está diseñado para usuarios que quieren la máxima expresividad y libertad sintáctica de RDF pero sin garantías computacionales. Es improbable que un software de razonamiento sea capaz de soportar un razonamiento completo en OWL Full.

Hay que recordar que todo lo que se puede expresar/concluir con OWL Lite se puede expresar / concluir también con OWL DL y OWL Full, sin embargo, al revés no se cumple. En la figura 2.6. puede observarse el vocabulario principal de los tres sublenguajes (W3C, 2004).

Tanto OWL DL como OWL Full utilizan el mismo vocabulario, aunque OWL DL está sujeto a algunas restricciones para permitir la computabilidad. A grandes rasgos, OWL DL requiere una separación de tipos (una clase no puede ser también individuo o propiedad, una propiedad no puede ser ni individuo ni clase), lo cual está permitido en OWL Full. Por otra parte, OWL DL requiere que las propiedades sean *ObjectProperties* (relaciones entre instancias de dos clases) o *DatatypeProperties* (relaciones entre las instancias de clases y literales de RDF y tipos de datos de XML Schema).

OWL DL se basa, en particular, en el sistema *SHIQ* de la Lógica Descriptiva, gracias a lo cual se consigue, puedan aplicar algoritmos de razonamiento conocidos, por lo que permite llevar cabo procesos de razonamiento automáticos. Razonar sobre una representación del modelo de dominio puede ayudar a guiar el proceso de modelado, así como a descubrir inconsistencias o conocimiento no declarado explícitamente gracias a la inferencia.

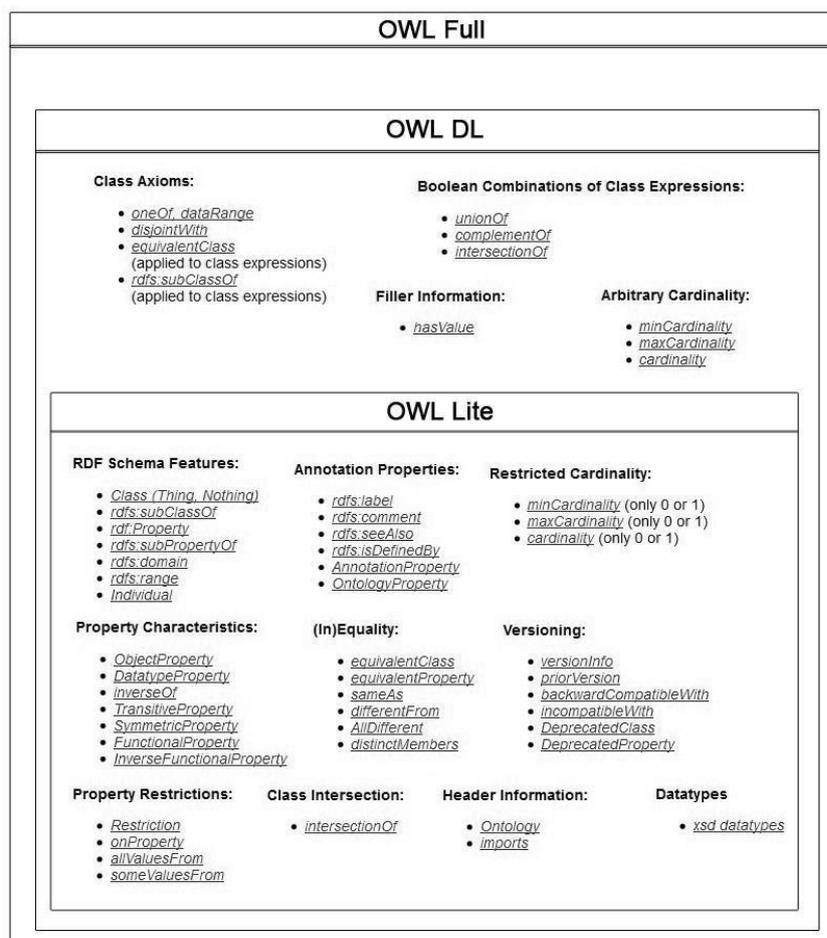


Figura 2.6. Vocabulario OWL Lite, OWL DL, OWL Full. Fuente: (W3C, 2004).

Todos estos motivos convierten a OWL DL en el sub-lenguaje preferido por los investigadores y desarrolladores para el modelado conceptual de un dominio mediante ontologías. Por tanto, parece que OWL DL es la opción más adecuada para obtener un modelo conceptual de alto nivel acerca de un dominio y, a la vez, permitir inferir nuevo conocimiento a partir del declarado explícitamente en una base de conocimiento (Baader et al., 2003).

Ya se ha observado en la pila de estándares de la web semántica (figura 2.1.) que OWL está construido sobre RDF y RDFS por lo que es considerado una extensión. Todas las variedades de OWL usan RDF para su sintaxis (aunque se añaden algunas construcciones sintácticas). Las instancias se declaran como en RDF, usando descripciones RDF e información de tipos. Los constructores OWL como `owl:Class`, `owl:DatatypeProperty` y `owl:ObjectProperty` son especializaciones de sus homólogos RDF.

En la figura 2.7. se muestran las relaciones de subclase entre algunas primitivas de modelado OWL y RDF/RDFS (Antoniou and Harmelen, 2003).

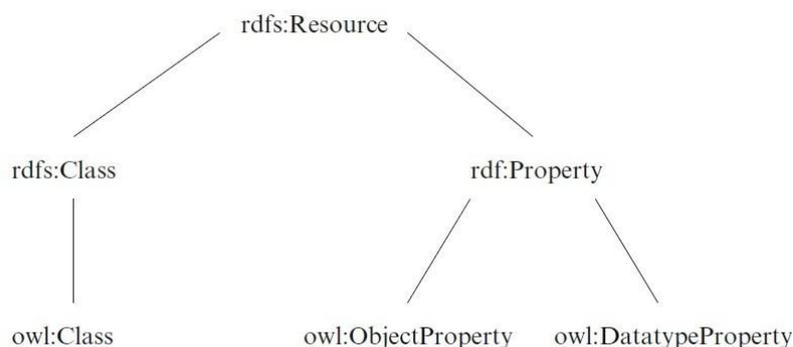


Figura 2.7. Relaciones de subclase entre OWL y RDF/RDFS. Fuente: (Antoniou and Harmelen, 2003)

Clases OWL

Las clases se definen mediante `owl:Class`, por ejemplo, es posible definir la clase `profesorAsociado` como:

```

<owl:Class rdf:ID="profesorAsociado">
  <rdfs:subClassOf rdf:resource="#miembroPersonalAcademico" />
</owl:Class>
  
```

Se puede afirmar que es disjunta de las clases `profesor` y `profesorAsistente`:

```

<owl:Class rdf:about="profesorAsociado">
  <owl:disjointWith rdf:resource="#profesor"/>
  <owl:disjointWith rdf:resource="#profesorAsistente"/>
</owl:Class>
  
```

También es posible describir equivalencias de clases:

```

<owl:Class rdf:ID="profesorAsociado">
  <owl:equivalentClass
    rdf:resource="#miembroPersonalAcademico"/>
</owl:Class>
  
```

Existen también dos clases predefinidas `owl:Thing`, que es la clase más general que contiene a todo (todo es una cosa) y `owl:Nothing` que es la clase vacía.

Las instancias de las clases se declaran como en RDF:

```

<rdf:Description rdf:ID="873410">
  <rdf:type rdf:resource="#miembroPersonalAcademico"/>
</rdf:Description>
  
```

O de forma equivalente abreviada como:

```

<miembroPersonalAcademico rdf:ID="873410"/>
  
```

Al contrario que los sistemas de base de datos típicos, OWL no adopta la asunción del mundo cerrado: todo lo que no es conocido es falso. Tampoco adopta la asunción de nombres únicos, es decir, el hecho de que dos instancias tengan distinto nombre (o ID), no implica que sean

individuos diferentes. Por ejemplo, afirmando que cada curso es impartido por “como máximo uno” miembro del personal:

```
<owl:ObjectProperty rdf:ID="impartidoPor">
  <rdf:type rdf:resource="&owl;FunctionalProperty"/>
</owl:ObjectProperty>
```

la siguiente declaración NO llevará a error en el razonamiento, sino que seguramente inferirá que se trata de la misma persona, o sea que los recursos “873410” y “873412” son aparentemente iguales:

```
<curso rdf:about="Science12-1">
  <impartidoPor rdf:resource="873410"/>
  <impartidoPor rdf:resource="873412"/>
</curso>
```

Para asegurar que individuos diferentes se reconocen como tal, hay que afirmarlo explícitamente:

```
<profesor rdf:ID="873410"/>
  <owl:differentFrom rdf:resource="873412"/>
<profesor>
```

Como esto ocurre muy a menudo, OWL proporciona una notación abreviada para afirmar que una lista de individuos son diferentes:

```
<owl:AllDifferent>
  <owl:distinctMembers rdf:parseType="Collection">
    <profesor rdf:about="873410"/>
    <profesor rdf:about="873411"/>
    <profesor rdf:about="873412"/>
  <owl:distinctMembers>
</owl:AllDifferent>
```

Algunas limitaciones del lenguaje OWL

Aunque, o quizás porque, OWL 1.1, ha tenido éxito, se han identificado varios problemas en su diseño ((Antoniou and Harmelen, 2003), (Horrocks et al., 2003), (Grau et al., 2008)), lo cual indicaba una necesidad de revisión de OWL1.

En (Grau et al., 2008) se citan los principales problemas de la primera versión de OWL, (que motivan la nueva versión). Algunos de ellos son los siguientes:

1. **Limitaciones de expresividad.** Para paliar algunos de los problemas de expresividad que la falta de ciertos constructores en OWL1 trae consigo, la comunidad OWL ha desarrollado una serie de patrones. Pero en muchos casos, esto tampoco es capaz de proporcionar la semántica deseada. Además, es generalmente difícil, si no imposible, identificar todos los

¹⁴ A partir de ahora llamaremos OWL1 a lo que antes referíamos como OWL para poder relizar las comparativas.

casos en que esos patrones llevarían a resultados incorrectos. Por lo que parece ser que la única solución es extender el conjunto de constructores.

- Restricciones de cardinalidad cualificadas (QCR). Restringir el número de valores de un tipo particular en una propiedad. A menudo es útil expresar restricciones tales como “tiene exactamente cuatro partes que son las piernas”, “tiene al menos dos grupos que son grupos de fosfato”, “tiene exactamente una característica que es la temperatura”, etc. En cada uno de estos casos, lo que se desea limitar no es el número total de valores para una propiedad, sino el número de valores de un tipo dado. Esas restricciones se llaman restricciones de cardinalidad cualificadas (QCR), porque son “cualificadas” según el tipo del valor. Existen una serie de patrones de representación para lidiar con esto¹⁵.
 - Expresividad relacional. La falta de esta expresividad se ha considerado uno de los mayores problemas de OWL. Por ejemplo, expresividad sobre las propiedades. Los casos más importantes son: la propagación por propiedades y las propiedades de propiedades.
 - Expresividad de los tipos de datos. OWL proporciona un poder expresivo muy limitado para la descripción de clases cuyas instancias están relacionadas con valores concretos (como enteros o cadenas). En OWL1, es posible expresar las restricciones a las propiedades de tipo de datos cualificados por un tipo de datos unario. Por ejemplo, se podría afirmar que cada ciudadano británico debe tener un número de pasaporte, que es un `xsd:string`. En OWL1, sin embargo, no es posible representar los siguientes tipos de sentencias: “un vendaval es un viento cuya velocidad está en el intervalo de 34 a 40 nudos” (restricciones a un subconjunto de valores de tipo de dato), “una mesa cuadrada es una mesa cuyo ancho es igual a su largo” (relaciones entre valores y propiedades de datos en un objeto), “gente que es mayor que sus jefes” (relaciones entre los valores de las propiedades de datos sobre objetos diferentes) o “la duración de un proceso es la suma de las duraciones de sus subprocesos” (funciones de agregación).
 - Llaves. La falta de restricciones de llave única (como en base de datos) se ha reconocido como una importante limitación en el poder expresivo de OWL1. El hecho de añadirlas directamente dañaría las propiedades computacionales del lenguaje e incluso podría llevar a indecidibilidad.
2. **Problemas de sintaxis.** OWL1 presenta dos sintaxis normativas: la Sintaxis abstracta¹⁶ y OWL1 RDF¹⁷. La sintaxis abstracta es confusa para los desarrolladores, lo cual resulta muchas veces en diseños subóptimos. Además, se ha probado que ambas conllevan dificultades a la hora de ser parseadas. La relación compleja entre las dos sintaxis provoca también dificultades a la hora de tener que transformar la una en la otra.
 3. **Metamodelado.** En muchos casos prácticos, la distinción entre las clases y los individuos no es clara. Por ejemplo, las declaraciones de que “Harry es un águila” y “Águilas son una especie en peligro de extinción”. La primera declaración puede ser modelada en OWL1 al afirmar que el individuo Harry es una instancia de la clase de *Aguila*. La segunda afirmación, sin embargo, habla de las águilas, como especie, y no como un conjunto de todos los seres vivos águilas; por lo tanto, debe modelarse afirmando que el individuo

¹⁵ <http://www.w3.org/2001/sw/BestPractices/OEP/QCR/>

¹⁶ Sintaxis abstracta <http://www.w3.org/TR/owl-semantics/>

¹⁷ Sintaxis OWL/RDF <http://www.w3.org/TR/owl-ref/>

Aguila es una instancia de la clase `EspeciesEnPeligroDeExtincion`. Por lo tanto, Aguila juega el papel de una clase en un contexto y de un individuo en otro. Este estilo de modelado, a menudo se llama metamodelado. De hecho, en ocasiones, los usuarios se ven obligados a elegir entre metamodelado, que puede ser necesario en su aplicación, y una herramienta de apoyo para el desarrollo de la ontología.

4. **Importación y versiones.** OWL1 proporciona un mecanismo básico que permite a una ontología O importar otra ontología O' , ganando así acceso a todas las entidades y los axiomas en O' . Sin embargo, las dos ontologías están separadas físicamente. Las importaciones en OWL DL 1 son "por el nombre y la ubicación": la ontología O debe contener una URI que apunta a la ubicación de O' , y esta ubicación debe coincidir con el nombre de O' . Muchas veces, las ontologías (tanto en web como en local) cambian de sitio y la ubicación de una ontología finalmente no se corresponde con su nombre. En la mayoría de los casos esto debe solucionarse manualmente. Estos problemas se agravan cuando se deben mantener distintas versiones de una ontología.
5. **Anotaciones.** En OWL1 se permiten anotaciones para añadir información adicional, pero no es posible definir una subpropiedad o añadir restricciones de dominio o rango a una anotación. Esto es lógico en OWL DL porque en realidad no sirven para el razonamiento ya que no se pueden utilizar en los axiomas, por lo tanto, el tipo de semántica que añaden es no-razonable, lo cual es una contradicción con las bases de OWL DL. Sin embargo, ya en la comunidad de usuarios de OWL ha habido quejas por no poder utilizar axiomas en las anotaciones, por ejemplo: decir que el rango de la anotación `proteinID` es "String".
6. **Validación de especies.** Se refiere al problema de validar si una ontología OWL1 está escrita en OWL Lite, OWL DL o OWL Full, lo cual puede no ser tan obvio.

2.3.4 OWL2

Debido a las limitaciones del lenguaje OWL1 para construir ontologías, el W3C comenzó, en 2007, un nuevo grupo de trabajo para desarrollar un sucesor: OWL2. Este lenguaje extiende al anterior, añadiendo nuevas características para el razonamiento, de modo que se puedan mejorar la satisfacción de los usuarios según sus necesidades.

La sintaxis principal para OWL2 es RDF/XML¹⁸, que es de hecho, la única sintaxis que debe ser apoyada por todas las herramientas de OWL2. Pero esta no es la única sintaxis permitida. Se muestra en la figura 2.8 se muestra todas las sintaxis que pueden utilizarse y cual es su principal motivación y utilidad.

¹⁸ <http://www.w3.org/TR/rdf-syntax-grammar/>

SINTAXIS	ESTADO	Propósito
<i>RDF/XML</i>	Obligatorio	Intercambio (puede ser escrita y leída por todo el software conforme a OWL 2)
<i>OWL/XML</i>	Opcional	Facilita el procesamiento con herramientas XML
<i>Funcional</i>	Opcional	Es más fácil ver la estructura formal de las ontologías
<i>Manchester</i>	Opcional	Facilita leer y escribir ontologías DL
<i>Turtle</i>	Opcional	Facilita leer y escribir triples RDF

Figura 2.8. Posibles sintaxis de OWL2. Fuente: W3C.

Perfiles de OWL2

Los perfiles de OWL2 (OWL 2 *EL*, OWL 2 *QL* y OWL 2 *RL*) son sublenguajes (subconjuntos sintácticos) de OWL2 que ofrecen importantes ventajas en determinados escenarios. Son subconjuntos de elementos estructurales que pueden usarse según la ontología y cada uno es más restrictivo que OWL *DL*.

- OWL 2 *EL*: permite algoritmos de tiempo polinomiales para todas las tareas de razonamiento estándar. Es adecuado para aplicaciones donde se necesitan ontologías de gran tamaño, y donde el poder expresivo no es tan importante como las garantías de rendimiento.
- OWL 2 *QL*: permite consultas conjuntivas que deben ser respondidas utilizando la tecnología estándar de base de datos relacional; es particularmente adecuado para aplicaciones donde se utilizan ontologías relativamente ligeras para organizar un gran número de personas o en las que se suele acceder a los datos directamente a través de consultas relacionales (como SQL).
- OWL 2 *RL*: permite la implementación de algoritmos de tiempo de razonamiento polinomial que operan directamente en triples RDF. Es particularmente adecuado para aplicaciones donde se utilizan ontologías relativamente ligeras para organizar un gran número de personas o en las que se suele operar directamente con los datos en forma de triples RDF.

En la figura 2.9 se aprecian las diferencias entre OWL1 y sus sublenguajes, OWL2 y sus sublenguajes con perfiles y cómo RDFS se corresponde con ambos.

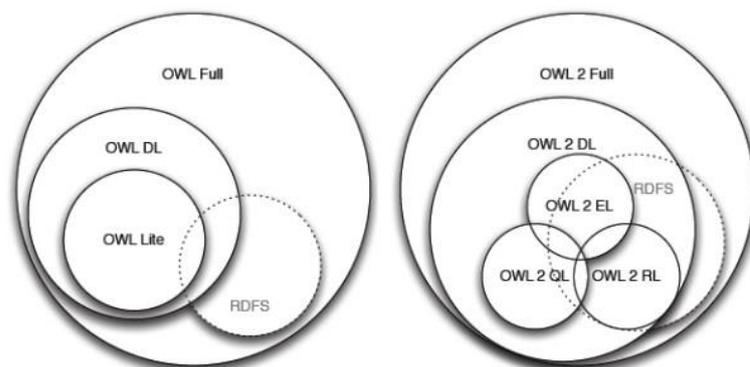


Figura 2.9. Sublenguajes OWL1 vs OWL2 . Fuente: (Hoekstra, 2009).

Diferencias entre OWL 1 y OWL 2

La nueva sintaxis funcional de OWL2, aunque difiere muy poco de la de OWL1, es más próxima a la representación de grafos RDF y puede capturar aun más grafos RDF; además tiene una equivalencia formal con UML (Unified Modeling Language)¹⁹. OWL2, se beneficia también de un mapeo especificado explícitamente desde los grafos RDF hacia las estructuras de la ontología.

OWL2 añade nueva funcionalidad respecto a OWL1. Algunas de estas nuevas características son “azúcar sintáctico” (como la unión disjunta de clases) mientras que otras ofrecen nueva expresividad, incluyendo:

- **llaves:** OWL 2 permite definir llaves al estilo de las llaves de base de datos mediante un nuevo constructor `HasKey`. Un axioma `HasKey` afirma que cada instancia con nombre de una clase se identifica de forma única mediante una propiedad o un conjunto de propiedades (objeto o de datos). Es decir, si dos instancias de una clase coinciden en todos los valores de las propiedades llave, entonces, esos dos individuos, son el mismo²⁰.
- **cadena de propiedades:** permiten la “herencia” transitiva de alguna propiedad a través de una cadena de propiedades (Horrocks et al., 2006).
- **tipos de datos más ricos, rangos de datos:** se mejoran y se extienden los tipos de datos, por ejemplo, OWL2 extiende los tipos de datos unarios de OWL1 a tipos de datos n-arios, permitiendo comparar valores de las propiedades para un objeto dado.
- **restricciones de cardinalidad cualificada QCR:** permite restringir la cardinalidad de una propiedad a un rango particular.
- **propiedades asimétricas, reflexivas y disjuntas:** las propiedades asimétricas son aquellas que no se cumplen en sentido contrario, o sea, no se cumple “viceversa” en la propiedad. Por ejemplo, afirmar que algo es más grande que algo. Dos propiedades son disjuntas cuando sus conjuntos de modelos son disjuntos; ningún par de individuos puede estar en ambos conjuntos. Las propiedades reflexivas son aquellas que se afirman sobre algo de sí mismo, por ejemplo, las personas narcisistas son aquellas que se gustan a sí mismas (Hoekstra, 2009).

¹⁹ Lenguaje de modelado definido por el consorcio OMG - Object Management Group. <http://www.uml.org/>

²⁰ Nuevas características de OWL2: <http://www.w3.org/TR/2008/WD-owl2-new-features-20081202/>

- **capacidades de anotación mejoradas:** se pueden añadir anotaciones a entidades, propiedades, ontologías e incluso a axiomas²¹.

Las ontologías en OWL2 y sus elementos se identifican mediante IRIs (Internationalized Resource Identifiers), mientras que OWL1 usaba URIs (Uniform Resource Identifiers). Las IRIs deben ser absolutas, por lo que dos IRIs son estructuralmente equivalentes si y solo si sus cadenas de representación son idénticas²².

2.3.5 SKOS

SKOS es un vocabulario RDF que sirve como modelo para expresar la estructura básica y el contenido de esquemas de conceptos (tesauros, esquemas de clasificación, listas de encabezamientos de materia, taxonomías, terminología, glosarios y otros tipos de vocabulario controlado). Está publicado y mantenido por el W3C.

El vocabulario SKOS Core está formado por las siguientes etiquetas (W3C, 2009):

- Conceptos y esquemas de conceptos. Un concepto SKOS puede ser visto como una idea o noción, una unidad de pensamiento y un esquema de conceptos como una agregación de uno o varios conceptos.
 - `skos:Concept`
 - `skos:ConceptScheme`
 - `skos:inScheme`
 - `skos:hasTopConcept`
 - `skos:topConceptOf`
- Etiquetas léxicas. Representan a cadenas de caracteres UNICODE. Se utilizan para expresar la preferencia de uso de términos.
 - `skos:prefLabel`
 - `skos:altLabel`
 - `skos:hiddenLabel`
- Relaciones semánticas. Se usan para representar enlaces inherentes al significado entre conceptos SKOS.
 - `skos:semanticRelation`
 - `skos:broaderTransitive`
 - `skos:narrowerTransitive`
 - `skos:broader`
 - `skos:narrower`
 - `skos:related`
- Documentación. También llamadas Notas. Proporcionan cualquier tipo de información relativa a los conceptos SKOS. Pueden ser texto, hipertexto o incluso imágenes.
 - `skos:note`
 - `skos:changeNote`

²¹ Anotaciones en OWL2 <http://www.w3.org/TR/owl2-syntax/#Annotations>

²² IRIs: <http://www.w3.org/TR/owl2-syntax/#IRIs>

- `skos:scopeNote`
- `skos:historyNote`
- `skos:definition`
- `skos:editorialNote`
- `skos:example`
- Colecciones de conceptos. Son grupos etiquetados u ordenados de conceptos. Son útiles cuando un grupo de conceptos comparte algo en común o cuando varios conceptos pueden colocarse en u orden significativo.
 - `skos:Collection`
 - `skos:OrderedCollection`
 - `skos:member`
 - `skos:memberList`
- Propiedades de mapeado. Se usan para hacer mappings entre conceptos de diferentes esquemas. Estos links son inherentes al significado.
 - `skos:mappingRelation`
 - `skos:exactMatch`
 - `skos:broadMatch`
 - `skos:narrowerMatch`
 - `skos:relatedMatch`
- Notaciones. Son cadenas de caracteres que identifican unívocamente a un concepto en un esquema determinado.
 - `skos:notation`

En el documento (W3C, 2009) se explica cada etiqueta y se introducen las posibles relaciones entre OWL y SKOS; por ejemplo, un concepto SKOS puede ser entendido como una instancia de una clase OWL.

2. 4. Proceso de Desarrollo de Ontologías

Noy y McGuinness (Noy and McGuinness, 2001) proponen una metodología de desarrollo de ontologías haciendo especial énfasis en las siguientes reglas como fundamentales en el diseño de ontologías, que ayudarán en muchos casos a la hora de tomar decisiones de diseño:

- No hay un modo correcto único de modelar un dominio, siempre hay varias alternativas. La mejor solución siempre dependerá de la aplicación que se quiere desarrollar y las extensiones que se puedan anticipar.
- El desarrollo de ontologías es necesariamente un proceso iterativo.
- Los conceptos de la ontología deberían ser similares a los objetos (físicos o lógicos) y a las relaciones en el dominio de trabajo. Esto se conseguirá mediante el uso de nombres (objetos) o verbos (relaciones) en sentencias que describan el dominio.

En el desarrollo de ontologías recomiendan los siguientes pasos:

PASO 1 - Determinar alcance y ámbito de la ontología. Esto es, contestar a las siguientes preguntas básicas:

- ¿Cuál es el dominio a cubrir por la ontología?
- ¿Para qué vamos a usarla?
- ¿Para qué tipo de cuestiones deberá proporcionar respuestas?
- ¿Quién usará y mantendrá ontología?

Las respuestas a estas cuestiones podrían cambiar durante el proceso de diseño, pero siempre van a ayudar a limitar el ámbito del modelo.

PASO 2 - Considerar la reutilización de ontologías existentes.

Suele merecer la pena considerar lo que otros han hecho y comprobar si es posible perfeccionar y ampliar las fuentes existentes para nuestro dominio y tarea particular. La reutilización de ontologías existentes puede ser un requisito si nuestro sistema tiene que interactuar con otras aplicaciones que ya se han comprometido a las ontologías particulares o vocabularios controlados. Muchas ontologías ya están disponibles en formato electrónico y se pueden importar en el entorno de desarrollo de la ontología que se esté utilizando.

PASO 3 - Enumerar los términos importantes de la ontología.

Es útil escribir una lista de todos los términos sobre los que nos gustaría hacer declaraciones o para explicar a un usuario. ¿Cuáles son los términos de los que nos gustaría hablar? ¿Qué propiedades tienen esos términos? ¿Qué nos gustaría que decir acerca de esos términos?. Inicialmente, es importante obtener una lista completa de los términos sin tener que preocuparnos por el solapamiento entre conceptos, relaciones entre ellos, o propiedades las que los conceptos pueden tener, o si los conceptos son clases o propiedades.

Los siguientes pasos son los más importantes en el proceso de diseño y están muy entrelazados. Es difícil hacer uno de ellos completamente primero y luego el otro. Normalmente, se crean algunas definiciones de los conceptos en la jerarquía y, a continuación, se procede a la descripción de sus propiedades, y así sucesivamente, de forma iterativa.

PASO 4 - Definir las clases y la jerarquía de clases.

Para realizar la taxonomía de clases es posible seguir varios enfoques: *top-down* (se empieza con los conceptos - clases - generales y se van creando subclases), *bottom-up* (partiendo de los conceptos - clases - más específicos y agrupándolos en superclases o conceptos más generales) o un híbrido entre ambos. Sea cual sea el enfoque elegido, por lo general, se comenzará por la definición de clases. De la lista creada en el Paso 3, se seleccionan los términos que describen los objetos que tienen existencia independiente (en lugar de los términos que se usan para describirlos). Estos términos serán las clases de la ontología. Se organizan las clases en una taxonomía jerárquica preguntando si por ser una instancia de una clase, el objeto necesariamente (es decir, por definición) es instancia de alguna otra clase: Si una clase A es una superclase de B, entonces cada instancia de B es también una instancia de A.

PASO 5 - Definición de las propiedades de las clases (slots).

Una vez definidas algunas de las clases, es necesario describir la estructura interna de los conceptos. Ya están seleccionadas las clases de la lista de términos creadas en el Paso 3. La mayoría de los términos restantes seguramente sean las propiedades de estas clases. Para cada propiedad en la lista, se determina qué clase describe. Estas propiedades se convierten en las propiedades unidas a las clases. Todas las subclases de una clase heredan las propiedades de esa

clase. La propiedad, por tanto, debe asociarse a la clase más general que puede tener esa propiedad.

PASO 6 - Definir las facetas o restricciones de los slots (facets).

Los slots pueden tener diferentes facetas que describan el tipo de valor, los valores permitidos, el número de los valores (cardinalidad), y otras características de los valores que la propiedad pueda tomar. Los más importantes son: cardinalidad, tipo de valor y dominio y rango del slot.

PASO 7 - Crear instancias.

El último paso es la creación de instancias individuales de las clases en la jerarquía. La definición de una instancia individual de una clase requiere (1) la elección de una clase, (2) la creación de una instancia individual de esa clase y (3) rellenar los valores de los slots. Es posible, entonces, crear una base de conocimiento mediante la definición de instancias individuales de las clases rellenando el valor de información específico del slot y sus restricciones adicionales.

2.5. Herramientas para la edición y manejo de ontologías

En este apartado se procede a comentar algunas de las herramientas de edición y manejo de ontologías más utilizadas.

2.5.1 Protégé

Protégé (PROTÉGÉ, 2013) es un editor de ontologías libre de código abierto y un marco de trabajo sobre bases de conocimiento. Está desarrollado por la Universidad de Stanford, en colaboración con la Universidad de Manchester.

La plataforma soporta dos formas principales de modelado de ontologías, los editores Protégé-Frames y Protégé-OWL.

- **Protégé-Frames:** este editor permite a los usuarios construir y poblar ontologías basadas en frames de acuerdo con el protocolo OKBC²³ (Open Knowledge Base Connectivity). En este modelo, una ontología consiste en un conjunto de clases organizadas en una ontología de subsunción (inclusión de conceptos dentro de otros), para representar los conceptos relevantes de un dominio, un conjunto de slots asociados a las clases para describir sus propiedades y relaciones y un conjunto de instancias de esas clases.
- **Protégé-OWL:** este editor permite a los usuarios construir ontologías para la Web Semántica, en particular en OWL. A partir de ellas se permite el razonamiento y derivar sus consecuencias lógicas, es decir, hechos que no aparecen directamente en la ontología explícitamente pero se deducen por la semántica, esto es posible gracias a la semántica formal de OWL. Estas implicaciones pueden estar basadas en un único documento o documentos múltiples distribuidos que se combinan usando diferentes mecanismos OWL.

Las ontologías Protégé pueden ser exportadas a una variedad de formatos, incluyendo RDF(S), OWL y XML Schema.

²³ OKBC <http://www.ai.sri.com/~okbc/>

Protégé está basado en Java, es extensible y personalizable, ya que proporciona un entorno de plug-and-play que le convierte una base flexible para la creación rápida de prototipos y desarrollo de aplicaciones. Tiene una extensión para soportar desarrollo colaborativo de ontologías (aunque actualmente sólo existe para la versión 3.x) y una versión web ligera del editor llamada WebProtégé²⁴ para soportar el proceso de desarrollo colaborativo.

2.5.2 Jena

Jena²⁵ (McBride, 2001) fue originalmente desarrollado por investigadores de los Laboratorios de HP, en Bristol, Reino Unido, en el año 2000. Jena ha sido siempre un proyecto de código abierto, ampliamente utilizado en una gran variedad de aplicaciones de web semántica y demostraciones. En 2009, HP decidió una reorientación de las actividades de desarrollo fuera del mantenimiento directo de Jena, aunque sigue siendo de apoyo de los objetivos del proyecto. Desde noviembre de 2010, Jena es propiedad de la Apache Software Foundation.

Apache Jena es un framework para desarrollar aplicaciones en Java con tecnologías de la Web Semántica. En la figura 2.10 se muestra la arquitectura de Jena.

El marco de trabajo Jena 2 incluye:

- Una API para leer, procesar y escribir datos en XML, N-Triples y Turtle²⁶.
- Una API de ontologías para manejar ontologías en OWL y RDFS.
- Un motor de inferencia basado en reglas para razonamiento con fuentes de datos RDF y OWL.
- Permite que se puedan almacenar en disco un gran número de triples RDF de forma eficiente.
- Un motor de consultas compatible con la última especificación de SPARQL.
- Servidores para permitir la publicación de datos RDF para que pueda ser usado por otras aplicaciones diferentes.

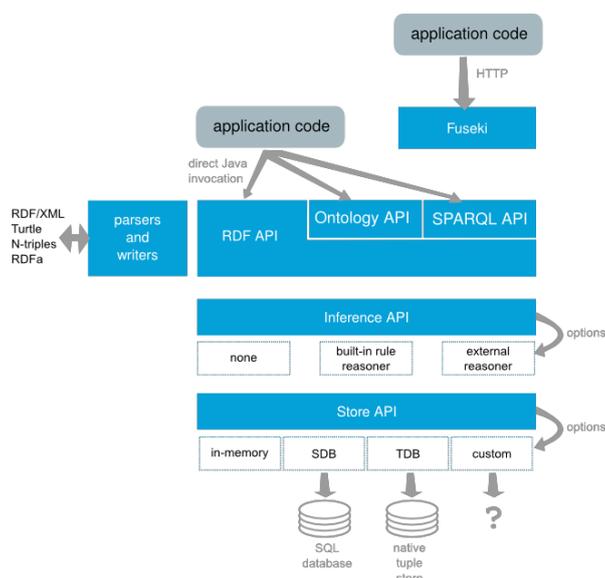


Figura 2.10. Arquitectura de Jena. Fuente: Apache Jena.

²⁴ <http://protegewiki.stanford.edu/wiki/WebProtege>

²⁵ <http://jena.apache.org/>

²⁶ Sintaxis Turtle de RDF: <http://www.w3.org/2007/02/turtle/primer/>

2.5.3 OWL API

OWL API²⁷ (Horridge and Bechhofer, 2011) es una API Java e implementación de referencia para crear, manipular y serializar ontologías OWL. La última versión está basada en OWL2. Es de código abierto y disponible bajo licencias de Apache y LGPL.

Incluye, entre otros, los siguientes componentes:

- Una API para OWL2 y una implementación de referencia eficiente en memoria.
- Parser y escritor para RDF/XML.
- Parser y escritor para OWL/XML.
- Parser y escritor para OWL Functional Syntax.
- Parser y escritor para Turtle.
- Interfaces de razonamiento para diferentes razonadores (en el apartado 2.8 se explican algunos de los razonadores existentes).

Desde su versión 3, está desarrollada y mantenida principalmente por la Universidad de Manchester. Actualmente se encuentra en la versión 4²⁸.

2.5.4 Herramientas de Gestión de XML

Puesto que OWL y RDF son también XML, se pueden usar herramientas de gestión y *parseo* de XML para operar sobre su estructura. Las herramientas más comunes utilizadas son:

- DOM²⁹ (“Document Object Model”) es una especificación definida por el W3C como una plataforma y una interfaz independiente del lenguaje que permite a los programas y scripts acceder dinámicamente y modificar el contenido, estructura y estilo de los documentos. DOM genera un árbol jerárquico en memoria del documento o información en XML que permite, mediante un *parser* (como por ejemplo, *Java Xerces*³⁰ de Apache), el acceso y la manipulación del contenido.
- SAX³¹ (“Simple API for XML”) fue la primera API Java para XML con más aceptación, lo que llevó a su estandarización. Actualmente se encuentra en su versión SAX 2.0 y existen versiones de SAX para otros lenguajes de programación además de Java. Su principal característica es que está basada en eventos y no en la jerarquía de árbol. Informa sobre eventos de parseo (como comienzo y final de un elemento) directamente a la aplicación y no suele construir un árbol interno por lo que ocupa menos espacio en memoria y es, por norma general, más rápido.
- JAXP³² (“Java API for XML Processing”). Como su nombre indica es la API de Java para el procesamiento de XML. Tiene soporte para DOM y SAX proporcionando una capa de abstracción sobre ellos. Permite a las aplicaciones, parseo, transformación, validación y consulta sobre documentos XML. Forma parte de la plataforma estándar de Java por lo que está incluido en su distribución.
- JDOM (Hunter and McLaughlin, 2000) obtiene una representación Java de un documento XML. JDOM proporciona una manera de representar ese documento para su

²⁷ <http://owlapi.sourceforge.net/>

²⁸ A 30 de Noviembre de 2015

²⁹ DOM <http://www.w3.org/DOM>

³⁰ Xerces <http://xerces.apache.org>

³¹ SAX <http://www.saxproject.org/>

³² JAXP <http://jaxp.java.net/>

lectura, manipulación y escritura de forma fácil y eficiente. Tiene un API sencilla, ligera y rápida y está optimizada para el programador Java. Está disponible bajo licencia de Apache de código abierto de modo que permite que los desarrolladores que usan JDOM para crear nuevos productos no tengan que lanzarlos como de código abierto obligatoriamente. Surge debido a las carencias de DOM que hacían prácticamente imposible su uso en Java, ya que es independiente del lenguaje. Por ello, JDOM está diseñado únicamente para XML y únicamente para Java. Al igual que DOM, JDOM representa un documento XML como un árbol compuesto de elementos, atributos, comentarios, nodos de texto, secciones CDATA, etc. Al contrario que SAX, el árbol está siempre disponible para acceder a él y a cualquiera de sus partes de forma directa en cualquier momento y al contrario que DOM, todos los diferentes tipos de nodos de un árbol se representan como clases en lugar de como interfaces. Además, no hay interfaz genérica, ni clase Node, que deban implementar o extender los demás nodos. Actualmente se encuentra en su versión JDOM 2. La clase SAXBuilder de JDOM usa los eventos SAX generados por cualquier parser XML para construir el árbol JDOM. El parser que usa JDOM por defecto es el seleccionado por JAXP (ya que JDOM usa JAXP si está disponible), pero JDOM puede usar prácticamente cualquier parser que se le indique. A diferencia de JAXP, JDOM no contiene un parser.

- XPath³³ (XML Path Language) (Berglund et al., 2003) es una recomendación W3C y debe su nombre al uso de una notación de ruta para navegar a través de la estructura jerárquica de un documento XML. Es una sintaxis no-XML, para la definición de partes de un documento XML facilitando su uso con URIs y valores de atributos XML, por lo que permite buscar y seleccionar teniendo en cuenta la estructura jerárquica del XML. Contiene una biblioteca de funciones estándar que incluye más de 100 funciones *built-in*. Hay funciones para cadenas, valores numéricos, comparación de fecha y hora, manipulación QName y nodos, manipulación de secuencias, valores booleanos, etc. XPath es un elemento importante en el estándar XSLT.
- XSLT³⁴ (Extensible Stylesheet Language Transformations) o Transformaciones XSL. Es un lenguaje para transformar documentos XML en nuevos documentos XML. Una transformación en XSLT se expresa en forma de hoja de estilos XSL³⁵, cuya sintaxis es XML bien formado. Una transformación expresada en XSLT describe reglas para la transformación de cero o más árboles de código fuente en uno o más árboles de resultados. La estructura de estos árboles se describe mediante el Modelo de Datos XPath³⁶. La transformación se logra mediante un conjunto de reglas de plantilla o “templates”. Una template asocia un patrón, que coincide con los nodos en el documento de origen, con un constructor de secuencia. En muchos casos, la evaluación de dicho constructor causará la formación de nuevos nodos, que pueden ser utilizados para producir parte de un árbol resultado, cuya estructura puede ser completamente diferente de la estructura de los árboles del código fuente original. En la construcción de un árbol de resultado, los nodos de los árboles del código fuente pueden ser filtrados y reordenados, y se puede añadir una estructura arbitraria. Este mecanismo permite que

³³ Versión actual 3.1 <http://www.w3.org/TR/xpath-3/>

³⁴ Versión actual 2.0 <http://www.w3.org/TR/xslt20/>

³⁵ <http://www.w3.org/TR/xsl/>

³⁶ <http://www.w3.org/TR/xpath-datamodel-3/>

una hoja de estilo pueda ser aplicable a una amplia clase de documentos que tienen estructuras de árbol similares.

2. 6. Lógica Descriptiva

La Lógica Descriptiva (Description Logics o DL) es una familia de lenguajes basados en lógica para la representación del conocimiento que se puede utilizar para representar el conocimiento de la terminología de un dominio de aplicación de una manera estructurada. El nombre está motivado por el hecho de que, por un lado, las ideas importantes del dominio son descritas mediante descripciones de conceptos que son expresiones que se construyen a partir de conceptos atómicos (predicados unarios) y funciones atómicas (predicados binarios), que utilizan los constructores propios de DL de concepto y función (o rol) proporcionado por el DL en particular, por otro lado, las DL difieren de sus predecesores, tales como redes semánticas y marcos, en que están equipadas con una semántica formal, basada en la lógica (Baader et al., 2003) y (Baader et al., 2005).

Por lo tanto, la DL es un potente formalismo para la representación del conocimiento con una semántica formal bien definida para describir dominios de aplicación (el mundo) de forma estructurada, y en el que la prestación de servicios de razonamiento automático desempeña un papel central. Para ello, primero se definen los conceptos relevantes del dominio mediante propiedades y roles, así como sus relaciones (ésta es conocida como la parte terminológica o TBox) y, posteriormente, estos conceptos son utilizados para especificar propiedades concretas sobre individuos del dominio (parte de aserciones o ABox³⁷).

En lo que a los procesos de razonamiento se refiere, éstos se encaminan fundamentalmente a comprobar propiedades de los conceptos definidos en una terminología, así como deducir conocimiento implícito a partir de su definición.

Algunas de las propiedades clásicas verificables son:

- Satisfacibilidad de un concepto: un concepto es satisfacible en una terminología τ si la definición de dicho concepto denota un conjunto de individuos que puede no ser vacío.
- Subsunción de conceptos: un concepto D subsume a otro concepto C , ($C \sqsubseteq D$), si D es considerado más general que C . En otras palabras, la subsunción comprueba si el conjunto denotado por C es un subconjunto del denotado por D .
- Consistencia de una base de conocimiento (conjunto formado por una terminología y las declaraciones acerca de las instancias individuales de los conceptos que define): una base de conocimiento es consistente si cada concepto admite la existencia de al menos una instancia individual.

2.6.1 Decidibilidad y complejidad computacional en DL

La semántica de las fórmulas usadas en las lógicas descriptivas se define estableciendo su equivalencia con expresiones en lógica de predicados de primer orden. Más concretamente, la semántica de los distintos tipos de conceptos representados mediante lógicas descriptivas se pueden expresar haciendo uso de diferentes fragmentos (o subconjuntos) de la lógica de primer orden para los que, a su vez, se pueden definir diferentes técnicas de razonamiento especializadas y variables en su nivel de complejidad (Baader et al., 2003).

³⁷ ABox proviene de "Assertional" pero en español no existe el adjetivo de aserción

Puesto que la Lógica Descriptiva es un formalismo de representación del conocimiento, y como tal, se supone que dará respuesta a las consultas de un usuario en un tiempo razonable, los investigadores en procedimientos para razonadores en DL centran la mayor parte de su interés en los procesos de decisión; esto es, a diferencia de como ocurre, por ejemplo, con los demostradores de teoremas de lógica de primer orden, los procedimientos para DL deben terminar siempre, tanto para producir respuestas positivas como negativas. La garantía de respuesta en un tiempo finito no siempre implica que se produzca en un tiempo razonable. Por esta razón, la investigación de la complejidad computacional de una DL dada, con problemas de inferencia decidibles, es también una cuestión importante.

La decidibilidad y complejidad de los problemas de inferencia dependen del poder expresivo de cada DL particular. De nuevo, se presenta la cuestión del equilibrio que el diseñador de un sistema debe buscar entre la potencia expresiva de un lenguaje y el coste de gestión del mismo. Por un lado, las DLs muy expresivas es probable que presenten problemas de inferencia de alta complejidad, o incluso sean no decidibles. Por el contrario, una DL demasiado débil (aun con procesos de razonamiento eficientes), puede ser insuficiente para expresar los conceptos más importantes de un dominio de aplicación (Baader et al., 2003).

A continuación se muestra la nomenclatura, la sintaxis y semántica comúnmente utilizada para expresar fórmulas en DL, el tipo de conocimiento que se puede representar con cada una de ellas y su relación con las versiones de OWL DL.

2.6.2 Lenguajes de descripción

En lo que se refiere a la nomenclatura utilizada en DL, las descripciones más básicas corresponden a conceptos atómicos y roles atómicos (recordemos que los roles también se pueden llamar propiedades o relaciones). A partir de ellos, es posible construir de forma inductiva descripciones complejas mediante los constructores de conceptos.

Por convenio, en notación abstracta se utilizan las letras A y B para los conceptos atómicos, la letra R para los roles atómicos y las letras C y D para las descripciones de conceptos. Las distintas versiones de DLs se distinguen por los constructores que proporcionan y que determinan su poder expresivo. En este contexto, *AL* (Attributive Language) es un lenguaje de descripción que, a su vez, da nombre a una familia de lenguajes ampliamente estudiada en el ámbito de DL. Las descripciones de conceptos en *AL* se forman de acuerdo a las siguientes reglas sintácticas (Baader et al., 2003):

$C, D \rightarrow A$		(concepto atómico)
		(concepto universal)
\perp		(concepto base)
$\neg A$		(negación atómica)
$C \text{ u } D$		(intersección)
$\forall R. C$		(restricción de valores)
$\exists R.$		(cuantificación existencial limitada)

Como puede verse, en *AL* el operador de negación (\neg) sólo puede aplicarse a conceptos atómicos, y el ámbito del cuantificador existencial (\exists) se aplica siempre a todos los conceptos,

es decir, no se puede especificar a qué clase deben pertenecer todos los elementos del rango de la propiedad a que se aplica (por eso aparece el concepto universal, \forall , como su ámbito de aplicación).

Otros lenguajes de la familia *AL* tienen mayor capacidad expresiva. Cada lenguaje de esta familia representa una extensión al lenguaje básico *AL* y su nombre viene dado añadiendo una letra al final de la cadena *AL* indicando qué semántica se puede expresar (*AL*[*C*][*Y*][*P*][...]).

A continuación se enumeran algunas de las letras utilizadas para denotar distintas capacidades expresivas:

- *C*, negación de conceptos compleja, (o arbitraria, extendiendo la negación sobre conceptos atómicos que permitía *AL*; p.e., para expresar el conjunto complementario de la unión de otros conceptos);
- *U*, un concepto como resultado de la unión de otros dos (p.e., la clase Humano como la unión de Hombres y Mujeres);
- *P*, axiomas complejos para definir propiedades (o relaciones, roles, slots; p.e., composición de propiedades, disyunción de propiedades, propiedades reflexivas e irreflexivas y el complemento de una relación);
- *O*, clases o conceptos, a partir de conjuntos enumerados (p.e., podría declararse la clase Dia_de_la_semana como la formada por los elementos lunes, martes, etc.);
- *I*, la propiedad inversa de otra dada (p.e., una propiedad padre puede especificarse como la inversa de la relación hijo);
- *N*, restricciones de cardinalidad no cualificadas (p.e. owl:cardinality);
- *Q*, restricciones de cardinalidad cualificadas sobre propiedades (o relaciones, o roles), esto es, expresiones que permiten especificar la cardinalidad de una propiedad determinada y, además, la clase a la que deben pertenecer los valores que pueda tomar dicha propiedad (p.e., una persona puede tener dos progenitores biológicos, sin embargo, sólo tendrá una madre y un padre);
- *H*, jerarquía de roles. Permite añadir sub-roles a los roles (subProperty);
- *D*, uso de propiedades de tipo de datos (datatypeProperties).

De esta forma, el lenguaje *ALCI*, es un lenguaje de descripción que permite expresar la negación de un concepto cualquiera (por ejemplo, $\neg\forall R.C$), y también definir una propiedad como la inversa de otra.

Normalmente, la semántica de cada uno de los constructores del lenguaje se expresa mediante fórmulas en Lógica de Primer Orden (en adelante, *FOL* –First-order Logic–). La figura 2.11. muestra la equivalencia entre lenguaje de descripción y la semántica en *FOL* de algunos de los constructores utilizados en DLs.

DL	FOL
$a : C$	$C(a)$
$\langle a, b \rangle : P$	$P(a, b)$
$C \sqsubseteq D$	$\forall x. C(x) \rightarrow D(x)$
$P^+ \sqsubseteq P$	$\forall x, y, z. (P(x, y) \wedge P(y, z)) \rightarrow P(x, z)$
$\top \sqsubseteq \leq 1 P$	$\forall x, y, z. (P(x, y) \wedge P(x, z)) \rightarrow y = z$
$P \equiv Q^-$	$\forall x, y. P(x, y) \iff Q(y, x)$
$C_1 \sqcap \dots \sqcap C_n$	$C_1(x) \wedge \dots \wedge C_n(x)$
$C_1 \sqcup \dots \sqcup C_n$	$C_1(x) \vee \dots \vee C_n(x)$
$\neg C$	$\neg C(x)$
$\{a_1, \dots, a_n\}$	$x = a_1 \vee \dots \vee x = a_n$
$\exists P.C$	$\exists y. (P(x, y) \wedge C(y))$
$\forall P.C$	$\forall y. (P(x, y) \rightarrow C(y))$
$\geq n P.C$	$\exists y_1, \dots, y_n. \bigwedge_{1 \leq i \leq n} (P(x, y_i) \wedge C(y_i))$
$\leq (n-1) P.C$	$\forall y_1, \dots, y_n. (\bigwedge_{1 \leq i < n, i < j \leq n} y_i \neq y_j \rightarrow (\bigvee_{1 \leq i < n, i < j \leq n} y_i = y_j))$

Figura 2.11. Equivalencia entre sintaxis DL y semántica *FOL*. Fuente: (Grosz et al., 2003).

Con objeto de evitar nombres muy largos en las DL más expresivas se utiliza la abreviatura S para denotar a ALC_{R^+} , es decir, la DL que extiende a *ALC* con roles transitivos. Los miembros más prominentes de la familia-S son:

- *SIN*, extiende ALC_{R^+} (o S) con restricciones numéricas y roles inversos.
- *SHIF*, extiende S con jerarquías de roles, roles inversos y restricciones numéricas de tipo $\leq 1 R$
- *SHIQ*, extiende a S con jerarquías de roles, roles inversos y restricciones numéricas cualificadas.

En la actualidad, *SIN*, *SHIF* y *SHIQ* son algo menos expresivas de lo que su nombre indica puesto que se restringe el uso de roles en las restricciones numéricas: los roles que tienen un subrol transitivo no pueden ocurrir en restricciones numéricas (Baader et al., 2003).

La familia *SH*, como se puede deducir fácilmente, es aquella que extiende a S con jerarquías de roles, es la familia base para el desarrollo del lenguaje OWL DL. Se analiza esta relación con un poco más de detalle en el apartado siguiente.

2.6.3 Familias *AL* relacionadas con versiones de OWL DL

La semántica del lenguaje OWL DL está basada en DL. En particular, la primera versión de este lenguaje (OWL 1) estaba basada en la semántica de la lógica de descripción *SHOIN* (D), es decir, una extensión a *ALC* con soporte para expresar roles transitivos (R^+), jerarquías de roles (*H*), clases a partir de conceptos enumerados (*O*), restricciones de cardinalidad sin cualificar (*N*) y un tipo especial de roles o propiedades que relacionan instancias de clases con valores literales (cadenas que representan valores concretos, como por ejemplo, `true`) o pertenecientes a algún de tipo de dato (*D*, `datatype`).

SHOIN (D), a pesar de estar considerada como una lógica de descripción con una potencia expresiva notable, presenta algunas limitaciones para satisfacer las demandas de los usuarios de

OWL. El diseño original de este lenguaje siguió un enfoque algo conservador para asegurar la decidibilidad de los procesos de razonamiento sobre especificaciones en este lenguaje, así como, la existencia de soporte para ellos. A partir de su publicación en febrero de 2004 como estándar del W3C, el uso extensivo del lenguaje OWL (sobre todo en su versión DL), y la celebración de un congreso anual en torno al mismo, ha dado como fruto la identificación de una serie de constructores útiles con los que extender OWL DL a un coste razonable. El desarrollo y avance en paralelo de la teoría de DL también ha proporcionado las bases para llevar a cabo razonamiento con estos constructores que no formaban parte del estándar inicial de OWL DL.

Ya se expusieron en el apartado 2.3.3 algunos de los problemas y limitaciones de OWL1, como la falta de expresividad, que condujeron a un esfuerzo para su mejora resultando en la versión OWL2.

OWL2 DL extiende OWL DL con nuevos elementos, como:

- constructores adicionales para expresar, por ejemplo, restricciones de cardinalidad cualificadas sobre las propiedades, composición de propiedades y características de las propiedades (reflexiva, irreflexiva, simétrica o asimétrica);
- azúcar sintáctico para facilitar la expresión de algunos conceptos complejos;
- extensión del soporte para tipos de datos (p.e., tipos de datos definidos por el usuario a partir de otros tipos de datos primitivos de XML-Schema como `Integer` o `String`);
- nuevas anotaciones de ontologías y anotaciones de metamodelado para permitir “punning”, es decir, que una entidad de una ontología pueda comportarse como clase en un contexto, y como instancia en otro (Grau et al., 2008).

La semántica subyacente a OWL 2 se corresponde de forma unívoca con la de los constructores de la lógica de descripción *SR0IQ* (Horrocks et al., 2006) (extensión de *ALC* con roles transitivos, roles reflexivos, roles negativos, rol universal, reflexividad local, restricciones numéricas cualificadas), exceptuando las pequeñas extensiones que permite el lenguaje para soportar tipos de datos (D), como ya ocurría con la primera versión de OWL, y punning, que no están soportadas por *SR0IQ* (W3C, 2012a).

Además del TBox y el ABox, *SR0IQ* introduce el RBox para las sentencias sobre roles. En *SHIQ* y (sus familias derivadas), la aplicación de las restricciones numéricas cualificadas se restringían a ciertos roles, llamados roles simples, para preservar la decidibilidad. En el contexto de *SR0IQ*, esta definición se ha modificado ligeramente y ahora estos roles simples también aparecen en otras construcciones (Horrocks et al., 2006).

SR0IQ (con unas pequeñas restricciones) es decidible, esta decidibilidad es demostrada por Horrocks, Kutz y Sattler mediante un Algoritmo de Tableau en (Horrocks et al., 2006).

2.7. Reglas y SWRL

Las reglas nos permiten un mayor grado de expresividad del que ofrece OWL (en aras de la decidibilidad). Cuando OWL no alcanza la expresividad necesitada, se utilizan las reglas, que además de ser más expresivas, también disponen de bases de razonamiento eficiente sobre ellas y son bien conocidas en la práctica. También, en ocasiones, se utilizan reglas donde se podrían utilizar sentencias OWL para hacer aserciones de forma más sencilla o para producir una inferencia más elegante. Ya se comentó en el apartado 2.1 la existencia de la capa RIF (Rule

Interchange Format) dentro de la pila de tecnologías de la Web Semántica. Esta tecnología, como su nombre indica, no es un lenguaje en sí, sino un formato común de intercambio de reglas entre diferentes sistemas de reglas, como las reglas SWRL.

SWRL (Horrocks et al., 2004) (Semantic Web Rule Language) es un lenguaje de reglas expresivo basado en una combinación de OWL y de RuleML (Rule Markup Language) (Boley et al., 2001). SWRL permite a los usuarios escribir reglas en términos de conceptos OWL para proporcionar mayor capacidad de razonamiento deductivo del que aporta OWL. Está semánticamente construido usando la misma base de Lógica Descriptiva que OWL y proporciona garantías formales similares en el desempeño de la inferencia. SWRL extiende el conjunto de axiomas de las ontologías OWL para incluir cláusulas de Horn³⁸ pudiendo así combinar reglas de tipo Horn con una base de conocimiento OWL.

Las reglas propuestas están formadas por un antecedente y un consecuente donde el significado puede interpretarse como: siempre que las condiciones especificadas en el antecedente se cumplan, deben cumplirse las del consecuente. O lo que es lo mismo: “si el antecedente se cumple entonces el consecuente se cumple”. Tanto el antecedente como el consecuente consisten en cero o más átomos que no pueden ser ni negativos ni disjuntos:

$$\text{claus}_1 \wedge \text{claus}_2 \dots \rightarrow \text{claus}_x \wedge \text{claus}_y$$

A su vez, una cláusula es una expresión de la forma:

$$p(\text{arg}_1, \text{arg}_2, \dots, \text{arg}_n)$$

donde p es un símbolo predicado y $\text{arg}_1, \text{arg}_2, \dots, \text{arg}_n$ son los argumentos o términos de la expresión. Un antecedente vacío o sin átomos se trata como trivialmente cierto (es decir, se satisface para cada interpretación), por lo tanto, el consecuente también debe satisfacerse en cada interpretación. Un consecuente vacío se trata como trivialmente falso por lo que ni el antecedente ni el consecuente se satisfacen en interpretación alguna. Átomos múltiples se tratan como conjunciones ya que las reglas con consecuentes conjuntivos pueden ser transformadas en múltiples reglas cada una con un consecuente atómico de forma simple.

En SWRL, los símbolos predicados pueden incluir clases OWL, propiedades o tipos de datos. Los argumentos pueden ser individuos, valores de datos o variables que los referencian. Todas las variables en SWRL se tratan como universalmente cuantificadas con su ámbito limitado a una regla dada.

Existen siete tipos de átomos en SWRL (Horrocks et al., 2004):

1. *Clases*. Un átomo clase consiste en una clase OWL nombrada o una expresión de clase y un único argumento representando un individuo OWL. Por ejemplo:

$$\text{Person} (?p)$$

$$\text{Man} (\text{Fred})$$

donde Person y Man son clases OWL nombradas, $?p$ es una variable representando a un individuo OWL, y Fred es el nombre de un individuo o instancia OWL.

Una regla simple de ejemplo usando átomos clase, para declarar que todo individuo de tipo Man es también de tipo Person , podría ser:

$$\text{Man} (?p) \rightarrow \text{Person} (?p)$$

³⁸ Una cláusula de Horn es una regla de inferencia lógica con al menos un literal positivo

(por supuesto, esta sentencia se puede escribir directamente en OWL).

2. *Propiedades sobre individuos.* Un átomo de propiedad sobre individuos consiste en una propiedad objeto OWL (ObjectProperties) y dos argumentos representando individuos OWL. Por ejemplo:

$$\text{hasBrother} (?x, ?y) \text{hasSibling} (\text{Fred}, ?y)$$

donde `hasBrother` y `hasSibling` son propiedades objeto, `?x` y `?y` son variables representando individuos y `Fred` es el nombre de un individuo OWL.

Una regla de ejemplo para explicar que una persona con `sibling`³⁹ masculino tiene un hermano, requerirá capturar los conceptos de OWL: `persona`, `hombre (masculino)`, `sibling` y `hermano`. El concepto de `hombre` y `persona` puede ser capturado usando una clase OWL `Person` con una subclase `Man`; las relaciones `hermano` y `sibling` se pueden expresar mediante propiedades objeto de OWL `hasSibling` y `hasBrother` con dominio en el rango de `Person`. La regla SWRL, por tanto, sería:

$$\text{Person} (?p) \wedge \text{hasSibling} (?p, ?s) \wedge \text{Man} (?s) \rightarrow \text{hasBrother} (?p, ?s)$$

Se asocia así la propiedad `hasBrother` con todos los individuos OWL con uno o más `siblings` masculinos y se asigna su valor a esos `siblings`.

3. *Propiedades de valores de datos.* Consiste en una propiedad de datos (DatatypeProperty) y dos argumentos, el primero es un individuo y el segundo un valor. Por ejemplo:

$$\text{hasHeight} (\text{Fred}, ?h) \text{hasAge} (?x, 232) \text{hasName} (?x, \text{"Fred"})$$

donde `hasHeight`, `hasAge`, y `hasName` son propiedades de datos.

Una regla ejemplo con un valor booleano en la propiedad sería afirmar que todas las personas que poseen un coche deberían ser clasificadas como conductoras:

$$\text{Person} (?p) \wedge \text{hasCar} (?p, \text{true}) \rightarrow \text{Driver} (?p)$$

Los individuos nombrados (`NamedIndividuals`) también pueden referirse directamente. Por ejemplo, se reescribe la regla anterior para clasificar a `Fred` (que debe existir previamente en la ontología) como conductor:

$$\text{Person} (\text{Fred}) \wedge \text{hasCar} (\text{Fred}, \text{true}) \rightarrow \text{Driver} (\text{Fred})$$

4. *Mismo individuo.* Mediante el símbolo `sameAs` se afirma que los dos argumentos individuos son el mismo. Por ejemplo:

$$\begin{aligned} &\text{sameAs} (?x, ?y) \\ &\text{sameAs} (\text{Fred}, \text{Freddy}) \end{aligned}$$

5. *Individuos diferentes.* Mediante el símbolo `differentFrom` se afirma que los dos argumentos individuos son diferentes. Por ejemplo:

$$\begin{aligned} &\text{differentFrom} (?x, ?y) \\ &\text{differentFrom} (\text{Fred}, \text{Joe}) \end{aligned}$$

³⁹ en OWL la relación `hasSibling` se refiere a dos clases o individuos que son subclase de una misma clase.

6. *Rango de datos*. Consiste en un nombre de tipo de datos (`datatype`) o un conjunto de literales y un único argumento representando un valor:

$$\text{xsd : int(?x) [3, 4, 5] (?x)}$$

7. Incorporados (*built-ins*). Una de las características más potentes de SWRL es su habilidad de soportar incorporaciones construidas por el usuario. Un *built-in* o incorporación es un predicado que toma uno o más argumentos y los evalúa a ciertos si los argumentos satisfacen el predicado. Existen unos *built-ins* predefinidos, llamados “core built-ins”⁴⁰ para operaciones matemáticas y de cadenas comunes, por convención, cuando se usan en reglas van precedidos del calificador de espacio de nombres `swrlb`. Por ejemplo, para indicar que una persona mayor de 17 años es un adulto:

$$\text{Person(?p \wedge hasAge(?p, ?age \wedge swrlb : greaterThan(?age, 17) \rightarrow Adult(?p)}$$

Los *built-ins* pueden tomar cualquier combinación de valores de propiedades de tipos de datos. Los tipos de datos pueden ser XML Schema o tipos definidos por usuario. Pero no pueden tomar ni clases ni objetos ni valores de propiedad (existe un extensión llamada *SWRLTab* que sí lo permite).

La propuesta de SWRL del W3C (Horrocks et al., 2004) no proporciona un mecanismo para especificar el número o el tipo de los argumentos de los *built-ins*, por lo que el chequeo de los argumentos es responsabilidad del implementador del *built-in*. Sin embargo, sí que especifica que un *built-in* al que le pasen un número incorrecto de argumentos, o de tipo erróneo, debería evaluar a falso. SWRL permite que los usuarios construyan nuevas librerías de *built-ins* para emplearse en las reglas, por ejemplo, conversión de moneda, manipulaciones temporales o búsquedas taxonómicas. Las descripciones de clase son bloques básicos de axiomas de clase que la describen y también son soportadas por SWRL. Ejemplo de una regla que clasifica a un individuo como padre si es un miembro de una clase con la propiedad `hasChild` con una cardinalidad mínima de uno:

$$(\text{hasChild} \geq 1) (?x) \rightarrow \text{Parent} (?x)$$

Como OWL (al igual que SWRL) sigue la asunción del mundo abierto esta regla también haría “match” con aquellos individuos que no tienen valores para la propiedad `hasChild` en la ontología actual, pero para las cuales la existencia de esos valores podría deducirse de los axiomas OWL. No es posible expresar este tipo de matching en SWRL. Hay que tener presente que un objetivo fundamental de SWRL es aumentar OWL, por lo que se espera que los usuarios de OWL quieran utilizar estas reglas y utilizarlas en combinación con un razonador en sus ontologías existentes. Por lo tanto, los razonadores que se vayan a usar deben tener soporte para dichas reglas. Pero SWRL es indecidible, es, de hecho, semi-decidible. Si es un axioma es implicado, entonces hay un algoritmo que lo puede demostrar, pero si un axioma no está implicado, es posible entrar en un bucle infinito. Por este motivo, se añade una restricción a SWRL: las variables sólo podrán ligarse a individuos nombrados. Esta propiedad se conoce como DL Safety y el nuevo SWRL restringido, como SWRL DL Safe. Añadir esta restricción convierte a SWRL en decidible.

⁴⁰ Un número de core built-ins vienen recogidos en <http://www.daml.org/rules/proposal/builtins.html>

Nota: En este trabajo se ha utilizado SWRL. Aunque no ha llegado nunca a ser un estándar del W3C, hasta el momento tampoco existe otro lenguaje estándar de reglas para complementar una ontología OWL2 DL.

2. 8. El razonamiento ontológico

La equivalencia de OWL y la Lógica Descriptiva es lo que permite a OWL explotar el razonamiento para cubrir importantes requerimientos lógicos como satisfacibilidad de conceptos, subsunción de clases, consistencia de clases, equivalencia de clases y comprobación de instancias. Uno de los usos de la semántica formal es permitir a los humanos razonar sobre el conocimiento. Los tipos de razonamiento que se pueden llevar a cabo son (Antoniou and Harmelen, 2003):

- **Pertenencia a clases:** si x es una instancia de una clase C y C es subclase de D , se puede inferir que x es una instancia de D .
- **Equivalencia de clases:** si la clase A es equivalente a la B y la clase B es equivalente a la C , entonces A es equivalente a C .
- **Consistencia:** las declaraciones deben ser consistentes en la ontología o podrían llevar a errores.
- **Clasificación:** si se ha declarado que una propiedad es condición suficiente para pertenecer a una clase A , entonces si un individuo X satisface dicha condición debe ser una instancia de A .

Un razonador es una pieza de software capaz de deducir las consecuencias lógicas de un conjunto de hechos afirmados o axiomas. La noción de un razonador semántico generaliza la de un motor de inferencia, al ofrecer un conjunto más amplio de mecanismos para trabajar con ellos. Las reglas de inferencia comúnmente se especifican mediante un lenguaje de ontologías y con frecuencia un lenguaje de descripción. Algunos de los razonadores que son capaces de trabajar sobre OWL y que se pueden utilizar tanto en Protégé como en OWL API son FaCT++, HermiT y Pellet se explican a continuación.

3.8.1 FaCT++

FaCT⁴¹ (Fast Classification of Terminologies) es un clasificador de lógica descriptiva (DL) que también se puede utilizar para las pruebas de satisfacibilidad de lógica modal. El Sistema de Información incluye dos razonadores, los cuales utilizan algoritmos de Tableaux robustos y completos. FaCT++ (Tsarkov and Horrocks, 2006) es la nueva generación de FaCT que usa los algoritmos FaCT pero con diferente arquitectura interna y razona sobre OWL DL y (parcialmente) OWL2. Está implementado en C++ para crear herramientas software más eficientes y facilitar la portabilidad. El razonador implementa un procedimiento de decisión de tableaux de *SHOIQ*, con apoyo adicional para los tipos de datos, incluyendo cadenas y enteros. FaCT++ se distribuye bajo una licencia pública de GNU y está disponible para su descarga, tanto como un archivo binario y como el código fuente⁴².

⁴¹ FaCT <http://www.cs.man.ac.uk/~horrocks/FaCT/>

⁴² FaCT++ <http://code.google.com/p/factplusplus/>

3.8.2 Hermit

Hermit⁴³ (Shearer et al., 2008) es un razonador de ontologías en el lenguaje OWL. Dado un fichero OWL, Hermit puede determinar al menos si la ontología es consistente e identificar las relaciones entre las clases de subsunción. Hermit es el primer razonador OWL a disposición del público que está basado en un cálculo hypertableau novedoso y que ofrece un razonamiento mucho más eficiente que cualquier algoritmo previamente conocido. Las ontologías, que antes requerían horas o minutos para clasificarse, pueden a menudo clasificarse en segundos gracias a Hermit, y además es el primer razonador capaz de clasificar ontologías que se han tomado como demasiado complejas de manejar anteriormente por cualquier sistema disponible. Hermit utiliza la semántica directa y pasa todas las pruebas de conformidad OWL2 para razonadores de semántica directa. Es de código abierto y se distribuye bajo licencia GNU.

3.8.3 Pellet

Pellet⁴⁴ (Sirin et al., 2007) es un razonador de código abierto para OWL 2 DL en Java e incluye soporte para OWL2 EL. Proporciona servicios estándar de última generación de razonamiento para ontologías OWL. Pellet proporciona funcionalidades para ver la validación de especies, chequear consistencia de ontologías, clasificar la taxonomía, comprobar implicaciones lógicas y contestar consultas ABox de DL. Las consultas pueden ser formuladas mediante SPARQL y tiene buen soporte para reglas SWRL (Sirin et al., 2007). Los componentes principales del razonador Pellet pueden verse en la figura 2.12.

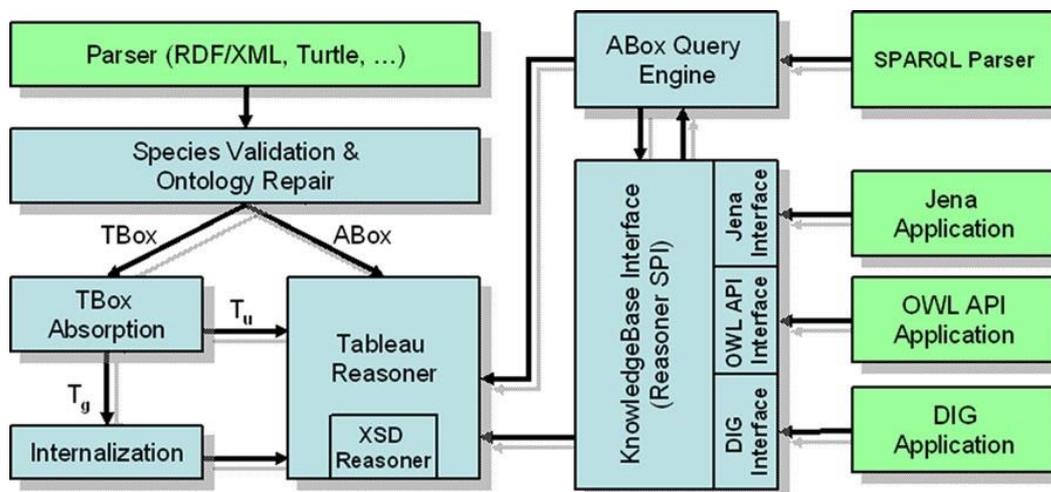


Figura 2.12. Componentes principales de Pellet. Fuente: (Sirin et al., 2007).

Referencias del capítulo

- ANTONIOU, G. & HARMELEN, F. V. 2003. Web Ontology Language: OWL. *Handbook on Ontologies in Information Systems*. Springer-Verlag.
- BAADER, F., HORROCKS, I. & SATTLER, U. 2003. Description Logics as ontology languages for the Semantic Web. *In Festschrift in honor of Jörg Siekmann, Lecture Notes in Artificial Intelligence*. Springer-Verlag.

⁴³ Hermit <http://www.hermit-reasoner.com/>

⁴⁴ Pellet <http://clarkparsia.com/pellet/>

- BAADER, F., HORROCKS, I. & SATTTLER, U. 2005. Description logics as ontology languages for the semantic web. *Mechanizing Mathematical Reasoning*. Springer Berlin Heidelberg.
- BERGLUND, A., BOAG, S., CHAMBERLIN, D., FERNÁNDEZ, M. F., KAY, M., ROBIE, J. & SIMÉON, J. 2003. Xml path language (xpath). *World Wide Web Consortium (W3C)*.
- BERNERS-LEE, T., HENDLER, J. & LASSILA, O. 2001. The semantic web. *Scientific american*, 284, 28-37.
- BOLEY, H., TABET, S. & WAGNER, G. 2001. Design Rationale for RuleML: A Markup Language for Semantic Web Rules. *SWWS (Vol. 1, pp. 381-401)*, 1, 381-401.
- BORST, W. N. 1997. *Construction of engineering ontologies for knowledge sharing and reuse*. Universiteit Twente.
- CHANDRASEKARAN, B., JOSEPHSON, J. R. & BENJAMINS, V. R. 1999. What are ontologies, and why do we need them? *IEEE Intelligent Systems*, 1, 20-26.
- GARSHOL, L. M. 2004. Metadata? Thesauri? Taxonomies? Topic maps! Making sense of it all. *Journal of information science*, 30, 378-391.
- GILCHRIST, A. 2003. Thesauri, taxonomies and ontologies-an etymological note. *Journal of documentation*, 59, 7-18.
- GÓMEZ-PÉREZ, A., FERNÁNDEZ-LÓPEZ, M. & CORCHO, O. 2007. *Ontological Engineering: with examples from the areas of Knowledge Management, e-Commerce and the Semantic Web*, (Advanced Information and Knowledge Processing). Secaucus, NJ, USA, Springer-Verlag New York.
- GRAU, B. C., HORROCKS, I., MOTIK, B., PARSIA, B., PATEL-SCHNEIDER, P. & SATTTLER, U. 2008. OWL 2: The next step for OWL. *Web Semantics: Science, Services and Agents on the World Wide Web*, 6, 309-322.
- GRUBER, T. R. 1995. Toward principles for the design of ontologies used for knowledge sharing? *International journal of human-computer studies*, 43, 907-928.
- GRUBER, T. R. 1993. A translation approach to portable ontology specifications. *Knowledge Acquisition*, 5, 199-220.
- GRUNINGER, M., ATEFI, K. & FOX, M. S. 2000. Ontologies to support process integration in enterprise engineering. *Computational & Mathematical Organization Theory*, 6, 381-394.
- GUARINO, N. 1998. *Formal Ontology in Information Systems: Proceedings of the 1st International Conference June 6-8, 1998, Trento, Italy*, IOS Press.
- HOEKSTRA, R. Ontology representation design patterns and ontologies that make sense. Proceedings of the 2009 conference on Ontology Representation, 2009 Amsterdam, The Netherlands. IOS Press, 1-236.
- HORRIDGE, M. & BECHHOFFER, S. 2011. The OWL API: A Java API for OWL ontologies. *Semantic Web*, 2, 11-21.
- HORROCKS, I., KUTZ, O. & SATTTLER, U. The Even More Irresistible SROIQ. Proceedings of the 10th Int. Conf. on Principles of Knowledge Representation and Reasoning (KR2006), 2006. AAAI Press, 57-67.
- HORROCKS, I., PATEL-SCHNEIDER, P. & HARMELEN, F. V. 2003. From SHIQ and RDF to OWL: The making of a web ontology language. *Journal of Web Semantics*, 1, 7-26.
- HORROCKS, I., PATEL-SCHNEIDER, P. F., BOLEY, H., TABET, S., GROSOFF, B. & DEAN, M. 2004. *SWRL: A semantic web rule language combining OWL and RuleML* [Online]. W3C Member submission. Available: <http://www.w3.org/Submission/SWRL/> [Accessed 30 November 2015].
- HUNTER, J. & MCLAUGHLIN, B. 2000. *The jdom project* [Online]. Available: <http://www.jdom.org> [Accessed 3th March 2015].
- LASSILA, O. & MCGUINNESS, D. 2001. The role of frame-based representation on the semantic web. Technical Report KSL-01-02. Knowledge Systems Laboratory, Stanford University, Stanford, CA.

- MCBRIDE, B. I. S. Jena: Implementing the RDF Model and Syntax Specification. Proceedings of 2nd Int. Workshop on the Semantic Web, 2001.
- MCGUINNESS, D. L. 2003. Ontologies come of age. *Spinning the Semantic Web: Bringing the World Wide Web to Its Full Potential*. MIT Press.
- MCGUINNESS, D. L. & HARMELEN, F. V. 2004. OWL web ontology language overview. *W3C recommendation, W3C*.
- MILES, A., MATTHEWS, B., WILSON, M. & BRICKLEY, D. SKOS Core: Simple knowledge organisation for the web. International Conference on Dublin Core and Metadata Applications, 2005.
- NOY, N. F. & MCGUINNESS, D. L. 2001. Ontology development 101 a guide to creating your first ontology. Technical Report. Stanford Knowledge Systems Laboratory: Stanford University, Stanford, AC.
- OWL WORKING GROUP. 2012. *OWL (Web Ontology Language)* [Online]. W3C. Available: <http://www.w3.org/2001/sw/wiki/OWL> [Accessed 26th November 2015].
- PROTÉGÉ. 2013. *Protégé ontology editor* [Online]. Available: <http://protege.stanford.edu/> [Accessed 30 Nov 2015].
- SHEARER, R., MOTIK, B. & HORROCKS, I. 2008. HermiT: A Highly-Efficient OWL Reasoner. *OWLED*, 432, 91.
- SIRIN, E., PARSIA, B., GRAU, B. C., KALYANPUR, A. & KATZ, Y. 2007. Pellet: A practical owl-dl reasoner. *Web Semantics: science, services and agents on the World Wide Web*, 5, 51-53.
- STUDER, R., BENJAMINS, V. R. & FENSEL, D. 1998. Knowledge engineering: Principles and methods. *Data & Knowledge Engineering*, 25, 161 – 197.
- TSARKOV, D. & HORROCKS, I. 2006. FaCT++ description logic reasoner: System description. *Automated reasoning. Springer Berlin Heidelberg*, 292-297.
- VAN HEIJST, G., SCHREIBER, A. T. & WIELINGA, B. J. 1997. Using explicit ontologies in KBS development. *International Journal of Human-Computer Studies*, 46, 183–292.
- W3C. 2001. *DAML+OIL Reference Description* [Online]. Available: <http://www.w3.org/TR/daml+oil-reference> [Accessed 3th March 2015].
- W3C. 2004. *OWL Web Ontology Language Reference* [Online]. Available: <http://www.w3.org/TR/owl-ref/> [Accessed 3th March 2015].
- W3C. 2009. *SKOS Simple Knowledge Organization System Reference* [Online]. Available: <http://www.w3.org/TR/skos-reference/> [Accessed 3th March 2015].
- W3C. 2012a. *OWL 2 Direct Semantics (Second Edition)* [Online]. Available: <http://www.w3.org/TR/owl2-direct-semantics/> [Accessed 30 November 2015].
- W3C. 2012b. *SPARQL 1.1* [Online]. Available: <http://www.w3.org/TR/sparql11-query/> [Accessed 10 November 2015].
- W3C. 2013. *RIF Overview (Second Edition)* [Online]. Available: <http://www.w3.org/TR/rif-overview/> [Accessed 10 October 2015].
- W3C. 2014a. *RDF Current Status* [Online]. Available: http://www.w3.org/standards/techs/rdf#w3c_all [Accessed 26th November 2015].
- W3C. 2014b. *RDF Schema 1.1 W3C Recommendation* [Online]. Available: <http://www.w3.org/TR/rdf-schema/> [Accessed 6th November 2015].
- W3C. 2014c. *Web Cryptography API - W3C Candidate Recommendation* [Online]. Available: <http://www.w3.org/TR/WebCryptoAPI/> [Accessed 10 October 2015].

CAPÍTULO 3

SISTEMAS DE RECOMENDACIÓN SEMÁNTICOS

En el capítulo 1 se abordó el estudio de los Sistemas de Recomendación, cuáles son sus objetivos y su funcionamiento básico. También se expusieron algunas limitaciones de esos sistemas, en particular, que la comunicación entre los agentes participantes, incluso entre agentes y usuarios, se encuentra limitada debido a la heterogeneidad en la representación de los datos. Esta heterogeneidad provoca también que la información no se pueda reutilizar en otros procesos o aplicaciones. Ambas carencias van en detrimento de la calidad de las recomendaciones. Una de las tendencias más utilizadas en los últimos años para solventar estas limitaciones es utilizar las técnicas de la Web Semántica en su desarrollo, y que se han introducido en el capítulo 2. En el presente capítulo se llevará a cabo un estudio más detallado acerca de los Sistemas de Recomendación Semánticos y cómo utilizan las técnicas de la Web Semántica en aras de mejorar estas deficiencias y conseguir una mayor calidad en las recomendaciones ofrecidas a los usuarios.

3.1. Introducción

Los Sistemas de Recomendación (SR) estudiados en el capítulo 1 presentaban algunas carencias (como los problemas del arranque en frío, o del nuevo usuario). Los Sistemas Híbridos permitían enriquecer las recomendaciones debido a que explotaban las ventajas de los paradigmas basados en contenido y colaborativos suavizando, a su vez, sus limitaciones individuales (Adomavicius and Tuzhilin, 2005). A pesar de ello, estos sistemas muestran aún algunas deficiencias.

Los SR Basados en Contenido, para su funcionamiento, parten generalmente de técnicas de filtrado de información relacionadas con la recuperación de texto. Utilizan palabras clave para hacer comparaciones (matching) entre los atributos de los ítems y las necesidades o gustos de los usuarios a partir de la información recuperada. Sin embargo, la mayoría de estos sistemas no intentan comprender al usuario. Se puede decir que su funcionamiento se basa en la sintaxis del contenido, lo que conlleva una “falta de inteligencia” (Lops et al., 2011). En el momento en que se requieren características más avanzadas, estas técnicas basadas en palabras clave empiezan a mostrar sus limitaciones.

Otros problemas que quedan sin resolver en los Sistemas Híbridos tradicionales son que requieren muchas recomendaciones de “entrenamiento” (con la participación del usuario) antes de proporcionar recomendaciones más precisas, o la dificultad para ofrecer explicaciones o justificaciones de las recomendaciones, lo cual disminuye su confianza por parte del usuario (McSherry, 2005).

Se comentó también anteriormente que las principales carencias de los SR estudiados tenían su origen en la heterogeneidad en la representación de los datos, que provocaba que los sistemas sólo pudieran trabajar intrínsecamente, es decir, que no pudieran mejorar sus recomendaciones reutilizando el conocimiento o aprendizaje de otros sistemas existentes, ni la comunicación con otros agentes o aplicaciones capaces de trabajar en dominios similares o incluso complementarios. Esta circunstancia es conocida, en el ámbito de los SR, como el

problema del “portfolio” (Burke, 2002), ya que sólo se trabaja con elementos representados en el sistema y no se pueden descubrir nuevos ítems ni nuevas propiedades para mejorar la calidad de las recomendaciones: *lo que no está, no existe*.

Una de las propuestas más extendidas en los últimos años para afrontar estos problemas es la inclusión de las técnicas de la Web Semántica en los SR, dando lugar a los conocidos como Sistemas de Recomendación Semánticos (SRS). La línea de investigación más generalizada en este tipo de sistemas es la producción de recomendaciones a partir de un conocimiento *semántico* del dominio (Wang and Kong, 2007).

3. 2. Características de los Sistemas de Recomendación Semánticos

Las estrategias adoptadas para introducir semántica en los procesos de recomendación de los SRS pueden describirse según diversos criterios (Lops et al., 2011):

- El tipo de base de conocimiento sobre el que se sostiene.
- La caracterización del perfil del usuario.
- Las técnicas adoptadas para la representación de los ítems o anotaciones.
- Estrategias de *matching* entre los ítems y los usuarios.

3.2.1 Tipo de base de conocimiento

Existen diferentes modelos a partir de los cuales se sustenta el conocimiento de los ítems contenidos en una base de conocimiento semántico.

Dentro del ámbito de la extracción de información a partir de documentos de texto, está muy extendido el uso de lexicones o tesauros externos al sistema, como WordNet⁴⁵, que permite aportar sentido semántico y la desambiguación de términos (Magnini and Strapparava, 2001), (Degemmis et al., 2007).

Otro modelo semántico utilizado son las folksonomías y las taxonomías (Ziegler et al., 2004). Las folksonomías consisten en anotaciones semánticas creadas por la comunidad de usuarios para la clasificación de documentos. Las taxonomías extienden a las anteriores incluyendo jerarquías y relaciones de parentesco entre los términos utilizados para la anotación (etiquetas o tags). Podría decirse que las folksonomías son listas de palabras clave, mientras que las taxonomías están organizadas jerárquicamente (relaciones *is-a*).

Las ontologías aportan el conocimiento sobre el dominio clasificando también los ítems en categorías y recogiendo relaciones entre los objetos que la forman. Estas relaciones proporcionan mayor potencia semántica que las anteriores y favorecen el razonamiento sobre los ítems del dominio, por lo que son capaces de aportar “inteligencia semántica”.

El uso de estos modelos de bases de conocimiento semántico se puede encontrar mezclado en varios sistemas, por ejemplo: extracción de términos del lexicón para la construcción de taxonomías, folksonomías u ontologías del dominio específico.

⁴⁵WordNet es un lexicón o una base de datos léxica para el idioma inglés que agrupa los sinónimos en grupos llamados *synsets* con definiciones breves y contiene algunas relaciones semánticas entre los conjuntos.

3.2.2 Modelado del usuario y perfiles de usuario

La representación efectiva y concisa de los usuarios en los SR es un elemento clave que condiciona, en la mayoría de los casos, la forma en que se obtengan las recomendaciones puesto que éstas están basadas principalmente en las preferencias y necesidades de los usuarios (Trajkova and Gauch, 2004). Un conocimiento detallado sobre el usuario permite al sistema no sólo deducir *qué* recomendaciones ofrecerle sino también *cómo* y *cuándo*.

Los modelos de usuario representan características del usuario que vienen principalmente determinadas por el dominio de conocimiento y el contexto del sistema. Una instanciación de este modelo se conoce como un perfil de usuario. Es posible adquirir información considerando perfiles de usuario de diferentes fuentes para construir un modelo para un sistema o aplicación (Niedritis et al., 2011).

La adquisición del conocimiento sobre los usuarios para su representación mediante un modelo de usuario puede realizarse de dos formas: *explícitamente*, donde el usuario participa activamente, como por ejemplo, contestando cuestionarios o rellenando formularios, o *implícitamente*, también conocida como *no-intrusiva*, donde el usuario participa pasivamente, se estudia su comportamiento o se infieren sus intereses y características sin que éste lo note. Es común usar ambos enfoques en sistemas adaptativos para construir un mismo perfil; en primer lugar, cuando el usuario se registra en el sistema, se le preguntan sus datos demográficos (género, edad, nacionalidad, etc.) y, si es necesario, algunas cuestiones básicas sobre sus gustos o intereses. Conforme va utilizando el sistema, se mantiene su perfil aprendiendo sus preferencias de forma implícita mediante el registro y la observación de su actividad.

Según (Middleton et al., 2004) el modelado del usuario se basa típicamente en conocimiento o en comportamiento. Un enfoque basado en conocimiento construye modelos estáticos de los usuarios y realiza matchings dinámicos de los usuarios al modelo más cercano. Un enfoque basado en comportamiento usa la conducta del usuario en el sistema como un modelo, empleando generalmente técnicas de inferencia automática para descubrir patrones útiles de comportamiento. Ambos enfoques pueden modelarse mediante ontologías. La representación ontológica del modelo de usuario permite el análisis de preferencias en diferentes niveles de abstracción y permite aprovechar las capacidades de razonamiento en un sistema de recomendación (Sieg et al., 2007).

Algunos de estos modelos ontológicos son sub-ontologías ponderadas de la ontología del dominio. Por ejemplo, en (Middleton et al., 2004) se recomiendan artículos de investigación *online* a usuarios, cuyos perfiles se modelan como sub-ontologías de una ontología de *topics* donde cada clase tiene un valor de *interés* según el historial de navegación del usuario. En otros enfoques, los perfiles ontológicos son conceptualizaciones personalizadas independientes de la estructura de la ontología del dominio aunque relacionadas con ella, persiguiendo la reutilización de dichos perfiles en dominios similares. Por ejemplo, en (Razmerita et al., 2003), el usuario se representa mediante una ontología que describe sus características, permitiendo identificar necesidades personalizadas para cada usuario. A partir de ahora nos centraremos en este segundo enfoque.

Una de las principales cuestiones a considerar al crear un modelo de usuario ontológico en un dominio particular es la elección entre la creación de la ontología desde cero o la reutilización de una ya existente. Encontrar una ontología en el mismo dominio que se adapte a los requisitos de un sistema no es una tarea fácil, ya que la mayoría son demasiado dependientes de la utilización final del sistema para el que fueron creados. En la mayoría de los casos, estas

ontologías necesitan un proceso de modularización y la adaptación compleja de modo que puedan ser reutilizados por otros sistemas o agentes (Thakker et al., 2011).

El sistema debe ser capaz de separar suposiciones acerca de los usuarios que figuran en el modelo de usuario del resto de los conocimientos del sistema (Wahlster and Kobsa, 1989). La representación del usuario debe permitir recoger la información del mismo como parte del sistema mediante operaciones ontológicas y a su vez, extraerlo como un todo cuando sea necesario. Este principio ha de ser considerado durante el diseño del modelo de usuario.

Otro principio es que una ontología debe ser *monotónicamente* extensible (Gruber, 1995), es decir, debe ser posible definir nuevos términos en la ontología basados en el vocabulario existente sin tener que revisar las definiciones previas. Cuando se trata de un SRS, es posible utilizar una ontología general para modelar el perfil de usuario de modo que se puede ampliar el dominio específico.

Modelos de usuario ontológicos generales

Uno de los primeros modelos de usuario ontológicos generales se introdujo en (Razmerita et al., 2003), donde los autores proponen OntobUM, una arquitectura ontológica de modelado genérico de usuario usando el formato RDF / RDFS para almacenar datos. En (Heckmann et al., 2005), los autores afirman que una ontología de usuario, en general, debe estar representada en un idioma semántico moderno como OWL y ser accesible *online* para todos los sistemas adaptativos que lo requieran. Proponen la ontología GUMO –fusionando SUMO (Niles and Pease, 2001) y Ubisworld (Heckmann and Krueger, 2003)– que puede ampliarse con las técnicas de inferencia o nuevo conocimiento sobre el dominio específico. Una ventaja de esta propuesta es la simplificación para el intercambio de modelo de usuario entre diferentes sistemas de adaptación del usuario a través del servicio de modelo de usuario `u2m.org` por peticiones HTTP simples.

En (Loizou and Dasmahapatra, 2006), los autores proponen un formato de perfil de usuario unificado que puede extenderse a múltiples aplicaciones. Este perfil se modela a través del Modelo de Contexto de Usuario Unificado (UUCM – Unified User Context Model), que representa las características y el contexto de los usuarios a través de *facets* que apoyan probabilidades y cualificadores. Estos cualificadores están ligados a ontologías o vocabularios con el fin de apoyar la personalización entre sistemas.

En (Gauch et al., 2007), se define una estrategia para crear una ontología de dominio de usuario que puede extenderse considerando otros modelos existentes. Sin embargo, como la ontología sólo incluye un modelo de usuario estático, no tiene en cuenta el comportamiento del usuario en el sistema, por lo que no es capaz de extraer patrones o tendencias de comportamiento.

3.2.3 Estrategias de matching ítem-usuario

La representación de los ítems mediante una base de conocimiento semántica ofrece nuevas oportunidades en los procesos de *matching* de los diferentes paradigmas de recomendación.

En el enfoque basado en restricciones del paradigma de conocimiento es posible definir nuevas restricciones tanto a partir de la clasificación de un ítem o sus propiedades en la base de conocimiento ontológica como mediante reglas SWRL. Lo mismo ocurre en los enfoques basados en casos y en contenido, donde además se aplican técnicas que calculan similitudes

semánticas entre los requisitos del usuario y las propiedades de los ítems o entre los ítems del dominio. Estos dos enfoques utilizan también técnicas de similaridad para extraer ítems similares a los que se conoce como *interesantes* o de utilidad al usuario.

Similaridad Semántica

En (Lee et al., 2008) se define la similaridad semántica entre dos ítems como su proximidad en una ontología dada, y la distancia entre ambos como una representación numérica que indica la lejanía entre ellos en un espacio geométrico.

Se pueden aplicar diversas técnicas de similaridad semántica basándonos en las categorías a las que pertenecen los ítems. Por ejemplo, en (Wang and Kong, 2007), se calcula la similaridad entre dos objetos t_i y t_j que comparten n_{ij} categorías, como el ratio de las categorías de la ontología compartidas:

$$SIM(t_i, t_j) = \frac{n_{ij}}{N} \quad (1)$$

donde N es el número total de categorías del dominio. A partir de cada ítem del vector de ítems interesantes para el usuario se le ofrecen los Top-K ítems más similares.

Existen otras métricas más complicadas que tienen en cuenta las distancias entre ítems, los pesos asignados, uso de métricas basadas en grafos dirigidos, etc. Pesquita y colaboradores (Pesquita et al., 2009) distinguen entre dos enfoques para el cálculo de similaridad en ontologías: basado en taxonomías (*edge based*) y basado en los nodos (*node based*).

El enfoque **basado en taxonomías** se basa principalmente en la estructura jerárquica de la ontología y la posición de los ítems dentro de ella.. La técnica más común de este enfoque es la *distancia* que selecciona el camino más corto entre dos términos de la ontología. En (Blanchard et al., 2005) y (Shenoy et al., 2012) se presenta un estudio de las diferentes enfoques usados para medir la distancia entre conceptos de una ontología o un tesoro basándose en su posición taxonómica. Sin embargo, medir la similaridad semántica entre dos conceptos considerando únicamente sus relaciones de subsunción no contempla las demás relaciones semánticas entre ellos, lo que conlleva a recomendaciones menos efectivas (Ganesan et al., 2003). Una de las formas de potenciar dichas métricas es el uso de pesos, es decir, ponderar de algún modo, los conceptos de la ontología, para usar posteriormente dichos pesos en las medidas de similitud entre los diversos ítems.

Existen diferentes formas de ponderar elementos de una ontología. En (Lord et al., 2003) se utilizan probabilidades de aparición de los conceptos de la ontología en un tesoro para posteriormente calcular la similaridad entre dos conceptos c_1 y c_2 como:

$$SIM(c_1, c_2) = -\ln p_{m_s}(c_1, c_2) \quad (2)$$

donde p_{m_s} es la probabilidad de la clase común menor que los subsume.

Un enfoque muy utilizado es la adición de pesos basados en la frecuencia de aparición de un término en un objeto (o documento) mediante la técnica de pesado TF-IDF (Term Frequency-Inverse Document Frequency) (Salton and Buckley, 1988), que determina la importancia de una característica i en un objeto j . La representación de los ítems se realiza mediante VSM (Vector Space Model) donde cada ítem es un vector de pesos. Para el cálculo de la importancia, TF-IDF emplea las siguientes ecuaciones:

Ecuación de frecuencia

$$TF_{ij} = \frac{f_{ij}}{\max_k N_{kj}} \quad (3)$$

donde f_{ij} es el número de veces que aparece i en j y $\max_k N_{kj}$ es el máximo ocurrencias de todas las características en j .

Ecuación de frecuencia inversa

$$IDf_i = \log \frac{N}{n_i} \quad (4)$$

donde N es el número de objetos del sistema y n_i es el número de objetos donde aparece la característica i .

Finalmente, el peso w_{ij} se obtiene como:

$$w_{ij} = TF_{ij} \cdot IDf_i \quad (5)$$

A partir de estos valores es posible representar un objeto j mediante un Vector de Pesos como:

$$w_j = (w_{1j}, w_{2j}, w_{3j}, \dots, w_{kj}) \quad (6)$$

donde cada w_{ij} es la importancia de la característica i para un objeto j .

Una vez obtenidos los vectores de pesos de los objetos del sistema, se puede calcular la similaridad semántica de un ítem del usuario u con un ítem i del sistema empleando la fórmula del coseno del ángulo:

$$Sim_{ju} = \cos_{ju} = \frac{\sum_{i=1}^k w_{ij} \cdot w_{iu}}{\sqrt{\sum_{i=1}^k w_{ij}^2} \cdot \sqrt{\sum_{i=1}^k w_{iu}^2}} \quad (7)$$

El enfoque **basado en nodos** se fundamenta en la comparación de las propiedades de los términos que intervienen. Un concepto muy usado en este enfoque es el de *Information Content* (IC), que proporciona una medida sobre cómo de específico e informativo es un término. El IC de un término c puede calcularse como el logaritmo negativo de la probabilidad:

$$IC(c) = -\log p(c) \quad (8)$$

donde $p(c)$ es la probabilidad de la ocurrencia de c en un corpus específico.

IC puede aplicarse a los ancestros comunes de un término para cuantificar la información que comparten y de ahí medir la similaridad. Esto puede hacerse de dos formas: mediante la técnica MICA (Resnik, 1995), donde solo se considera al ancestro de mayor IC, o la técnica DCA (Couto et al., 2005) donde se consideran todos los ancestros disjuntos, es decir, aquellos que no subsumen ningún otro ancestro común.

Otros autores, como (Sánchez et al., 2012), refinan esta clasificación y distinguen entre: basado en taxonomías (*edge-counting*), basado en características (*feature-based*) y basado en contenido de información (*IC-based*).

Una estrategia muy generalizada es el uso de **enfoques híbridos** como se presenta en las propuestas de (Schickel-Zuber and Faltings, 2007) o (Wang et al., 2007). En el capítulo 7, se aborda el análisis de diferentes estrategias de cálculo similaridad semántica para su aplicación en esta propuesta.

3.3. Ejemplos de SRS

SiteIF (Magnini and Strapparava, 2001) sigue la actividad del usuario cuando navega por su sitio Web de forma no intrusiva. El sistema aprende los intereses del usuario a partir de las páginas que visita. Dichas páginas se analizan para generar o modificar un modelo del mismo mediante documentos basados en “sentido”, en lugar de en las palabras. Utiliza fuentes semánticas externas (WordNet) para aportar el sentido al vocabulario. Los ítems de la web son noticias y cada una de ellas se asocia a un conjunto de *synsets*. El perfil de usuario se construye como una red semántica cuyos nodos representan los *synsets* que aparecen en las páginas visitadas. Para detectar los documentos (noticias) relevantes se hace *matching* entre la representación *synset* de un documento y el modelo del usuario mediante medidas de similaridad semántica en dicha red.

Quickstep (Middleton et al., 2004) es un SRS híbrido para la recomendación online de artículos académicos de investigación. Observa la navegación de los usuarios de forma no intrusiva para construir su perfil. El sistema utiliza una ontología de temas (*topics*) de artículos de investigación basada en el Open Directory Project de DMOZ⁴⁶. Se realizan anotaciones sobre los artículos asociándolos con clases de la ontología de temas empleando un clasificador de los *k* vecinos más cercanos. Los perfiles de los usuarios abarcan un subconjunto de *topics* con valores de interés sobre esos *topics*. Para realizar el *matching* entre los artículos y los perfiles se calcula una correlación entre los tres *topics* de mayor interés para el usuario y los artículos pertenecientes a ese *topic*.

Foafing-the-music (Celma and Serra, 2008) es un SRS basado en contenido. Recomienda contenidos musicales usando descripciones FOAF⁴⁷ (Friend of a Friend) para representar los perfiles de los usuarios. El dominio musical se describe mediante una ontología común en OWL-DL, y la información del contexto de los usuarios respecto a los contenidos musicales se extrae de diversos RSS recuperados a través de la web. FOAF utiliza perfiles de usuarios para la web semántica mediante páginas legibles por máquinas, y crea enlaces entre ellas y las cosas que los usuarios crean, hacen y en las que están interesados. Existen otros sistemas de recomendación que utilizan FOAF. Por ejemplo, FilmTrust (Golbeck, 2005), que recomienda películas a los usuarios basándose en redes sociales aumentadas con valores de confianza mediante los perfiles FOAF.

Sistemas más recientes utilizan ontologías para representar el contexto en entornos móviles y ubicuos. Este es el caso de SMARTMUSEUM (Ruotsalo et al., 2013), un sistema de recomendación para la Web de Datos enfocado al turismo donde se tiene en cuenta el contexto de los usuarios para ofrecer recomendaciones culturales “in-situ”. Por ejemplo, museos o

⁴⁶DMOZ Open Directory Project <http://www.dmoz.org>

⁴⁷FOAF. The Friend of a Friend Project: <http://www.foaf-project.org>

edificios de interés arquitectónico y objetos dentro de esos lugares, como esculturas u otras obras de arte, proporcionando descripciones y contenidos multimedia asociados a los objetos individuales. El sistema utiliza lenguajes de la Web Semántica como solución a los problemas de descripciones de contenido, datos de sensores y perfiles de usuario heterogéneos.

Referencias del capítulo

- ADOMAVICIUS, G. & TUZHILIN, A. 2005. Toward the next generation of recommender systems: A survey of the state-of-the-art and possible extensions. *IEEE Transactions on Knowledge and Data Engineering*, 17, 734-749.
- BLANCHARD, E., HARZALLAH, M., BRIAND, H. & KUNTZ, P. 2005. A Typology Of Ontology-Based Semantic Measures. *Proceedings of the Open Interop. Workshop on Enterprise Modelling and Ontologies for Interoperability*.
- BURKE, R. 2002. Hybrid Recommender Systems: Survey and Experiments. *User Modeling and User-Adapted Interaction*, 12, 331-370.
- CELMA, Ò. & SERRA, X. 2008. FOAFing the music: Bridging the semantic gap in music recommendation. *Web Semantics: Science, Services and Agents on the World Wide Web*, 6, 250-256.
- COUTO, F. M., SILVA, M. J. & COUTINHO, P. M. Semantic similarity over the gene ontology: family correlation and selecting disjunctive ancestors. *Proceedings of the 14th ACM international conference on Information and knowledge management 2005*. ACM, 343-344.
- DEGEMMIS, M., LOPS, P. & SEMERARO, G. 2007. A content-collaborative recommender that exploits WordNet-based user profiles for neighborhood formation. *User Modeling and User-Adapted Interaction*, 17, 217-255.
- GANESAN, P., GARCIA-MOLINA, H. & WIDOM, J. 2003. Exploiting hierarchical domain structure to compute similarity. *ACM Transactions on Information Systems (TOIS)*, 21, 64-93.
- GAUCH, S., SPERETTA, M., CHANDRAMOULI, A. & MICARELLI, A. 2007. User profiles for personalized information access. *The adaptive web*. Springer Berlin Heidelberg.
- GOLBECK, J. Semantic web interaction through trust network recommender systems. *Proceedings of the ISWC 2005 Workshop on End User Semantic Web Interaction, 2005 Galway, Ireland*.
- GRUBER, T. R. 1995. Toward principles for the design of ontologies used for knowledge sharing? *International journal of human-computer studies*, 43, 907-928.
- HECKMANN, D. & KRUEGER, A. 2003. A user modeling markup language (UserML) for ubiquitous computing. *User Modeling*. Springer Berlin Heidelberg.
- HECKMANN, D., SCHWARTZ, T., BRANDHERM, B., SCHMITZ, M. & VON WILAMOWITZ-MOELLENDORFF, M. 2005. Gumo—the general user model ontology. *User Modelling*. Springer Berlin Heidelberg.
- LEE, W. N., SHAH, N., SUNDLASS, K. & MUSEN, M. Comparison of ontology-based semantic-similarity measures. *AMIA annual symposium proceedings, 2008*. American Medical Informatics Association, 384.
- LOIZOU, A. & DASMAHAPATRA, S. Recommender systems for the semantic web. *Proceedings of the ECAI 2006, 2006*. 76-81.
- LOPS, P., GEMMIS, M. D. & SEMERARO, G. 2011. Content-based recommender systems: State of the art and trends. *Recommender systems handbook*. Springer US.
- LORD, P. W., STEVENS, R. D., BRASS, A. & GOBLE, C. A. 2003. Investigating semantic similarity measures across the Gene Ontology: the relationship between sequence and annotation. *Bioinformatics*, 19, 1275-1283.

- MAGNINI, B. & STRAPPARAVA, C. 2001. Improving user modelling with content-based techniques. *User Modeling*. Springer Berlin Heidelberg.
- MCSHERRY, D. 2005. Explanation in Recommender Systems. *Artificial Intelligence Review*, 24, 179-197.
- MIDDLETON, S. E., SHADBOLT, N. R. & DE ROURE, D. C. 2004. Ontological user profiling in recommender systems. *ACM Transactions on Information Systems (TOIS)*, 22, 54-88.
- NIEDRITIS, A., NIEDRITE, L. & KOZMINA, N. Integration Architecture of User Models. Perspectives in Business Informatics Research Local Proceedings of 10th International Conference BIR2011 Associated Workshops and Doctoral Consortium, 2011 Riga Technical University. 323-330.
- NILES, I. & PEASE, A. Towards a standard upper ontology. Proceedings of the international conference on Formal Ontology in Information Systems, 2001. ACM, 2-9.
- PESQUITA, C., FARIA, D., FALCAO, A. O., LORD, P. & COUTO, F. M. 2009. Semantic similarity in biomedical ontologies. *PLoS Computational Biology*, 5.
- RAZMERITA, L., ANGEHRN, A. & MAEDCHE, A. 2003. Ontology-based user modeling for knowledge management systems. *User Modeling*. Springer Berlin Heidelberg.
- RESNIK, P. Using information content to evaluate semantic similarity in a taxonomy. Proceedings of the 14th International Joint Conference on Artificial Intelligence, 1995. 448-453.
- RUOTSALO, T., HAAV, K., STOYANOV, A., ROCHE, S., FANI, E., DELIAI, R. & HYVÖNEN, E. 2013. SMARTMUSEUM: a mobile recommender system for the web of data. *Web semantics: Science, services and agents on the world wide web*, 20, 50-67.
- SALTON, G. & BUCKLEY, C. 1988. Term weighting approaches in automatic retrieval. *Information Processing and Management*, 24, 513-523.
- SÁNCHEZ, D., BATET, M., ISERN, D. & VALLS, A. 2012. Ontology-based semantic similarity A new feature-based approach. *Expert Systems with Applications*, 39, 7718-7728.
- SCHICKEL-ZUBER, V. & FALTINGS, B. 2007. OSS: A Semantic Similarity Function based on Hierarchical Ontologies. *IJCAI* 7, 551-556.
- SHENOY, K. M., SHET, K. C. & ACHARYA, U. D. 2012. A new similarity measure for taxonomy based on edge counting. *International Journal of Web & Semantic Technology*, 3, 23-30.
- SIEG, A., MOBASHER, B. & BURKE, R. D. 2007. Learning Ontology-Based User Profiles: A Semantic Approach to Personalized Web Search. *IEEE Intelligent Informatics Bulletin*, 8, 7-18.
- THAKKER, D., DIMITROVA, V., LAU, L., DENAUX, R., KARANASIOS, S. & YANG-TURNER, F. A priori ontology modularisation in ill-defined domains. Proceedings of the 7th International Conference on Semantic Systems, 2011. ACM, 167-170.
- TRAJKOVA, J. & GAUCH, S. 2004. Improving Ontology-Based User Profiles. *RIAO, 2004*.
- WAHLSTER, W. & KOBASA, A. 1989. *User models in dialog systems* Springer Berlin Heidelberg.
- WANG, J. Z., DU, Z., PAYATTAKOOL, R., PHILIP, S. Y. & CHEN, C. F. 2007. A new method to measure the semantic similarity of GO terms. *Bioinformatics*, 23, 1274-1281.
- WANG, R. Q. & KONG, F. S. 2007. Semantic-enhanced personalized recommender system. *International Conference on Machine Learning and Cybernetics*. IEEE.
- ZIEGLER, C. N., LAUSEN, G. & SCHMIDT-THIEME, L. Taxonomy-driven computation of product recommendations. Proceedings of the thirteenth ACM international conference on Information and knowledge management, 2004. ACM, 406-415.

CAPÍTULO 4

SISTEMAS DE RECOMENDACIÓN EN EL DOMINIO NUTRICIONAL

Este capítulo está centrado en los Sistemas de Recomendación en entornos nutricionales. En él se indican las motivaciones principales para su desarrollo, y se realiza una revisión acerca de los distintos desafíos que surgen. Igualmente, se analizan algunas fuentes de información nutricional para su explotación en un sistema de estas características. A continuación, se lleva a cabo un estudio del trabajo relacionado con los elementos clave de dichos sistemas como son las estrategias de recomendación, el modelado del usuario, la monitorización y el aprendizaje de preferencias alimenticias. El final del capítulo presenta un estudio de Sistemas de Recomendación existentes en el dominio de nutrición: cómo abordan los desafíos mencionados y qué enfoques utilizan para la representación e implementación de los elementos clave reseñados previamente.

4. 1. Introducción

Existe mucha información dietética y nutricional dispersa en la red que se puede encontrar en forma de foros, revistas, páginas de información nutricional, de consejos alimenticios, sitios de recetas, etc. Este amplio abanico de posibilidades, que en principio es positivo, conlleva ciertos problemas para el usuario. En primer lugar, cuando las personas navegan por la red, necesitan confiar en los contenidos que encuentran y conocer a partir de qué fuentes se han elaborado, para decidir si deben o no tomarlos como referencia confiable. Por otro lado, la cantidad de información es tan amplia y dispersa, que el usuario puede verse confuso o saturado y no ser capaz de asimilarla, sintetizarla y ponerla en común para lograr sus objetivos.

Una solución a este problema es el uso de Sistemas de Recomendación (SR) Nutricionales que, apoyados en conocimiento nutricional o dietético experto, proporcionan recomendaciones personalizadas a los usuarios que las requieran. La importancia de las recomendaciones individualizadas en el ámbito de la salud y la nutrición se ha recogido en diversos estudios (Brug et al., 1996), (Lawrence et al., 2001), (Hawkins et al., 2008), (Wiesner and Pfeifer, 2010), que demuestran que la personalización de las recomendaciones es fundamental para fomentar el cumplimiento de las mismas por parte de los usuarios.

4. 2. Desafíos en las recomendaciones nutricionales

En la construcción de SR nutricionales, existen ciertos desafíos a afrontar. Éstos se refieren tanto a la forma en que se desarrolla el sistema y las técnicas empleadas, como a la interacción con el usuario (Mika, 2011). Los retos más destacados relacionados con la conexión del usuario al sistema son los siguientes:

- **¿Qué datos sobre el usuario hay que obtener para ofrecerle recomendaciones personalizadas?**

Es necesario conocer qué tipo de usuario utilizará el sistema y en función de dicho tipo, fijar aquellas características sobre el mismo que se deberán tener en cuenta en el proceso de recomendación.

- **¿De qué modo se van a recoger los intereses de los usuarios?**
- Dependiendo del tipo de recomendaciones a ofrecer, los intereses pueden obtenerse de varias formas tanto explícita como implícitamente. De forma explícita, se consiguen mediante formularios o preguntas adecuadas al perfil del usuario que comienza a usar el sistema, y en el resto de conexiones el usuario puede aportar valoraciones sobre las recomendaciones ofertadas mediante *ratings*. Ejemplos de *ratings* explícitos son: puntuar de una a cinco estrellas o indicar “me gusta / no me gusta” sobre una recomendación. De forma implícita, pueden recogerse estudiando las selecciones que han realizado sobre las recomendaciones que se le han ofrecido.
- **¿Cómo se van a conocer los alimentos que han tomado últimamente o las dietas que han seguido?**
- Si los usuarios siguen las recomendaciones ofertadas y lo indican al sistema, éste podrá tener control de los alimentos tomados desde el comienzo de su uso; en el caso que el usuario no siguiera todas las recomendaciones, habría que emplear métodos para mantener al sistema informado sobre los alimentos ingeridos en su lugar.
- **¿De qué forma se puede tener información sobre los efectos del uso del sistema en el usuario?**
- Una de las formas más típicas es mostrando temporalmente formularios que sean capaces de recoger, tanto los cambios físicos, como los cambios en los hábitos alimenticios, si éstos se han producido.

Algunos de estos desafíos se enfrentan al problema conocido como incumplimiento del usuario (*user non-compliance*), que se produce cuando el usuario no sigue las instrucciones o las sigue de forma incorrecta, lo que representa uno de los mayores obstáculos en todos los sistemas relacionados con la salud (McNee et al., 2003).

En el caso de los SR en general, no es fácil conseguir que el usuario valore todos los ítems, por olvido, porque no le apetece, porque tiene prisa, etc. Centrándonos en la nutrición, ocurre lo mismo cuando se requiere que el usuario introduzca qué alimentos ha ingerido para reemplazar una recomendación que no ha seguido y cuáles han sido los motivos, o cuando es necesario que rellene formularios para valorar el efecto del uso del sistema.

Los desafíos a abordar por lo que se refiere a las técnicas empleadas para la construcción del sistema son los mismos que para cualquier Sistema de Recomendación, aunque una de las decisiones más importantes es la relativa a la estrategia de recomendación a realizar: utilizando como ítems, los diferentes alimentos, las recetas o las dietas. Algunos de los métodos propuestos en la literatura usan estrategias híbridas, por ejemplo, recomendar recetas en función de la valoración del usuario sobre sus ingredientes (Freyne et al., 2011). Otro de los grandes retos está relacionado con la obtención de la información necesaria para la construcción de la base de conocimiento y la forma de plasmar esta información en el sistema de modo que el proceso esté siempre guiado por un experto en el aspecto nutricional. En el apartado siguiente se citan algunas fuentes en línea de datos nutricionales confiables.

4. 3. Fuentes de información nutricional

Para la construcción de la base de conocimiento de un sistema de recomendación nutricional es útil partir de fuentes que recojan datos nutricionales que puedan ser reutilizados en otros

sistemas. Una medida de fiabilidad de los datos es que dichos datos estén avalados por alguna institución reconocida. Algunas de esas fuentes que se encuentran en la red son las siguientes:

- Área web de Nutrición de la Organización Mundial de la Salud⁴⁸ (OMS). En esta página se pueden encontrar base de datos nutricionales, y diversos estudios y publicaciones sobre nutrición mundial. La gran mayoría de ellas solamente en inglés.
- Base de datos de nutrientes del Departamento de Agricultura de los Estados Unidos⁴⁹ (USDA). Contiene información acerca de los nutrientes de casi 8.000 alimentos preparados, o crudos. La base de datos está disponible, tanto para descarga en formato Access o ASCII, como para realizar consultas online. Sólo se encuentra en inglés.
- Ontología PIPS Food. Es una ontología, en inglés, que proporciona a los usuarios información nutricional sobre los alimentos y algunas recomendaciones nutricionales. Está desarrollada dentro del 6º Programa Marco de la Unión Europea en el ámbito de Salud electrónica.
- Programa Dial⁵⁰. Es un programa para valoración de dietas y cálculo de alimentación realizado por profesores del Departamento de Nutrición y Bromatología I de la Facultad de Farmacia de la Universidad Complutense de Madrid. El programa y su base de datos es descargable previo pago.
- Base de Datos Española de Composición de Alimentos⁵¹ (BEDCA). Está publicada por la Red BEDCA del Ministerio de Ciencia e Innovación y bajo la coordinación y financiación de la Agencia Española de Seguridad Alimentaria y Nutrición del Ministerio de Sanidad y Política Social. Los valores de composición de alimentos recogidos en esta base de datos se han obtenido de distintas fuentes –que incluyen laboratorios, industria alimentaria y publicaciones científicas–, o han sido calculados. Está accesible para su consulta en español e inglés pero no para descarga.
- Tesauro AGROVOC de la FAO (Caracciolo et al., 2013). Contiene más de 40.000 conceptos en hasta 21 idiomas relacionados con alimentación, nutrición, agricultura y pesca. Está accesible para su descarga en diversos formatos, y para su consulta.

4. 4. Trabajo Relacionado

4.4.1. Perfil Nutricional del Usuario

Los primeros SR de alimentos representaban a los usuarios según sus preferencias o consultas al sistema. Los perfiles se definían mediante vectores en el espacio de características o atributos de los ítems, representando el interés del usuario en el rango de atributos definido (Svensson et al., 2000), (Lawrence et al., 2001). Estos intereses permitían agrupar a los usuarios en diferentes *clusters* según sus preferencias. Más adelante empezó a considerarse la inclusión de las necesidades relacionadas con la salud del usuario, por lo que los perfiles debían incluir aquellas características físicas o fisiológicas de éstos que debían regir las recomendaciones. Algunos sistemas almacenan dichas características mediante bases de datos (Sivilai et al., 2012) y otros utilizan tecnología de la Web Semántica como perfiles FOAF extendidos u ontologías.

⁴⁸ Área de Nutrición de la OMS. <http://www.who.int/nutrition/en/>

⁴⁹ Base de datos nutricional USDA. <http://www.usda.gov/wps/portal/usda/usdahome>

⁵⁰ Programa Dial. <http://www.alceingenieria.net/nutricion.htm>

⁵¹ Base de datos BEDCA. <http://www.bedca.net/bdpub/index.php>

El uso de ontologías para la representación de perfiles de usuario empezó a tomar gran relevancia debido a la semántica aportada y posibilidad de razonamiento y reutilización. Algunos autores crean su propia ontología (Fudholi et al., 2009) y (Suksom et al., 2010), y otros reutilizan alguna ontología de usuario en entornos médicos, como (Su et al., 2013). Otros autores (Al-Nazer et al., 2014) persiguen la creación de un modelo ontológico de usuario para su estandarización en entornos nutricionales.

4.4.2. Estrategias de Recomendación

Las recomendaciones finales en los SR nutricionales son:

- **Recomendación de dietas.** Este tipo de recomendaciones están generalmente ligadas a la oferta de planes dietéticos saludables según los requisitos nutricionales de los usuarios. Por ejemplo, una *dieta calórica semanal de verano sin gluten*. Las dos estrategias principales para la confección de dichas dietas son: la forma estática, en la cual se recomienda una dieta previamente establecida o, la forma dinámica, en la que la dieta se construye desde cero algorítmicamente. Sin embargo, si la recomendación se queda simplemente en la dieta, suele resultar un plan demasiado estricto para los usuarios. Es posible que, por disponibilidad, apetencia o cualquier otro motivo, el usuario desee modificar en un determinado momento ese plan. Por esto, es preferible que la dieta sea capaz de ofrecer distintas opciones a los platos contenidos en la dieta inicial a petición del usuario, para lo cual se usa el tipo de recomendación dado a continuación.
- **Recomendación de recetas o platos.** Se ofrecen recetas o platos a los usuarios según preferencias o necesidades. Generalmente, por sí solo no contempla las ingestas previas para proporcionar las recomendaciones, a diferencia del plan dietético. Sin embargo, este tipo de recomendación ha sido quizás el más estudiado por los investigadores. En trabajos previos se distingue entre dos estrategias principales: orientadas a la receta u orientadas al ingrediente.
 - Las recomendaciones orientadas a la receta no contemplan, en principio, qué alimentos contiene una receta para ofrecerla. Suele basarse en el paradigma colaborativo de recomendación, ofertando aquellas con mejor valoración por usuarios con gustos similares o, en el paradigma basado en contenido, donde se ofrecen aquellas recetas similares a las que el usuario valoró positivamente en el pasado. En algunos casos se ofrecen platos o recetas conteniendo algún alimento solicitado por el usuario. Por ejemplo, *recetas con cordero*; algunas características nutricionales, como *plato bajo en calorías*; o algún otro factor como *recetas de dificultad baja*. Por lo tanto, en esta estrategia se utilizan métricas de similaridad entre recetas utilizando como características las anotaciones de las recetas.
 - La estrategia orientada al ingrediente utiliza métricas de similaridad entre los ingredientes, aunque, en general, siguen recomendando platos o recetas. Estudios previos demuestran que esta estrategia de recomendación proporciona recomendaciones alimenticias más precisas (Freyne and Berkovsky, 2010). Dentro de esta estrategia existen dos técnicas diferenciadas: obtener aquellos ingredientes más valorados para ofrecer platos que los contengan o contemplar las propiedades nutricionales de los alimentos en los métodos de selección de características para los cálculos de similaridad.

4.4.3. Monitorización

En los SR nutricionales relacionados con la salud debe ofrecerse algún tipo de monitorización para conocer el comportamiento del usuario dentro del sistema y los efectos que su uso conlleva sobre su salud (Al-Nazer et al., 2014). De este modo, puede deducirse su grado de implicación al seguir las recomendaciones, sus preferencias y hábitos alimenticios, el impacto en su estado nutricional o la posible aparición de determinadas carencias o irregularidades (Keller et al., 2015). Toda esta información puede ser accesible, tanto por el cuidador o nutricionista, como por el propio usuario. En el caso del cuidador, éste puede solicitar informes de uso o de ingestas donde se permitan diferentes cálculos o estadísticas de consumo nutricional. Además, el sistema puede ser capaz de detectar irregularidades y enviar una notificación al cuidador y al propio usuario (De Pessemier et al., 2013).

La técnica más utilizada para conocer los cambios en el estado nutricional del usuario es el denominado *Screening nutricional*. Su propósito es predecir el riesgo de malnutrición del paciente y si el tratamiento nutricional seguido le está influenciando positiva o negativamente (Kondrup et al., 2003). Los autores de (Skipper et al., 2012) definen las siguientes consideraciones clave de cualquier herramienta de *Screening* nutricional:

- Debe ser rápida, fácil de usar y confiable para los pacientes o usuarios.
- La herramienta y sus parámetros deben ser definidos por dietistas expertos.
- El *Screening* debe ocurrir en periodos de tiempo bien definidos según necesidad.

Los mismos autores realizan un estudio de las diferentes herramientas, recogiendo entre las más importantes: MNA (Mini Nutritional Assesment) (Guigoz et al., 1997), MUST (Malnutrition Universal Screening Tool) (Stratton et al., 2004), DETERMINE checklist (Vailas et al., 1998) o NRI (Nutritional Risk Index) (Bouillanne et al., 2005).

4.4.4. Análisis de preferencias y personalización de las recomendaciones

Existen dos modos de interacción básicos, posiblemente complementarios, para conocer las preferencias de los usuarios:

- Explícitamente (*explicit feedback*): el usuario al registrarse en el sistema selecciona manualmente qué alimentos no le gustan (igualmente aquellos a los que sea alergias o presente contraindicaciones). Algunos sistemas también permiten al usuario introducir sus alimentos favoritos. Además, el usuario puede valorar mediante *ratings* las recomendaciones recibidas.
- Implícitamente (*implicit feedback*): mediante el análisis del seguimiento de las recomendaciones. En este caso, se suelen emplear técnicas de *machine learning* o *automated inference* para el aprendizaje de las preferencias. Este modo es transparente al usuario, por lo que se considera *no intrusivo*.

La mayoría de los sistemas que van más allá de un simple cuestionario inicial para conocer qué alimentos, a priori, agradan o no a los usuarios, o cuáles no puede tomar, utilizan la técnica de *ratings* sobre alimentos o dietas. A partir de ahí, calculan predicciones sobre el resto de las recetas, es decir, predicen el *rating* de las recetas aún no valoradas y ofrecen las de mayor *rating* inferido (Freyne and Berkovsky, 2010) y (Harvey et al., 2013). Algunos autores distinguen entre predicciones explícitas e implícitas dependiendo de si se obtienen mediante *feedback* explícito o implícito, combinando incluso ambas predicciones en la generación de recomendaciones (De Pessemier et al., 2013).

Algunas estrategias empleadas en la predicción de *ratings* se basan en la inferencia de *ratings* sobre ingredientes y otras en las recetas. A partir de ahí se emplean técnicas ya conocidas, como la media aritmética ponderada, coeficiente de Pearson o coeficiente del coseno de las similitudes junto con los ratings. Un ejemplo de aplicación de estas técnicas viene recogido en (Freyne et al., 2011) donde se realiza un análisis de resultados de las predicciones partiendo de diferentes paradigmas y estrategias de recomendación:

- En el paradigma colaborativo aplica la siguiente expresión para el cálculo de la predicción de la receta r_t y el usuario u_a :

$$pred(u_a, r_t) = \frac{\sum_n^N sim(u_a, u_n) rat(u_a, r_t)}{\sum_n^N sim(u_a, u_n)} \quad (9)$$

donde N es el número de usuarios vecinos de y y $sim(u_a, u_b)$ la similitud entre dos usuarios, a y b , obtenida mediante el coeficiente de correlación de Pearson.

- En el paradigma basado en contenido utiliza la siguiente expresión:

$$pred(u_a, r_t) = \frac{\sum_j^{r_t} score(u_a, ingr_j)}{j} \quad (10)$$

- donde j es el total de ingredientes de la receta y $score(u_a, ingr_j)$ son los ratings proporcionados por el usuario u_a a cada ingrediente.

4. 5. Otros Sistemas de Recomendación Nutricionales

A continuación, se analizan algunas de las propuestas que se encuentran en la literatura, centrándonos en las fuentes de conocimiento nutricional, las estrategias de recomendación, el perfil nutricional de usuario, la gestión de sus preferencias y el seguimiento de su estado nutricional. La mayoría de los trabajos relacionados con sistemas de recomendación nutricional son relativamente recientes. Los primeros sistemas se centraban más en las preferencias de los usuarios (Lawrence et al., 2001) y (Sobecki et al., 2006); en los últimos años se han hecho considerables esfuerzos en la inclusión de las necesidades relacionadas con la salud (Freyne and Berkovsky, 2010), (Suksom et al., 2010) y (El-Dosuky et al., 2012).

En (Arwan et al., 2013), los autores proponen un SR de alimentos para personas con diabetes de tipo 2 (o no-insulino dependientes). La base de conocimiento está formada por una ontología de menús donde una de las ramas clasifica las necesidades calóricas de los pacientes (*calories*) y otra rama clasifica los alimentos. El perfil de usuario viene modelado mediante una ontología que recoge algunas de sus características físicas, como el índice de masa corporal, y cuya rama *calories* hará *matching* con la rama de igual nombre en la ontología de menús. Utiliza reglas SWRL para el soporte a la clasificación de alimentos y a la recuperación de conceptos junto a SPARQL. La medida de similitud entre alimentos está basada en árboles con pesos (*weighted-trees*). El sistema no considera las preferencias de los usuarios ni el historial de consumo previo.

FoodManager (Iglesias et al., 2010) es un sistema de soporte a las personas mayores en actividades de alimentación y cocina. Uno de sus pilares es la recomendación de alimentos con

menús semanales basados en patologías, condiciones de salud y preferencias de los usuarios. Los usuarios pueden reemplazar una receta por otra para adaptarse a sus preferencias. Esta receta se elige de una lista obtenida de la base de datos de recetas sin tener en cuenta los alimentos ingeridos anteriormente o los nutrientes de las alternativas ofrecidas. Los usuarios indican sus preferencias señalando qué alimentos les gustan o no les gustan de una lista de ingredientes. El estado nutricional del usuario no parece ser considerado en el funcionamiento del sistema y no hay ningún tipo de monitorización de las ingestas de usuario o cambios en su estado nutricional. Aunque la interfaz de usuario ha sido diseñada teniendo en cuenta las restricciones de edad avanzada y se puede visualizar a través de un ordenador personal, el sistema requiere un electrodoméstico especial y sensores para su funcionamiento. Además, su elevado coste hace que sea económicamente inviable para la mayoría de la población de edad avanzada.

Aberg (Aberg, 2006) propone un sistema inteligente de planificación de alimentos para que las personas mayores lo puedan utilizar desde casa. El sistema considera como variables importantes en las recomendaciones el gusto, el coste, la dificultad de preparación, la diversidad, las restricciones y necesidades alimenticias, las propiedades nutricionales y los alimentos disponibles. Es un sistema híbrido: utiliza la técnica de filtrado colaborativo para hacer predicciones sobre lo que podría gustarle una receta (que aún no haya valorado) a un usuario, basándose en sus *ratings* pasados y en los de otros usuarios. Utilizan un marcado XML de recetas para representar la información de contenido almacenada en las recetas de la base de datos. Está enfocado a construir planes óptimos de alimentos teniendo en cuenta las variables consideradas anteriormente y el grado de satisfacción del usuario.

En su trabajo, han experimentado con dos formas de modelar el sistema: un enfoque *basado en parámetros*, donde se usan variables como parámetros de una receta (tiempo, coste, energía, proteínas, etc.) y los valores se basan en los existentes en la base de datos de recetas; y otro enfoque más sencillo, *basado en recetas*, donde se utiliza una única variable por alimento de la receta. Para ambos modelos, emplean un conjunto de restricciones sobre las preferencias de los usuarios, considerando gustos y alergias. El sistema no identifica las necesidades nutricionales. Éstas deben ser introducidas manualmente con anterioridad por los cuidadores (como pueden ser intervalos requeridos de grasa o colesterol). Parte de 50 recetas reales y, a partir de éstas, se generan otras similares automáticamente. Se permite al usuario sustituir una receta por otra contenida en una lista top-5 de similaridad. Las técnicas empleadas en la recomendación son algoritmos basados en ítem con similaridad del coseno ajustada mediante pesos en las predicciones. El perfil del usuario no cubre tampoco información alguna sobre el estado nutricional y el sistema no proporciona una forma de seguimiento de ingestas previas y cambios en el estado nutricional.

Los autores en (Sivilai et al., 2012) amplían y ajustan FOODS, para crear un sistema de menús de recomendación (Snae and Bruckner, 2008) que proporcione planes de dieta a personas mayores en entornos hospitalarios. Esta propuesta consiste en un sistema de recomendación en el que el perfil de usuario se basa en un registro de salud médica derivada del diagnóstico de un médico que contiene también el estado nutricional y qué alimentos le gustan o no. Estas preferencias se indican de forma explícita por el usuario y el plan de dieta recomendada no se puede personalizar más allá. La base de conocimiento contiene la ontología de alimentación y nutrición PIPS, y un conjunto de reglas para inferir las comidas recomendadas a partir de una base de datos de alimentos. A partir de ahí se construye un plan de dieta considerando las

necesidades de los usuarios y sus preferencias. Una vez más, no hay ninguna tarea de supervisión incluida en el sistema y las ingestas anteriores no se tienen en cuenta.

Este análisis desvela una grave falta de sistemas capaces de ayudar a los usuarios a construir un plan de dieta considerando conjuntamente sus necesidades nutricionales y preferencias, donde dichas preferencias sean más flexibles (no solamente le gusta / no le gusta), pudiendo también inferirse del comportamiento de los usuarios en el sistema. La personalización de las dietas también es un factor crucial para el ajuste de las mismas a las preferencias del usuario, motivando a los usuarios para utilizar el sistema. Por último, un gran inconveniente que se encuentra en los sistemas analizados es la ausencia de un componente de monitorización que permita la detección y notificación de los cambios en el estado nutricional del usuario o deficiencias en las ingestas, factores críticos a considerar en el estado de salud de usuarios con necesidades alimenticias más delicadas, como son los ancianos.

Referencias del capítulo

- ABERG, J. 2006. Dealing with Malnutrition: A Meal Planning System for Elderly. *AAAI Spring Symposium: Argumentation for Consumers of Healthcare*.
- AL-NAZER, A., HELMY, T. & AL-MULHEM, M. 2014. User's Profile Ontology-based Semantic Framework for Personalized Food and Nutrition Recommendation. *Procedia Computer Science*, 32, 101-108.
- ARWAN, A., PRIYAMBADHA, B., SARNO, R., SIDIQ, M. & KRISTIANTO, H. 2013. Ontology and semantic matching for diabetic food recommendations. *Information Technology and Electrical Engineering (ICITEE)*. IEEE.
- BOUILLANNE, O., MORINEAU, G., DUPONT, C., COULOMBEL, I., VINCENT, J. P., NICOLIS, I. & AUSSEL, C. 2005. Geriatric Nutritional Risk Index: a new index for evaluating at-risk elderly medical patients. *The American journal of clinical nutrition*, 82, 777-783.
- BRUG, J., STEENHUIS, I., ASSEMA, P. V. & VRIES, H. D. 1996. The Impact of a Computer-Tailored Nutrition Intervention. *Preventive Medicine*, 25, 236-242.
- CARACCILO, C., STELLATO, A., MORSHED, A., JOHANNSEN, G., RAJBHANDARI, S., JAQUES, Y. & KEIZE, J. 2013. The Agrovoc linked dataset. *Semantic Web*, 4, 341-348.
- DE PESSEMIER, T., DOOMS, S. & MARTENS, L. A food recommender for patients in a care facility. Proceedings of the 7th ACM conference on Recommender systems, 2013. ACM, 209-212.
- EL-DOSUKY, M. A., RASHAD, M. Z., HAMZA, T. T. & EL-BASSIOUNY, A. H. 2012. Food Recommendation Using Ontology and Heuristics. *Communications in Computer and Information Science*, 322, 423-429.
- FREYNE, J. & BERKOVSKY, S. Intelligent food planning: personalized recipe recommendation. Proceedings of the 15th international conference on Intelligent user interfaces, 2010. ACM, 321-324.
- FREYNE, J., BERKOVSKY, S. & SMITH, G. 2011. Recipe recommendation: Accuracy and reasoning. *User Modeling, Adaption and Personalization*. Springer Berlin Heidelberg.
- FUDHOLI, D. H., MANEERAT, N. & VARAKULSIRIPUNTH, R. Ontology-based daily menu assistance system. 6th International Conference on Electrical Engineering/Electronics, Computer, Telecommunications and Information Technology, ECTI-CON 2009, 2009. IEEE, 694-697.
- GUIGOZ, Y., VELLAS, B. & GARRY, P. J. 1997. Mini Nutritional Assessment: a practical assessment tool for grading the nutritional state of elderly patients. *The mini nutritional assessment: MNA. Nutrition in the elderly*.

- HARVEY, M., LUDWIG, B. & ELSWEILER, D. You are what you eat: Learning user tastes for rating prediction. *String Processing and Information Retrieval 2013*. Springer International Publishing, 153-164.
- HAWKINS, R. P., KREUTER, M., RESNICOW, K., FISHBEIN, M. & DIJKSTRA, A. 2008. Understanding tailoring in communicating about health. *Health education research*, 23, 454-466.
- IGLESIAS, R., IBARGUREN, I., DE SEGURA, N. G., UGALDE, J., COELLO, L. & ITURBURU, M. FoodManager: a cooking, eating and appliance controlling support system for the elderly. *Proceedings of the 3rd International Conference on Pervasive Technologies Related to Assistive Environments*, 2010. ACM.
- KELLER, H. H., MCCULLOUGH, J., DAVIDSON, B., VESNAVER, E., LAPORTE, M., GRAMLICH, L. & JEEJEEBHOY, K. 2015. The Integrated Nutrition Pathway for Acute Care (INPAC): Building consensus with a modified Delphi. *Nutrition journal*, 14, 1.
- KONDRUP, J. E. S. P. E. N., ALLISON, S. P., ELIA, M., VELLAS, B. & PLAUTH, M. 2003. ESPEN guidelines for nutrition screening 2002. *Clinical nutrition*, 22, 415-421.
- LAWRENCE, R. D., ALMASI, G. S., KOTLYAR, V., VIVEROS, M. & DURI, S. S. 2001. Personalization of supermarket product recommendations. *Data Mining and Knowledge Discovery*, 5, 11-32.
- MCNEE, S. M., LAM, S. K., KONSTAN, J. A. & RIEDL, J. Interfaces for eliciting new user preferences in recommender systems. *User Modeling*, 2003. Springer Berlin Heidelberg, 178-187.
- MIKA, S. 2011. Challenges for Nutrition Recommender Systems. *Proceedings of the 2nd Workshop on Context Aware Intel. Assistance*. Berlin, Germany.
- SIVILAI, S., SNAE, C. & BRUECKNER, M. 2012. Ontology-Driven Personalized Food and Nutrition Planning System for the Elderly. *ICB MIS*.
- SKIPPER, A., FERGUSON, M., THOMPSON, K., CASTELLANOS, V. H. & PORCARI, J. 2012. Nutrition Screening Tools An Analysis of the Evidence. *Journal of Parenteral and Enteral Nutrition*, 36, 292-298.
- SNAE, C. & BRUCKNER, M. FOODS: a food-oriented ontology-driven system. 2nd IEEE International Conference on Digital Ecosystems and Technologies, 2008. IEEE, 168-176.
- SOBECKI, J., BABIAK, E. & ŚLANINA, M. 2006. Application of hybrid recommendation in web-based cooking assistant. *Knowledge-Based Intelligent Information and Engineering Systems*. Heidelberg: Springer Berlin.
- STRATTON, R. J., HACKSTON, A., LONGMORE, D., DIXON, R., PRICE, S., STROUD, M. & ELIA, M. 2004. Malnutrition in hospital outpatients and inpatients: prevalence, concurrent validity and ease of use of the 'malnutrition universal screening tool' ('MUST') for adults. *British Journal of Nutrition*, 92, 799-808.
- SU, C. J., CHEN, Y. A. & CHIH, C. W. 2013. Personalized ubiquitous diet plan service based on ontology and web services. *International Journal of Information and Education Technology*, 5, 522-528.
- SUKSOM, N., BURANARACH, M., THEIN, Y. M., SUPNITHI, T. & NETISOPAKUL, P. A Knowledge-based Framework for Development of Personalized Food Recommender System. *Proceedings of the 5th Int. Conf. on Knowledge, Information and Creativity Support Systems*, 2010.
- SVENSSON, M., LAAKSOLAHTI, J., HÖÖK, K. & WAERN, A. 2000. A recipe based on-line food store. *Proceedings of the 5th international conference on Intelligent user interfaces*. ACM.
- VAILAS, L. I., NITZKE, S. A., BECKER, M. & GAST, J. 1998. Risk indicators for malnutrition are associated inversely with quality of life for participants in meal programs for older adults. *Journal of the American Dietetic Association*, 98, 548-553.

WIESNER, M. & PFEIFER, D. Adapting recommender systems to the requirements of personal health record systems. Proceedings of the 1st ACM International Health Informatics Symposium, 2010. ACM, 410-414.

CAPÍTULO 5

NutElCare: SISTEMA DE RECOMENDACIÓN SEMÁNTICO NUTRICIONAL

Este capítulo presenta una introducción a un Sistema de Recomendación Semántico Nutricional, que hemos desarrollado en el marco de esta tesis doctoral y que hemos denominado NutElCare (Nutrition for Elder Care). En primer lugar se explica su funcionamiento básico y posteriormente mostramos cuáles son los componentes que conforman su arquitectura, cómo se han desarrollado y cómo interactúan entre sí para proporcionar recomendaciones nutricionales a personas mayores de forma satisfactoria. Para finalizar, se explican los principios de usabilidad y accesibilidad que se han seguido para el desarrollo del sistema y adecuarlo al uso de personas mayores.

5.1. Introducción

El esfuerzo realizado en los últimos años en el ámbito de la investigación para la adecuación de las nuevas tecnologías a los requerimientos de personas de edad avanzada y el notable aumento de la presencia tecnológica en nuestros entornos domésticos, laborales, de recreo, etc., han contribuido a una mayor conciencia y familiaridad de nuestros mayores con las tecnologías (Barnard et al., 2013) y (Heart and Kalderon, 2013). Esto provoca, consecuentemente, una mayor predisposición al uso de las mismas en su vida cotidiana, y les permite beneficiarse de ellas a través de una participación activa y consciente en las actividades relacionadas con, entre otros, la salud, el ocio y la promoción de las relaciones sociales, es decir, factores determinantes que fomentan el *envejecimiento activo* (Boudiny, 2013) y (Sixsmith and Gutman, 2013). En efecto, las tres dimensiones principales que cubren gran parte de la atención de salud en el marco de las etapas tempranas e intermedias del envejecimiento activo son: el ejercicio físico, la alimentación sana y la estimulación cognitiva (World Health Organization, 2002b).

Sin embargo, diversos estudios demuestran que las personas mayores no se alimentan bien, tanto en España (Dapcich et al., 2004) y (Triana et al., 2006) como fuera de ella (Evans and Cyr-Campbell, 1997) y (Pirlich and Lochs, 2001). Las razones son varias: soledad, dentición, escasa movilidad, pérdida de apetito o dolor. El resultado es una dieta no balanceada que puede tener efectos dañinos para la salud.

No obstante, a la hora de confeccionar dietas, existen recomendaciones generales dirigidas a personas mayores, permitiéndoles cubrir las necesidades de energía y otras sustancias útiles para el cuerpo (nutrientes). La importancia de una dieta balanceada es especialmente relevante en este sector de la población y debería ser planificada o recomendada teniendo en cuenta los cambios asociados con la edad, que son aquellos que están relacionados con las carencias de nutrientes y una peor calidad de vida.

En el transcurso de esta tesis, se ha desarrollado NutElCare, un Sistema de Recomendación (SR) cuya finalidad es proporcionar recomendaciones alimenticias a personas mayores. En el sistema, para cumplir las recomendaciones nutricionales que requieren estas personas, se han seguido las guías presentes en diversos estudios (Arbonés et al., 2003), (Dapcich et al., 2004), (Pinilla and de Torres Aured, 2007), (SENPE and SEGG, 2007), y se ha contado con la

colaboración de expertos nutricionales durante el proceso de desarrollo. En dicho proceso se ha empleado el uso intensivo de tecnologías de la Web Semántica (ver capítulo 2) para dar solución a los problemas inherentes a los SR estudiados en el capítulo 1 y aprovechar las posibilidades de razonamiento y reutilización de dichas tecnologías (ver capítulo 3). El sistema pretende afrontar los retos que aparecen en los SR Nutricionales en general, explicados en el capítulo 4, particularmente, el seguimiento de las ingestas del usuario, la adecuación de las recomendaciones tanto a sus gustos personales como a sus requisitos de salud y, el cumplimiento del *user compliance* manteniendo al usuario motivado mientras utiliza el sistema, enfocándolos a un sector más delicado de la población, como es el de las personas mayores.

Actualmente, NutElCare es parte de VIRTRA-EL (Rute-Pérez et al., 2014), una plataforma cuya finalidad es dar soporte al envejecimiento activo. Esta integración tiene como objetivo contribuir al fomento del envejecimiento activo desde una perspectiva global, suministrando sustento a las tres dimensiones principales asociadas a la salud.

En el presente capítulo, se introduce el funcionamiento básico del sistema de recomendación nutricional, identificando sus elementos claves. A continuación, se presenta la arquitectura básica en función de sus componentes principales, cómo han sido desarrollados y cómo interaccionan entre sí para producir recomendaciones saludables personalizadas. Para finalizar, se explican los principios básicos de usabilidad y accesibilidad seguidos para ajustarse a las necesidades específicas de las personas mayores.

En posteriores capítulos, se profundizará en la adquisición y representación del conocimiento semántico para representar alimentos, dietas y perfiles de usuario, así como la explotación de dicho conocimiento y la producción de las recomendaciones al usuario.

5.2. Funcionamiento del Sistema

Previamente a su desarrollo, es fundamental identificar las funcionalidades y requisitos principales del sistema:

- Obtención del perfil de usuario. El sistema debe permitir obtener dicho perfil a partir de las cualidades que determinarán la pertenencia a un conjunto de usuarios, por lo que debe definirse cuáles son y cómo se recogerán esas cualidades.
- Estrategia de recomendación. Se debe definir cuál va a ser la estrategia de recomendación nutricional. Particularmente, necesitamos fijar:
 - si se van a recomendar dietas o recetas,
 - cuál va a ser el paradigma de recomendación: basado en contenido, colaborativo, basado en conocimiento o un híbrido entre ellos,
 - selección de características que se usarán en los cálculos de similaridad y si estarán basados en dietas, recetas o ingredientes.
- Personalización de las recomendaciones. Se han de definir cada uno de los grados de personalización para adaptar las recomendaciones, tanto a los gustos, como a las necesidades de los usuarios.
- Monitorización y aprendizaje. Debe definirse la metodología de monitorización y almacenamiento, qué información ha de obtenerse en el seguimiento, qué técnica de *screening*, las alarmas y notificaciones, así como, la estrategia de aprendizaje de preferencias.

Dar respuesta a esos puntos clave constituye el primer paso hacia la definición del funcionamiento del sistema y la correspondiente arquitectura. Aunque en capítulos posteriores se profundizará en la definición formal y cuestiones de diseño de dichos elementos clave, podemos introducir a grandes rasgos el funcionamiento de NutElCare:

- Cuando el usuario llega al sistema, si no está registrado, debe registrarse y rellenar un formulario con la información necesaria para la construcción de su perfil nutricional, que se almacenará en una ontología de perfil de usuario, constituyendo la parte explícita de dicho perfil.
- A partir de dichas características nutricionales, el sistema recuperará un modelo de dieta ajustado a su perfil mediante el paradigma de recomendación basado en conocimiento.
- Esta dieta debe ser personalizada inicialmente atendiendo a gustos y necesidades:
 - El sistema debe detectar qué alimentos de la dieta son contraproducentes (alergias, contraindicaciones, etc.) o no son del agrado del usuario.
 - Dichos alimentos detectados han de ser sustituidos por otros nutricionalmente similares, por lo que las características nutricionales han de ser tenidas en consideración.
- Cuando el usuario obtiene la dieta recomendada, puede querer o necesitar variar ciertos platos en cualquier momento, por lo que el sistema debe ser capaz de ofrecerle recomendaciones de características nutricionales similares entre las que poder escoger, indicando el motivo del cambio entre una lista de posibles opciones. El objetivo de esta funcionalidad es lograr un mayor grado de flexibilidad y satisfacción, a la hora de que el usuario tenga que confeccionar sus propias dietas que, al estar siempre guiadas por el sistema, cumplirán los requisitos nutricionales que contribuyen a su salud.
- Todas estas acciones que el usuario va realizando en el sistema deben ser almacenadas para su monitorización. Asimismo, el historial de selecciones de alimentos ha de ser utilizado para la inferencia de sus preferencias en recomendaciones posteriores. Cada cierto tiempo, atendiendo a los requisitos del *screening* nutricional escogido, el usuario deberá rellenar los formularios de *screening* permitiendo al sistema (y por ende, a nutricionistas, cuidadores y al mismo usuario) conocer su estado nutricional tras ciertos periodos de utilización del mismo. De igual modo, nutricionistas y cuidadores pueden solicitar diversos informes de índole nutricional.
- Siempre que detecte una deficiencia, el sistema debe enviar alarmas o notificaciones a quien corresponda.

En el apartado siguiente se introduce la arquitectura básica del sistema. Concluimos con los principios de usabilidad y accesibilidad dirigidos a personas mayores.

5.3. Arquitectura del Sistema

La arquitectura de NutElCare se basa en los elementos principales de los sistemas de recomendación semánticos, a saber:

- Base de conocimiento y representación de los ítems: la base de conocimiento se construye mediante una ontología nutricional y los ítems (es decir, las dietas y alimentos) son representados como individuos en la ontología.

- Técnicas de perfil de usuario y de aprendizaje de los intereses de los usuarios: el perfil de usuario se modela mediante una ontología y se llevan a cabo técnicas de inferencia automática para aprender del comportamiento de los usuarios.
- Obtención y comunicación de recomendaciones sobre los elementos de la base de conocimiento a través de medidas de similitud semántica.

Basándonos en los puntos clave de funcionamiento citados en el apartado anterior y en los elementos principales de los Sistemas de Recomendación Semánticos, podemos representar la arquitectura del sistema como se muestra en la figura 5.1, donde se diferencian los siguientes componentes: la interfaz de usuario, la base de conocimiento, formada, a su vez, por una ontología nutricional y el perfil de usuario ontológico, el administrador de dietas que gestiona un repositorio de dietas, y los componentes de recomendación, *screening*, monitorización y aprendizaje.

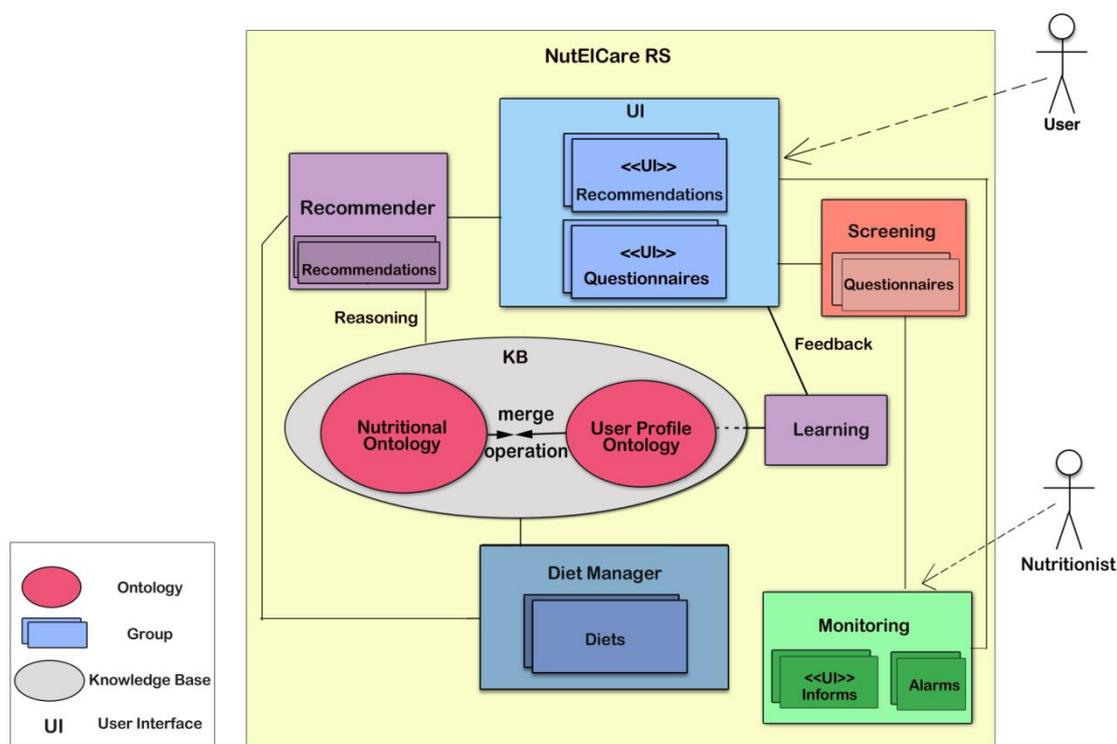


Figura 5.1. Arquitectura de NutEiCare. Fuente (Espín et al., 2016).

En dicha figura puede verse que la información y conocimiento necesarios acerca del usuario para llevar a cabo las recomendaciones se representan por medio de una ontología. Una vez que el usuario entra en el sistema a través de *la interfaz de usuario*, se crea una instancia del *perfil de usuario ontológico* correspondiente y se incluye en la *base de conocimiento*, que previamente contenía la *ontología nutricional*. La *ontología nutricional* está formada por una taxonomía de alimentos y dietas enriquecidas con propiedades nutricionales y contiene los aspectos que se combinan en los procesos de recomendación. En el momento en que se instancia el *perfil de usuario*, se enlazan ambas ontologías y son capaces de soportar el razonamiento como si se tratara de una sola ontología desde un punto de vista lógico.

A continuación, el *administrador de dietas* calcula y recupera la dieta que más se ajusta al perfil de usuario desde un *repositorio XML* que contiene dietas representadas en ese formato.

Esta dieta debe ser personalizada a fin de tener en cuenta las preferencias de usuario junto con los requisitos individuales, por lo que se envía al *componente de recomendación* que realiza operaciones de razonamiento sobre la base de conocimiento para adecuar la dieta al usuario, por ejemplo, reemplazar los alimentos contraindicados por otros. La salida del *componente recomendador* es la dieta ya personalizada, que se envía a la *interfaz de usuario*. Cuando el usuario solicita modificar algún alimento de la dieta, escoge entre las diferentes alternativas que se le ofrecen y el *componente de aprendizaje* infiere las preferencias implícitas del usuario mediante técnicas de inferencia automatizadas. El resultado de este análisis se instancia en la *ontología del perfil de usuario*, permitiendo, de esa manera, la inclusión de las preferencias aprendidas en los diferentes procesos de recomendación.

Los componentes de *screening* y *monitorización* (en la figura se representan separados entre sí para su distinción desde un punto de vista lógico de la arquitectura, aunque el *screening* puede considerarse parte del componente de *monitorización*), permiten realizar un seguimiento del comportamiento del usuario. Los resultados de los formularios de *screening*, una vez cumplimentados por el usuario, se envían al componente de *monitorización*. Dicho componente tiene, a su vez, una interfaz accesible a nutricionistas / cuidadores a través de la que pueden solicitar diferentes tipos de informes. A su vez, cuando el componente de *monitorización* detecta alguna irregularidad o algún hecho preocupante, como riesgo de desnutrición o deficiencia de nutrientes, notifica a los usuarios y / o a los nutricionistas / cuidadores por medio de sus interfaces y mediante correo electrónico si es necesario.

El sistema sigue una arquitectura basada en el modelo cliente-servidor y está desarrollado para ser utilizado en entorno web (aunque parte de su funcionamiento puede llevarse a cabo *offline*). El acceso a sus componentes se lleva a cabo mediante una API y unas interfaces bien definidas que permiten que dichos componentes puedan ser reemplazados o reutilizados en diferentes aplicaciones, favoreciendo además la evolución independiente de los mismos. La comunicación entre dichos componentes se realiza mediante servicios web REST. La parte de cliente, o *front-end* (incluyendo la interfaz de usuario), se implementa mediante PHP y *bootstrap*, mientras que la parte de servidor, o *back-end*, está desarrollada con Java JDK 1.8. Para la gestión y uso de ontologías se utiliza la librería Java OWL API (versión 4) y el razonador Pellet (ambos presentados en el capítulo 2).

5.3.1 Principios de usabilidad y accesibilidad del sistema

Una clave esencial para el éxito del sistema es que sea fácil de usar y agradable, motivando así a los usuarios a una utilización correcta y seria, potenciando el *user compliance*. Como consecuencia de las limitaciones funcionales y una menor experiencia tecnológica, los usuarios mayores experimentan más problemas de usabilidad que los más jóvenes (Ijsselsteijn et al., 2007). Por esta razón, el diseño de la interfaz de usuario se ha abordado teniendo especial cuidado de estos condicionantes particulares.

Aunque NutElCare se puede utilizar *offline*, su mayor potencial se consigue *online* debido a que algunos componentes necesitan acceder y mantener los datos de diferentes servidores. Además, en este momento, los dispositivos utilizados en las experiencias piloto consisten en computadoras personales o portátiles. Estos hechos han llevado a la aplicación de directrices, heurísticas y mejores prácticas en cuanto al diseño de interfaces de usuario para los ancianos en el contexto de las aplicaciones basadas en navegador, como los especificados en (Becker, 2004), (Kurniawan and Zaphiris, 2005) y (Johnson and Kent, 2007).

Los elementos clave que se consideran en el diseño para lograr una interfaz agradable y útil son los utilizados en el proyecto VIRTRA-EL y que ya han sido probados en (Rute-Pérez et al., 2014). Éstos se pueden resumir como:

- Aplicación de los requisitos de usabilidad generales para las personas mayores, como un tamaño de fuente personalizable, fuentes de fácil lectura, mayor contraste entre fondo y contenido, espaciado suficiente entre líneas y eludir el uso de *pop-ups* o de animaciones innecesarias (Kurniawan and Zaphiris, 2005).
- Evitar la frustración: cada acción debe tener una reacción. El usuario no puede perderse durante el uso del sistema y no debe haber elementos perturbadores que supongan interferencias, es decir, sin un propósito, en la interfaz de interacción. Los siguientes pasos deben estar claramente indicados y la interacción debe ser sencilla. Proporcionar guías para los usuarios durante el uso del sistema favorece la interacción con el mismo y aumentan la satisfacción del usuario.
- Evitar el aburrimiento: incluso cuando los usuarios son conscientes de que el objetivo del sistema es su propio beneficio para la salud, hay que evitar un uso aburrido del mismo. La inclusión de elementos de decoración no perturbadores, colores agradables y adecuados, sonidos, pictogramas, imágenes y frases habladas ayuda a que el usuario se sienta cómodo en el sistema sin sensación de aburrimiento.
- Promover la motivación: una retroalimentación positiva es muy efectiva para mantener al usuario motivado en el uso del sistema. Un ejemplo es una recompensa y un mensaje positivo cuando el usuario ha seguido el plan diario y una recompensa mayor cuando se logra una semana completa. Las explicaciones sobre las razones y los beneficios para la salud de las recomendaciones ayudan también a mantener al usuario motivado.

Una muestra significativa de los esfuerzos en el seguimiento de estas directrices es el uso de un sistema pictográfico destinado a personas con necesidades especiales cognitivas causadas por la edad o discapacidad. Los sistemas pictográficos tienen la ventaja de permitir, desde un nivel muy básico de comunicación para personas con niveles cognitivos bajos o en etapas muy tempranas, hasta un nivel de comunicación muy rico y avanzado. El sistema pictográfico elegido ha sido ARASAAC⁵², ampliamente utilizado en los distintos territorios de España. Un ejemplo de la interfaz de un cuestionario de *screening* sistemático mediante pictogramas ARASAAC se muestra en la figura 6.2.

Las propiedades no funcionales (como la adaptabilidad, la interacción y la heterogeneidad de anticipación) de los Home Care Systems (HCS) tienen un impacto importante en la aceptación general del sistema y, además, se han considerado las restricciones éticas, sociales, médicas y tecnológicas recogidas en (Kleinberger et al., 2007).

⁵²ARASAAC. Gobierno de Aragón, Portal Aragonés de la Comunicación Aumentativa y Alternativa. <http://www.catedu.es/arasaac/>.

A+ A A-

How many times did you eat last week...

MEAT
FISH
EGGS
RICE & PASTA
FRUITS
VEGETABLES
MILK PRODUCTS
SWEETS

RED MEAT (Beef or Veal)



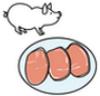
NEVER
 1 - 2
 3 - 4
 5 - 6
 EVERYDAY
 I DONT REMEMBER

POULTRY MEAT (Chicken or Turkey)



NEVER
 1 - 2
 3 - 4
 5 - 6
 EVERYDAY
 I DONT REMEMBER

PORK MEAT



NEVER
 1 - 2
 3 - 4
 5 - 6
 EVERYDAY
 I DONT REMEMBER

LAMB MEAT



NEVER
 1 - 2
 3 - 4
 5 - 6
 EVERYDAY
 I DONT REMEMBER

Figura 5.2. Ejemplo de uso de pictogramas ARASAAC en NutElCare. Fuente: (Espín et al., 2016).

Referencias del capítulo

- ARBONÉS, G., CARBAJAL, A., GONZALVO, B., GONZÁLEZ-GROSS, M., JOYANES, M., MARQUES-LOPES, I. & PUIGDUETA, I. 2003. Nutrición y recomendaciones dietéticas para personas mayores: Grupo de trabajo "Salud pública" de la Sociedad Española de Nutrición (SEN). *Nutrición hospitalaria*, 18, 109-137.
- BARNARD, Y., BRADLEY, M. D., HODGSON, F. & LLOYD, A. D. 2013. Learning to use new technologies by older adults: Perceived difficulties, experimentation behaviour and usability. *Computers in Human Behavior*, 29, 1715-1724.
- BECKER, S. A. 2004. A study of Web usability for older adults seeking online health resources. *ACM Transactions on Conmputer-Human Interaction*, 11, 387-406.
- BOUDINY, K. 2013. 'Active ageing': From empty rhetoric to effective policy tool. *Ageing and society*, 33, 1077-1098.
- DAPCICH, V., CASTELL, G. S., BARBA, L. R., RODRIGO, C. P., BARTRINA, J. A. & MAJEM, L. S. 2004. Necesidades Nutricionales en el Envejecimiento. *Guía de la Alimentación Saludable. Sociedad Española de Nutrición Comunitaria, Madrid, Everest*, 96-98.
- ESPÍN, V., HURTADO, M. V. & NOGUERA, M. 2015. Nutrition for Elder Care: a nutritional semantic recommender system for the elderly. *Expert Systems*, n/a-n/a.
- EVANS, W. J. & CYR-CAMPBELL, D. 1997. Nutrition, exercise, and healthy aging. *Journal of the American Dietetic Association*, 97, 632-638.
- HEART, T. & KALDERON, E. 2013. Older adults: are they ready to adopt health-related ICT? *International journal of medical informatics*, 82, e209-e231.
- IJSSELSTEIJN, W., NAP, H. H., KORT, Y. & POELS, K. 2007. Digital game design for elderly users. *Proceedings of the 2007 conference on Future Play. ACM*.

- JOHNSON, R. & KENT, S. 2007. Designing universal access: web-applications for the elderly and disabled. *Cognition, Technology & Work*, 9, 209-218.
- KURNIAWAN, S. & ZAPHIRIS, P. 2005. Research-derived web design guidelines for older people. *Proceedings of the 7th international ACM SIGACCESS conference on Computers and accessibility*. ACM.
- PINILLA, M. F. & DE TORRES AURED, M. L. 2007. La Dieta equilibrada: guía para enfermeras de Atención Primaria. *Sociedad Española de Dietética y Ciencias de la Alimentación*.
- PIRLICH, M. & LOCHS, H. 2001. Nutrition in the elderly. *Best Practice & Research Clinical Gastroenterology*, 15, 869-884.
- RUTE-PÉREZ, S., SANTIAGO-RAMAJÓ, S., HURTADO, M. V., RODRÍGUEZ-FÓRTIZ, M. J. & CARACUEL, A. 2014. Challenges in software applications for the cognitive evaluation and stimulation of the elderly. *Journal of NeuroEngineering and Rehabilitation*, 11, 88.
- SENPE & SEGG 2007. *Valoración nutricional en el anciano. Documento de Consenso*, Galenitas-Nigra Trea.
- SIXSMITH, A. & GUTMAN, G. M. 2013. *Technologies for Active Aging*, New York, Springer.
- TRIANA, F. C., GONZÁLEZ, C. R. & MARTÍN, P. M. 2006. Valoración nutricional en el anciano. *Medicine-Programa de Formación Médica Continuada Acreditado*, 9, 4037-4047.
- WORLD HEALTH ORGANIZATION, W. 2002. Active ageing: A policy framework.

CAPÍTULO 6

ADQUISICIÓN Y REPRESENTACIÓN DEL CONOCIMIENTO DESDE EL ENFOQUE SEMÁNTICO

Este capítulo describe los procesos de adquisición y formalización de conocimiento en NutElCare mediante su representación desde un enfoque semántico, en concreto, mediante la descripción ontológica de dicho conocimiento en OWL 2. En las fases de adquisición de conocimiento se recoge la información a introducir en el sistema, tanto del dominio nutricional como del perfil del usuario, constituyendo el marco conceptual del sistema. En las fases de representación de conocimiento se lleva a cabo la formalización del marco conceptual en la base de conocimiento utilizando ontologías OWL. Al inicio de cada proceso de representación explicado en el capítulo, se presentan los estudios de otras ontologías de alimentos y de usuario para su reutilización en nuestro sistema. Para cada ontología, se muestra su metodología de diseño y desarrollo y, finalmente, se explica cómo se unen la ontología nutricional y el perfil ontológico del usuario, para razonar como una sola ontología en la base de conocimiento. A lo largo de la descripción del proceso de desarrollo de las ontologías se muestran ejemplos de las clases, atributos, relaciones y restricciones que se han introducido.

6.1. Introducción

La base de conocimiento es el núcleo central de NutElCare. En ella se almacena toda la información nutricional necesaria para el funcionamiento del sistema explicado en el capítulo anterior. Para una representación completa y eficiente de dicha información en la base de conocimiento, es necesario establecer un *marco conceptual* que recoja los principales conceptos, sus atributos, relaciones y restricciones que se suelen manejar en el ámbito de los Sistemas de Recomendación (SR) nutricionales. Esta concepción de marco conceptual se asemeja bastante a varias de las definiciones de ontología vistas en el capítulo 2 (Gruninger and Fox, 1995), (Chandrasekaran et al., 1999) y (Noy and McGuinness, 2001). La mayor diferencia recae en la afirmación de que las ontologías constituyen una representación “formal”, lo cual no es inherente a un marco conceptual.

En este capítulo se presentan las fases de adquisición de conocimiento del sistema y de representación formal del marco conceptual. Dicho marco conceptual se formalizará mediante ontologías OWL 2 (Grau et al., 2008) para inferir nuevo conocimiento gracias a la capacidad de razonamiento que ofrece la equivalencia entre OWL y la Lógica Descriptiva (Description Logics, DL) (Baader et al., 2005).

La estrategia de recomendación de NutElCare, presentada en el capítulo 5, conlleva que sea necesario conocer la composición de los alimentos en general. Para ello hay que disponer de tablas de composición y de un esquema de agrupamiento de los diferentes alimentos por sus características, teniendo siempre presentes las recomendaciones de ingesta de energía y nutrientes que indican los organismos y comités de expertos (SENPE and SEGG, 2007). Toda esta información se recopila en la fase de adquisición de conocimiento nutricional a partir de diversas fuentes externas aportadas por los nutricionistas y geriatras expertos que colaboran en este trabajo.

Recordemos que una de las mayores ventajas del uso de ontologías es su reutilización, por lo que inicialmente, en la fase de representación de conocimiento nutricional, se realizó un análisis de fuentes ontológicas alimenticias existentes con el fin de ser reusadas en la base de conocimiento. Una vez seleccionada una de las fuentes estudiadas, se llevó a cabo una tarea de reducción de la misma mediante la eliminación de la información no relevante. A partir de la ontología obtenida, se realizó el proceso de extensión (Gruber, 1995) de la misma dando lugar a la ontología de dominio (Guarino, 1998) que se utilizará en el sistema. La extensión consistió en la adición de las clases, relaciones y restricciones que son relevantes para la formalización del marco conceptual pero que no venían recogidas en la ontología reutilizada. Este proceso también es conocido en la literatura como *enriquecimiento* de la ontología (*ontology enrichment*), el cual abarca la adición del conocimiento nuevo inferido a la propia ontología y la labor de completar la ontología con nueva información sobre el mismo dominio (Bühmann and Lehmann, 2012).

El enriquecimiento de una ontología puede ser un proceso manual o semi-automático. Las dietas que se van a recomendar en el sistema, fueron almacenadas como documentos XML en un repositorio de dietas. Para enriquecer la ontología nutricional con el modelo semántico de las dietas, se definió el modelo sintáctico mediante un documento XSD (XML Schema) y se establecieron reglas de mapeo entre los nodos del XSD y los conceptos OWL para posteriormente realizar una transformación XSLT que nos proporcionase la representación OWL de los conceptos que intervienen en la composición de las dietas. Este proceso semi-automático se explicará con más detalle en el apartado 6.3 y está publicado en (Espín et al., 2015).

En lo que respecta a la adquisición de conocimiento acerca del usuario para su representación mediante perfiles, se introduce inicialmente un análisis de los requisitos de usuario necesarios en los procesos de recomendación nutricional.

Sabiendo que las recomendaciones están basadas en las preferencias y necesidades de los usuarios, una representación efectiva y concisa de los usuarios es un elemento clave (Trajkova and Gauch, 2004). Los modelos de usuario representan las características del usuario que vienen principalmente determinadas por el dominio de conocimiento y el contexto del sistema, por lo que el perfil del usuario en NutElCare está modelado mediante una ontología que recoge todas las características y requisitos del usuario en el dominio nutricional. Dicha representación ontológica permite instanciar el perfil en la base de conocimiento cuando el usuario entra en el sistema. Durante esta fase se estudió también la posibilidad de reutilización de ontologías de perfil de usuario existentes y se realizó un proceso de enriquecimiento de la ontología para cubrir las necesidades del sistema.

Para que el sistema pueda llevar a cabo las recomendaciones, es necesario que ambas ontologías (nutricional y perfil de usuario) trabajen juntas en la base de conocimiento, por lo que es preciso un proceso de combinación de ambas ontologías en una sola, a la que, finalmente se le añaden las relaciones y restricciones necesarias para razonar con los conceptos integrados. En el apartado 6.6 se muestran ejemplos de las restricciones añadidas.

Como ya se vio en el capítulo 2, el razonador Pellet (Sirin et al., 2007), que es el utilizado en el sistema, soporta el razonamiento con reglas SWRL. Las reglas SWRL (Horrocks et al., 2004) aportan diferente nivel expresivo al proporcionado por OWL y pueden utilizarse para clasificar conceptos o propiedades y para representar ciertas restricciones u operaciones que no

pueden ser definidas mediante OWL. Al final del capítulo se presentan los distintos tipos de reglas SWRL introducidas, así como algunos ejemplos de cada tipo.

6.2. Adquisición de Conocimiento Nutricional

El paso previo al diseño de la base de conocimiento del sistema consistió en la recopilación de la información nutricional a incluir en la misma. Para ello, los expertos en nutrición con los que se ha contado han aportado material previamente confeccionado o revisado por ellos. Al provenir de fuentes expertas, el material puede considerarse totalmente fiable y lo más completo posible para poder ofrecer recomendaciones precisas y eficientes.

El material aporta información sobre:

- Características nutricionales de los alimentos.
- Tipos de dietas para personas mayores.
- Recomendaciones nutricionales saludables para personas mayores.
- Restricciones alimenticias a considerar.

En los siguientes sub-apartados se resume el contenido de dicho material.

6.2.1 Características Nutricionales de los Alimentos

Las propiedades nutricionales de los alimentos se han proporcionado mediante tablas de composición nutricional consistentes en alimentos con diferentes valores nutricionales. En la figura 6.1 se muestra un extracto de una tabla de composición nutricional. A continuación, en la figura 6.2 se muestran los principales nutrientes por grupo de alimentos.

1	ALIM	KCAL	PROTg	HCARg	GRASg	GSATg	GINASg	COLEmg	FIBRAg	SODIOg	CALCmg	FOSFmg	MAGNmg	HIERRmg	vitAMg	vitB1mg
2	NARANJA (ZUMO DE)	47,000	0,700	10,600	0,200	0,000	0,190	0,000	0,000	0,001	11,000	16,000	12,000	0,200	0,012	0,100
3	NATA	320,000	2,250	3,500	33,000	22,000	11,000	124,000	0,000	0,032	78,000	59,000	9,000	0,001	0,300	0,030
4	NATILLAS	94,000	2,800	18,000	1,200	0,750	0,450	5,000	0,000	0,040	100,000	80,000	10,000	0,001	0,030	0,030
5	NUECES	666,000	15,000	12,000	62,000	6,450	55,520	0,000	4,600	0,002	100,000	410,000	135,000	3,000	0,010	0,300
6	PALITOS SALADOS	389,000	11,000	75,000	5,000	3,000	1,980	0,000	0,001	1,800	147,000	310,000	30,000	0,700	0,190	0,010
7	PALOMETA FRESCA	125,000	20,000	0,000	5,000	0,000	4,970	70,000	0,000	0,110	25,000	210,000	28,000	0,700	0,000	0,050
8	PAN BLANCO	240,000	8,000	47,950	1,800	0,450	1,330	0,000	3,000	0,540	58,000	90,000	0,900	0,900	0,015	0,050
9	PAN INTEGRAL	208,000	7,500	41,125	1,500	0,370	1,210	0,000	7,000	0,527	43,000	220,000	70,000	3,600	0,050	0,190
10	PAN MORENO	225,000	7,000	46,100	1,400	0,350	1,020	0,000	4,700	0,536	23,000	183,000	40,000	2,300	0,000	0,180
11	PAN RALLADO	349,000	13,000	72,000	1,000	0,250	0,730	0,000	3,000	0,540	60,000	90,000	0,900	1,200	0,000	0,050
12	PASAS	280,000	3,000	65,200	0,800	0,000	0,790	0,000	11,000	0,002	80,000	15,000	42,000	3,000	0,030	0,120
13	PASTELITOS DE ALMEN	376,000	5,000	35,000	24,000	6,780	17,210	1,000	6,000	0,059	115,000	209,000	94,000	1,900	0,004	0,080
14	PASTELITOS DE NUECE	512,000	11,000	54,000	28,000	8,020	19,950	0,000	4,000	0,025	5000,000	195,000	50,000	1,500	0,001	0,100
15	PASTELITOS VARIADOS	450,000	11,000	70,000	14,000	3,600	10,380	0,000	0,001	0,060	4000,000	260,000	16,000	1,400	0,002	0,080
16	PASTILLAS (CALDO C)	242,000	23,500	10,000	12,000	6,290	5,700	93,000	0,000	0,003	0,000	75,000	20,000	0,000	0,000	0,000
17	PATATAS	71,000	2,000	15,525	0,100	0,000	0,090	0,000	2,500	0,003	10,000	50,000	25,000	1,000	0,002	0,100
18	PATATAS CHIPS	539,000	5,500	40,600	39,400	24,400	15,000	16,000	6,000	0,450	52,000	147,000	64,000	2,300	0,010	0,020
19	PATO	230,000	17,000	0,000	18,000	3,910	14,070	70,000	0,000	0,140	12,000	187,000	15,000	2,000	0,000	0,290
20	PAVO (ENTERO)	215,000	20,000	0,000	15,000	4,000	10,900	74,000	0,000	0,063	21,000	226,000	27,000	2,000	0,000	0,090
21	PAVO-MUSLO	114,000	20,625	0,000	3,500	0,930	2,460	75,000	0,000	0,086	13,000	190,000	17,000	2,000	0,000	0,090
22	PAVO-PECHUGA	105,000	24,000	0,000	1,000	0,260	0,720	60,000	0,000	0,046	13,000	195,000	20,000	1,000	0,001	0,050
23	PEPINILLO VINAGRE	17,000	1,000	2,800	0,200	0,000	0,190	0,000	0,400	0,960	30,000	30,000	5,000	1,600	0,002	0,030
24	PEPINOS	13,000	0,600	2,200	0,200	0,000	0,190	0,000	0,900	0,008	15,000	23,000	8,000	0,500	0,028	0,020
25	PERA	46,000	0,600	10,000	0,400	0,000	0,390	0,000	3,000	0,002	10,000	13,000	8,000	0,300	0,005	0,020
26	PERA EN ALMIBAR	76,000	0,300	18,250	0,200	0,000	0,190	0,000	2,000	0,004	6,000	8,000	4,000	0,300	0,002	0,010
27	PEREJIL	61,000	4,450	9,900	0,400	0,000	0,390	0,000	4,300	0,033	240,000	128,000	41,000	8,000	1,200	0,120

Figura 6.1. Extracto de tabla de composición nutricional.

	proteínas	lípidos	Carbohidratos	fibra	vitaminas	minerales
Cereales						
Bollería						
Aceite						
Pescado						
Lácteo						
Carne						
Legumbre						
Fruto seco						
Fruta						
Verdura						

Figura 6.2. Principales nutrientes por grupos de alimentos. Fuente: (Pinilla and de Torres Aured, 2007).

A partir de estas fuentes de información pueden obtenerse, tanto los diferentes nutrientes a contemplar en los procesos de recomendación (calorías, proteínas, hidratos de carbono, etc.), como las medidas utilizadas (gramos, miligramos, etc.) como una clasificación sencilla de alimentos (cereales, bollería, etc.) junto con su principal aporte nutricional. Todos estos conceptos han de formalizarse para representarlos en la base de conocimiento del sistema.

6.2.2 Tipos de Dietas para Personas Mayores

Por lo que respecta a la estrategia de recomendación propuesta, ésta debe partir de la obtención del tipo de dieta más adecuado para el usuario que utilizará el sistema. Para ello, los diferentes modelos de dieta para personas mayores se han clasificado según los siguientes criterios respondiendo a las necesidades de los diferentes perfiles de usuario:

- Dietas terapéuticas en función de:
 - Calorías: hipercalóricas e hipocalóricas.
 - Proteínas: hiperprotéicas e hipoprotéicas.
 - Trastornos de masticación: fácil masticación y trituradas.
 - Trastornos de deglución: fácil digestión, líquida o pastosa.
 - Intolerancias alimenticias: intolerancia al gluten, intolerancia a la lactosa.
- Dietas relacionadas con el contexto
 - Según estación del año: primavera-verano, otoño-invierno.
 - Según tipo de clima: húmedo, seco.
- Dietas restringidas por religión: Halal (musulmana), Kosher (judía).
- Dietas vegetarianas: semi-vegetariana, pescovegetariana, vegetariana estricta, ovo-vegetariana, lacto-vegetariana, lacto-ovo-vegetariana.
- Dietas estándar.

Es importante destacar que ésta es una clasificación inicial puesto que el sistema es flexible y permite crear nuevos tipos y clasificaciones desde la interfaz de administración. Algunos ejemplos de dietas se recogen en el Anexo A.

6.2.3 Recomendaciones Nutricionales para Personas Mayores

Según (Arbonés et al., 2003), (Dapcich et al., 2004) y (Pinilla and de Torres Aured, 2007) existen algunas recomendaciones generales para que las personas mayores puedan consumir una dieta prudente y saludable, teniendo en cuenta los cambios fisiológicos asociados con la edad. Algunos ejemplos son:

- No recalentar repetidas veces platos ya preparados.
- Evitar suprimir el desayuno y hacer más de una comida caliente al día.
- Disminuir el tamaño de las raciones, concentrando todos los nutrientes, en poca cantidad de alimento.
- Fomentar cinco ingestas al día (desayuno, almuerzo, comida, merienda y cena).
- Consumir grasas cardiosaludables. Estas grasas, como los ácidos omega-3, se encuentran en los pescados grasos (salmón, caballa, jurel, sardinas, etc.). Entre las grasas vegetales, el aceite de oliva es la mejor opción.
- Aumentar el consumo de fibra. Se logra consumiendo cereales integrales, frutas, verduras y legumbres. Así, se evita el estreñimiento y la diverticulitis, muy frecuentes en las personas mayores.

- Tomar leche cada día y/o derivados lácteos; carne o pescado o huevos; hortalizas cocidas y crudas; frutas enteras; 4-6 vasos de agua o infusiones o caldos de verduras.

Este tipo de recomendaciones son importantes puesto que fomentan la educación nutricional de la persona mayor, ayudando a mejorar sus conocimientos dietéticos y motivándole para una mejor elección de alimentos. En la figura 6.3 se recogen recomendaciones de grupos básicos de alimentos y nutrientes a consumir cada día:

GRUPOS BÁSICOS DE ALIMENTOS A CONSUMIR CADA DÍA ...			
FUNCIÓN EN EL ORGANISMO	GRUPO DE ALIMENTOS	NUTRIENTES Y OTRAS SUSTANCIAS SALUDABLES	RECOMENDACIONES
Construcción	Leche y lácteos	Proteínas y calcio	2 vasos de leche al día, o un vaso y 2 yogures, o un vaso y una cuajada, o leche y quesos magros (los hay bajos en sal, desnatados y semidesnatados)...
	Carne, pescado, Huevo y derivados	Proteínas y hierro	Como segundo plato de comida y cena: carnes, no más de 6 veces por semana; pescado, mínimo 3-4 veces por semana, y huevo, hasta 3 veces por semana.
Energía	Cereales Patatas Legumbres	Hidratos de carbono	Muy recomendables en cada una de las comidas del día. Las legumbres (fibra) entre 2 y 4 veces por semana.
	Aceites y grasas	Grasas esenciales y vitaminas liposolubles (A, D, E)	Controlar la cantidad.
Regulación	Verduras y hortalizas	Vitaminas, minerales, antioxidantes y fibra	Tanto en la comida como en la cena; como ingredientes de primeros platos y acompañamiento de los segundos. El tomate contribuye a un aporte de provitamina A cuando las dietas han de ser bajas en grasa.
	Frutas	Vitaminas, minerales, antioxidantes y fibra	Fresca, batida, al horno y en compota. Zumos no azucarados. Unas tres piezas diarias.

Figura 6.3. Grupos básicos de alimentos a consumir cada día y su función en el organismo.

Otras recomendaciones nutricionales, como el número de raciones recomendado semanalmente para cada tipo de alimento, se recogen en el Anexo B.

6.2.4 Restricciones Alimenticias

Existen algunas restricciones que deben ser también consideradas a la hora de recomendar una alimentación saludable para cada usuario. Por ejemplo, si una persona es celíaca, por tanto intolerante al gluten, aunque se le recomiende una dieta especial para celíacos, cuando el usuario quiera modificar algún plato o alimento de la dieta, el sistema debe tener en cuenta que no puede ofrecer alimentos que contengan gluten en las recomendaciones. A su vez, si un usuario es alérgico a cierto alimento, tipo de alimento o sustancia, ninguno de ellos puede aparecer en las recomendaciones. Este tipo de restricciones se deben considerar en todos los procesos de recomendación del sistema.

En el Anexo C se presenta el listado A100 (test de 100 alimentos) de las intolerancias alimenticias en la dieta mediterránea.

6.3. Representación del Conocimiento Nutricional

Toda la información recogida en la fase de adquisición de conocimiento se debe formalizar y representar en la base de conocimiento de NutElCare. La formalización del dominio nutricional se llevará a cabo mediante una *ontología de dominio*, es decir, tendrá un carácter general con el fin de ser reutilizada para describir el dominio nutricional en el ámbito de los sistemas de recomendación. Asimismo, esta ontología podrá verse como un metamodelo que guiará el proceso de recomendaciones en SR nutricionales de acuerdo con la filosofía de NutElCare.

Gracias a la posibilidad de reutilización de una ontología de dominio, puede llevarse a cabo una adaptación de la misma para el caso específico que nos ocupa: el sector de las personas mayores, sus recomendaciones concretas y la estrategia de recomendación propuesta. Uno de los principios en la creación de ontologías de dominio, es que éstas sean monotónicamente extensibles (Gruber, 1995), por lo que el enriquecimiento de la ontología podrá realizarse mediante una extensión de la ontología de dominio resultando en una *ontología de aplicación* (Guarino, 1998).

6.3.1 Creación de la Ontología de Dominio. Enfoque Top-Down

La primera parte de la creación de la ontología de dominio consistió en el seguimiento de los pasos propuestos en (Noy and McGuinness, 2001) presentados en el capítulo 2 (apartado 2.4) mediante un enfoque *top-down*:

PASO 1 - Determinar alcance y dominio de la ontología. En nuestro caso, el dominio viene determinado por los conceptos nutricionales, sus atributos y relaciones que son necesarios en el ámbito de los SR.

PASO 2 - Considerar la reutilización de ontologías existentes. Puesto que una de las mayores ventajas del uso de ontologías, es su reutilización, realizó un proceso de búsqueda de ontologías que pertenecieran al dominio de alimentación y nutrición de forma que sirvieran, tanto para la clasificación de los alimentos en una taxonomía jerárquica, como para la representación homogénea de las propiedades nutricionales y alimenticias de los mismos. En esta fase se encontraron varias ontologías relacionadas (capítulo 4), y finalmente se tomó la decisión de analizar en profundidad: Food Ontology de PIPS⁵³ (Personalized Information Platform for Health and Life Services) y Agrovoc Ontology de la FAO⁵⁴ (Food and Agriculture Organization of the United Nations), considerándose las más completas y con más carácter académico. En el análisis realizado se tuvo en cuenta el hecho de que la ontología a reutilizar pudiese contener información relevante para responder a cuestiones tales como:

- ¿Qué alimentos proporcionan nutrientes similares a un alimento específico?
- ¿Qué alimentos se toman como desayuno (media mañana, almuerzo, merienda o cena)?
- ¿Cuántas calorías contiene un plato?
- ¿Qué cantidad de un nutriente se debe tomar diariamente según el índice de masa corporal?
- ¿Cuántas raciones de un tipo de alimento se recomienda tomar semanalmente?

A continuación se presenta el análisis llevado a cabo para las dos ontologías mencionadas, con el fin de decidir si alguna se ajustaba a las necesidades del sistema NutElCare, o si tras el mismo se concluye que es necesaria la creación de una nueva ontología.

Ontología PIPS Food

PIPS es uno de los cinco proyectos integrados financiados con cargo a la primera convocatoria del 6º Programa Marco de la Unión Europea en el ámbito de Salud Electrónica (e-Health). El proyecto tiene como objetivo la creación de nuevo conocimiento para los Servicios de Apoyo de Salud, Vida y Medio Ambiente, para ayudar a los profesionales de la salud a tener

⁵³ PIPS <http://www.pips.eu.org>

⁵⁴ FAO <http://www.fao.org>

acceso al conocimiento médico pertinente y actualizado y, lo más importante, ayudar a los ciudadanos europeos a tomar decisiones informativas sobre sus estilos de vida. Una de las ontologías PIPS del dominio de Alimentación es PIPS Food Ontology, en formato OWL y en inglés, en el que los conceptos están organizados jerárquicamente de acuerdo con la clasificación estándar EUROCODE2⁵⁵, proporcionando a los usuarios información acerca de los diferentes tipos de alimentos como la cantidad de alimentos y líquidos que el usuario debe tomar un día, las características nutricionales de los mismos y cuáles son los ingredientes de algunos productos elaborados. Sin embargo, esta información no viene recogida en esta ontología, sino en otras complementarias donde los valores asociados a cada propiedad nutricional tampoco se encuentran instanciados.

En la figura 6.4. se muestra un extracto de la taxonomía jerárquica de la ontología PIPS Food desplegada mediante OntoGraf de Protégé (PROTÉGÉ, 2013), en su primer nivel y con la clase `Fruits` completamente desplegada. En ella se observa que no existen instancias definidas para cada tipo de alimento, es decir, no se encuentran las frutas específicas de cada tipo. Además, todos los tipos de alimento, incluso algunas instancias, descienden directamente de la clase `Thing` lo cual en principio ofrece poca información nutricional, puesto que no existe una jerarquía taxonómica que los relacione nutricionalmente.

⁵⁵ <http://www.ianunwin.demon.co.uk/eurocode/index.htm>

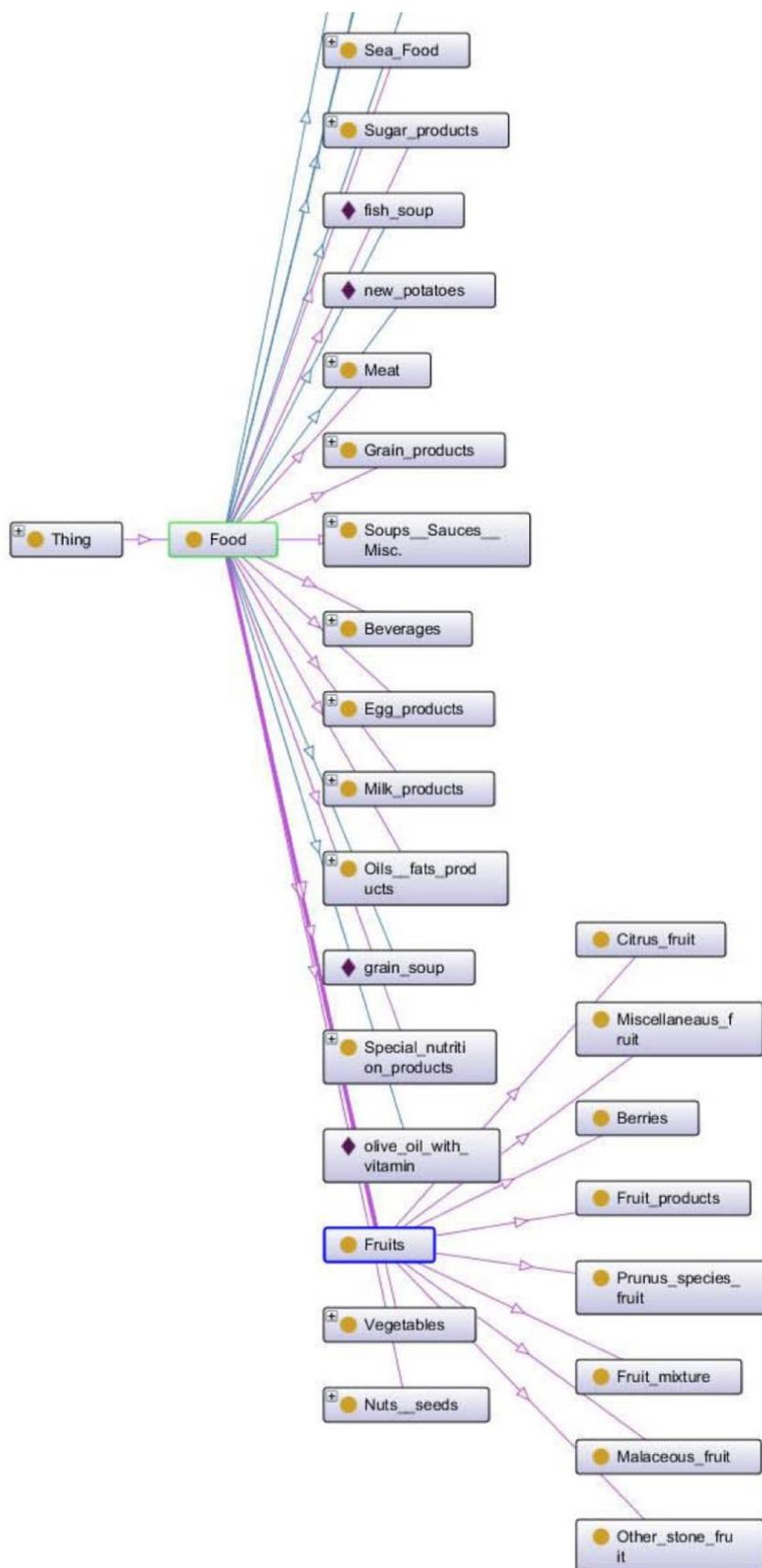


Figura 6.4. Extracto de la ontología PIPS Food desplegada en Protégé.

Tesauro AGROVOC

La Organización de las Naciones Unidas para la Agricultura y la Alimentación (FAO) es una organización específica de la ONU que conduce las actividades internacionales encaminadas a erradicar el hambre. Brinda sus servicios tanto a países desarrollados, como a países en vías de desarrollo; la FAO actúa como un foro neutral donde todas las naciones se reúnen como iguales para negociar acuerdos y debatir políticas. También, es fuente de conocimiento e información, ayudando a los países en vías de desarrollo y transición a modernizar y mejorar sus actividades agrícolas, forestales y pesqueras, con el fin de asegurar una buena nutrición para todos. Para el acceso a la tecnología de la información que usa la FAO, se ha creado un portal, llamado AIMS⁵⁶ (*Agricultural Information Management Standards*). AIMS es un espacio para acceder y discutir sobre los estándares de gestión de información, herramientas y tecnologías, conectando a los trabajadores de la información mundial para construir una comunidad de práctica global. Desde AIMS se puede acceder a los distintos vocabularios y uno de ellos es AGROVOC⁵⁷.

AGROVOC (Caracciolo et al., 2013) contiene más de 40.000 conceptos en hasta 21 idiomas que abarcan temas relacionados con la alimentación, la nutrición, la agricultura, la pesca, el medio ambiente y otros ámbitos relacionados. Se expresa en el Sistema de Organización Simple de Conocimiento (SKOS, ver capítulo 2, apartado 2.3) aunque está publicado en otros lenguajes como RDF y OWL. Además está alineado con otros vocabularios existentes.

Para un primer acercamiento a la ontología, y como familiarización, puede visitarse alguna de sus distintas versiones web:

- Navegación por el Tesauro AGROVOC. Se realiza a través del enlace IRI del concepto, por ejemplo, podemos visitar el concepto `products` mediante el enlace http://aims.fao.org/aos/agrovoc/c_6211.
- Búsqueda de términos AGROVOC. Esta navegación nos permite realizar búsquedas de conceptos en distintos idiomas y con diversos criterios de búsqueda. Se puede iniciar desde la URL <http://aims.fao.org/standards/agrovoc/functionalities/search>. Busquemos por ejemplo, el concepto “productos” en idioma español como “exact match”. Desde el resultado de esa búsqueda es posible acceder a cada término descendiente, bien para ver su jerarquía bien para ver su información.
- Versión de servicios web, que se pueden invocar desde cualquier aplicación cliente a través de llamadas SOAP estándar, devolviendo una respuesta SOAP estándar. El uso de servicios web, permite que se pueda acceder de forma inmediata desde las aplicaciones, reduciendo el tiempo y el esfuerzo necesario para descargar e incorporar la versión más reciente de los datos del servidor AGROVOC. La versión de descarga de Agrovoc Web Services está disponible en <https://bitbucket.org/aims-fao/agrovoc-web-services/>

Navegar en cualquiera de las versiones web de AGROVOC nos da suficiente información sobre la misma y su contenido. En este caso, al contrario que en PIPS Food, sí que es posible encontrar los alimentos específicos sobre los que nuestra ontología deberá razonar, sin embargo, la información nutricional de los mismos sigue siendo prácticamente nula.

⁵⁶ <http://aims.fao.org>

⁵⁷ <http://aims.fao.org/vest-registry/vocabularies/agrovoc-multilingual-agricultural-thesaurus>

Selección final de la ontología nutricional

Finalmente, ninguna de las dos ontologías ofrecía los datos nutricionales que eran necesarios en este trabajo. Llegados a este punto, se planteó usar una de las dos ontologías anteriores y completarla con otras fuentes nutricionales, o el desarrollo de una ontología desde cero. Por motivos académicos y de reutilización de dominios ontológicos se escogió reutilizar una de ellas. Y definitivamente se optó por el uso de AGROVOC debido a dos factores fundamentales: es más completa, en ella se encontraban más conceptos requeridos para realizar las recomendaciones y por razones de investigación e internacionalización, puesto que es más usada institucionalmente en organismos internacionales.

PASO 3 - Enumerar los términos importantes de la ontología. Durante el proceso de adquisición de conocimiento nutricional empezaron a identificarse los conceptos que definirían el marco conceptual de la base de conocimiento, ya que son aquellos términos necesarios para describir el conocimiento nutricional y que han de ser trasladados a la ontología. Algunos de estos conceptos son: *Alimentos (y tipos de Alimentos), Nutrientes (Proteínas, Calorías, etc.), Ingredientes, Platos, Dietas (y tipos de Dietas), Modo de preparación, Consumo recomendado, Intolerancias (A100-A200), Sustancias como gluten, lactosa, etc., Ingesta Diaria (Desayuno, Aperitivo, Almuerzo, Merienda, Cena), Plato de Ingesta (Entrante, Primer Plato, Segundo Plato, Guarnición, Postre), Ración casera, Raciones recomendadas,...*, etc.

PASO 4 - Definir las clases y la jerarquía de clases. Puesto que en el paso 2 se decidió reutilizar el tesoro AGROVOC en su versión OWL, es necesario un proceso de reajuste de dicha ontología para adaptarse a nuestro dominio. El tamaño del archivo OWL descargado, una vez descomprimido, es de 400 MB, lo cual impide que pueda abrirse con Protégé o cargarse en memoria con JDOM, por lo tanto, la tarea inicial en este paso fue reducir el tamaño de la ontología de modo que no se perdiera la información necesitada.

Reducción de AGROVOC

La complejidad de AGROVOC implica el uso de editores de texto para ficheros de gran tamaño, en este caso se ha usado LTF Viewer 5.2⁵⁸. Se muestra a continuación un fragmento de la ontología en su forma original descargada, por ejemplo, la codificación del concepto cuyo identificador de clase AGROVOC es c_6211 que corresponde al término “Products / Productos”:

```
<owl:Class rdf:about="&agrovoc;c_6211">
  <rdfs:subClassOf rdf:resource="&owl;Thing"/>
  <rdfs:subClassOf rdf:resource="c_domain_concept"/>
</owl:Class>
<agrovoc:c_6211 rdf:about="&agrovoc;i_6211">
  <hasScopeNote xml:lang="cs">Pokud mono pout specifitj deskriptor
</hasScopeNote>
  <hasScopeNote xml:lang="de">Spezifischeren Deskriptor benutzen, falls
geeignet </hasScopeNote>
  <hasScopeNote xml:lang="en">Use more specific descriptor when possible
</hasScopeNote>
  <hasScopeNote xml:lang="es">Use este descriptor solamente cuando otro mas
especifico no sea el adecuado </hasScopeNote>
```

⁵⁸ Large Text File Viewer <http://www.swiftgear.com/ltfviewer/features.html>

```
<hasScopeNote xml:lang="fr">Utiliser un ou des descripteurs plus
specifiques si possible </hasScopeNote>
<hasScopeNote xml:lang="hu">Ha lehetsges, pontosabb meghat&#225;roz&#225;st
haszn&#225;ljunk </hasScopeNote>
<hasScopeNote xml:lang="it">Usare il descrittore piu specifico possibile
</hasScopeNote>
<hasScopeNote xml:lang="pl">U&#380;ywaj specyficznych deskryptor&#243;w w
miar&#281; potrzeby </hasScopeNote>
<hasScopeNote xml:lang="pt">Usar descritor mais especifico sempre que
possivel </hasScopeNote>
<hasDateCreated rdf:datatype="&xsd:string">1981-01-09 00:00:00
</hasDateCreated>
<hasDateLastUpdated rdf:datatype="&xsd:string">1998-11-05 09:09:00
</hasDateLastUpdated>
<hasStatus rdf:datatype="&xsd:string">Published</hasStatus>
<isProducedBy rdf:resource="&agrovoc;i_24887"/>
<isObjectOfActivity rdf:resource="&agrovoc;i_16018"/>
<hasLexicalization rdf:resource="&agrovoc;i_ar_1299486359995"/>
<hasLexicalization rdf:resource="&agrovoc;i_cs_1299486360012"/>
<hasLexicalization rdf:resource="&agrovoc;i_de_1299486360029"/>
<hasLexicalization rdf:resource="&agrovoc;i_en_1299486360046"/>
<hasLexicalization rdf:resource="&agrovoc;i_es_1299486360065"/>
<hasLexicalization rdf:resource="&agrovoc;i_fa_1299486360086"/>
<hasLexicalization rdf:resource="&agrovoc;i_fr_1299486360110"/>
<hasLexicalization rdf:resource="&agrovoc;i_hi_1299486360135"/>
<hasLexicalization rdf:resource="&agrovoc;i_hu_1299486360161"/>
<hasLexicalization rdf:resource="&agrovoc;i_it_1299486360189"/>
<hasLexicalization rdf:resource="&agrovoc;i_ja_1299486360219"/>
<hasLexicalization rdf:resource="&agrovoc;i_ko_1299486360250"/>
<hasLexicalization rdf:resource="&agrovoc;i_pl_1299486360318"/>
<hasLexicalization rdf:resource="&agrovoc;i_pt_1299486360354"/>
<hasLexicalization rdf:resource="&agrovoc;i_ru_1299486360391"/>
<hasLexicalization rdf:resource="&agrovoc;i_sk_1299486360430"/>
<hasLexicalization rdf:resource="&agrovoc;i_th_1299486360472"/>
<hasLexicalization rdf:resource="&agrovoc;i_zh_1299486360515"/>
<hasLexicalization rdf:resource="&agrovoc;i_fa_1299486360642"/>
<hasLexicalization rdf:resource="&agrovoc;i_hi_1299486360693"/>
<hasLexicalization rdf:resource="&agrovoc;i_th_1299486360744"/>
<hasLexicalization rdf:resource="&agrovoc;i_cs_1299486360797"/>
<hasLexicalization rdf:resource="&agrovoc;i_sk_1299486360852"/>
<hasLexicalization rdf:resource="&agrovoc;i_zh_1299486360909"/>
<hasLexicalization rdf:resource="&agrovoc;i_it_1299486360967"/>
<hasLexicalization rdf:resource="&agrovoc;i_hu_1299486361027"/>
<hasLexicalization rdf:resource="&agrovoc;i_pl_1299486361090"/>
<hasLexicalization rdf:resource="&agrovoc;i_en_1299486361157"/>
<hasLexicalization rdf:resource="&agrovoc;i_ja_1299486361225"/>
<hasLexicalization rdf:resource="&agrovoc;i_de_1299486361292"/>
</agrovoc:c_6211>
```

Cada elemento contenido en `hasLexicalization` contiene a su valor en un idioma determinado, por ejemplo, en francés, `agrovoc;i_fr_1299486360110` nos está indicando qué elemento (`c_noun`) representa a ese concepto en francés:

```
<c_noun rdf:about="&agrovoc;i_fr_1299486360110">
  <rdfs:label xml:lang="fr">Produit</rdfs:label>
  <hasDateCreated rdf:datatype="&xsd:string">1981-01-21 00:00:00
</hasDateCreated>
  <hasCodeAgrovoc rdf:datatype="&xsd:int">6211</hasCodeAgrovoc>
  <isMainLabel rdf:datatype="&xsd:boolean">true</isMainLabel>
  <hasStatus rdf:datatype="&xsd:string">Published</hasStatus>
  <isLexicalizationOf rdf:resource="&agrovoc;i_6211"/>
</c_noun>
```

El número de idiomas en que se describe AGROVOC hace que su tamaño muy elevado. Para nuestro propósito sólo se consideraron relevantes los términos en español y en inglés (por motivos de internacionalización) por lo que la primera decisión tomada, consistió en la eliminación de la información en los demás idiomas. Aunque esta información no se encontrará en el sistema por motivos de eficiencia, siempre estará disponible gracias a que en la ontología final se mantendrá un enlace a su concepto AGROVOC correspondiente, permitiendo cargar dinámicamente otros idiomas. La tarea de eliminación de idiomas se realiza en tres pasos:

1. Eliminar el elemento `hasScopeNote` cuyo atributo `xml:lang` sea distinto de “en” y “es” para cada elemento `agrovoc:id_agrovoc`.
2. Notar que todos los identificadores de clase AGROVOC están formados por los caracteres “c_” seguidos por un número de identificación. Eliminar también sus elementos `hasLexicalization` que no contengan “i_en” o “i_es”.
3. Eliminar aquellos elementos `c_noun` que no contengan “i_en” ni “i_es”.

Para la eliminación de los elementos de otros idiomas se ha utilizado la librería JDOM de Java en entorno de programación eclipse⁵⁹ sobre pequeños fragmentos de la ontología, logrando reducirla de 400 a 91 MB, aproximadamente un 70 %. Con este tamaño, JDOM ya fue capaz de cargar la ontología completa en memoria.

A continuación, se eliminó la información extra de fecha, puesto que no es relevante para nuestro sistema, es decir, todas las etiquetas `hasDateCreated`, `hasDateLastUpdated` y `hasDate`. Además, se eliminaron aquellos elementos `c_noun` cuyo hijo `hasStatus` fuera distinto de “Published”, y después todos los tags `hasStatus`, ya que a partir de ese momento no muestran información relevante. Tras estos cambios, el tamaño de la ontología era de 65MB y ya era posible abrirla con Protégé. Al abrirla y guardarla en formato RDF/XML de OWL, el concepto “Productos” AGROVOC quedaría como sigue:

```
<!-- http://aims.fao.org/aos/agrovoc/c_6211 -->
<owl:Class rdf:about="&agrovoc;c_6211">
  <rdfs:subClassOf
rdf:resource="http://aims.fao.org/aos/common/c_domain_concept"/>
  <rdfs:subClassOf rdf:resource="&owl;Thing"/>
</owl:Class>
```

⁵⁹ Eclipse <http://eclipse.org>

```

<!-- http://aims.fao.org/aos/agrovoc/i_6211 -->
<owl:NamedIndividual rdf:about="&agrovoc;i_6211">
  <rdf:type rdf:resource="&agrovoc;c_6211"/>
  <hasScopeNote xml:lang="en">Use more specific descriptor when
possible</hasScopeNote>
  <hasScopeNote xml:lang="es">Use este descriptor solamente cuando otro mas
especifico no sea el adecuado</hasScopeNote>
  <isObjectOfActivity rdf:resource="&agrovoc;i_16018"/>
  <isProducedBy rdf:resource="&agrovoc;i_24887"/>
  <hasLexicalization rdf:resource="&agrovoc;i_en_1299486360046"/>
  <hasLexicalization rdf:resource="&agrovoc;i_es_1299486360065"/>
  <hasLexicalization rdf:resource="&agrovoc;i_en_1299486361157"/>
</owl:NamedIndividual>

<!-- http://aims.fao.org/aos/agrovoc/i_en_1299486360046 -->
<owl:NamedIndividual rdf:about="&agrovoc;i_en_1299486360046">
  <rdf:type rdf:resource="http://aims.fao.org/aos/common/c_noun"/>
  <rdfs:label xml:lang="en">products</rdfs:label>
  <hasCodeAgrovoc rdf:datatype="&xsd:int">6211</hasCodeAgrovoc>
  <isMainLabel rdf:datatype="&xsd:boolean">true</isMainLabel>
  <isLexicalizationOf rdf:resource="&agrovoc;i_6211"/>
  <hasNearSynonym rdf:resource="&agrovoc;i_en_1299486361157"/>
</owl:NamedIndividual>

<!-- http://aims.fao.org/aos/agrovoc/i_es_1299486360065 -->
<owl:NamedIndividual rdf:about="&agrovoc;i_es_1299486360065">
  <rdf:type rdf:resource="http://aims.fao.org/aos/common/c_noun"/>
  <rdfs:label xml:lang="es">Productos</rdfs:label>
  <hasCodeAgrovoc rdf:datatype="&xsd:int">6211</hasCodeAgrovoc>
  <isMainLabel rdf:datatype="&xsd:boolean">true</isMainLabel>
  <isLexicalizationOf rdf:resource="&agrovoc;i_6211"/>
</owl:NamedIndividual>

<!-- http://aims.fao.org/aos/agrovoc/i_en_1299486361157 -->
<owl:NamedIndividual rdf:about="&agrovoc;i_en_1299486361157">
  <rdf:type rdf:resource="http://aims.fao.org/aos/common/c_noun"/>
  <rdfs:label xml:lang="en">produce</rdfs:label>
  <hasCodeAgrovoc rdf:datatype="&xsd:int">13588</hasCodeAgrovoc>
  <isMainLabel rdf:datatype="&xsd:boolean">false</isMainLabel>
  <isLexicalizationOf rdf:resource="&agrovoc;i_6211"/>
  <hasNearSynonym rdf:resource="&agrovoc;i_en_1299486360046"/>
</owl:NamedIndividual>

```

Como se puede apreciar en el ejemplo de la figura 6.5, todas las clases OWL de AGROVOC aparecen como subclases directas de la clase OWL: Thing, por lo tanto, las clases descendientes de Producto aparecen repetidas (en la imagen se marca Productos agrícolas como ejemplo).



Figura 6.5. Segunda versión del concepto Productos de AGROVOC desplegado en Protégé.

Esto dificultaría el razonamiento más adelante, así que es necesaria una nueva modificación de las clases. Como los conceptos de primer nivel de la ontología son subclases directas de Concepto de dominio (ver figura 6.6), es posible distinguir las clases de primer nivel en la jerarquía de las demás.

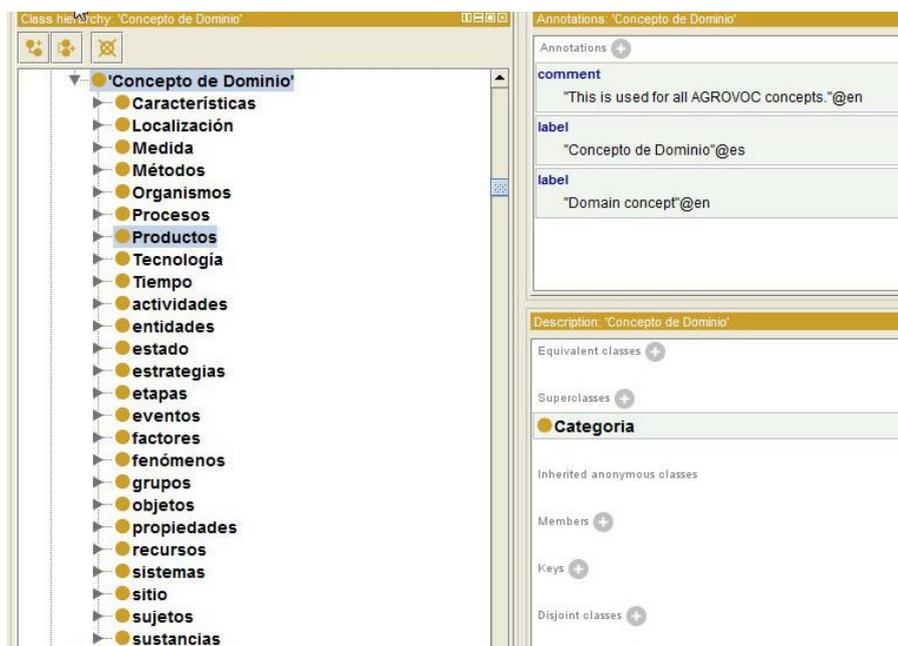


Figura 6.6. Concepto de Dominio AGROVOC y sus subclases directas en Protégé.

Una vez modificadas las clases prescindiendo de sus elementos `subClassOf` no necesarios, es posible la visualización de la ontología en el primer nivel jerárquico desde Protégé, como se muestra en la figura 6.7.



Figura 6.7. AGROVOC desplegada en su primer nivel.

En esta nueva versión, el concepto `Producto` resultaría en la clase OWL siguiente:

```
<owl:Class rdf:about="&agrovoc;c_6211">
  <rdfs:subClassOf rdf:resource="&owl;Thing"/>
  <rdfs:label xml:lang="en">Products</rdfs:label>
  <rdfs:label xml:lang="es">Productos</rdfs:label>
</owl:Class>
```

Toda la información adicional no necesaria se eliminó sabiendo que siempre estará disponible accediendo a AGROVOC mediante los identificadores de los conceptos principales que sí se han mantenido. En este momento, AGROVOC ya está preparada para que sirva como punto de partida para el desarrollo de nuestra propia ontología de dominio.

En principio, todos los conceptos que van a formar parte de la ontología nutricional son descendientes del concepto “Productos” de AGROVOC o están de algún modo relacionados con él. Por tanto, el primer paso ha consistido en la extracción de la rama “Productos” mediante una operación de poda que extrae los elementos indicados de una ontología para construir otra, utilizando la librería Java OWL API. El tamaño de la ontología en este momento ha quedado reducido a 9MB.

A continuación, se explica el desarrollo de nuestra ontología de dominio a partir de la versión reducida de la rama “Productos” de AGROVOC. En la figura 6.8 se muestra su subclase “Alimentos” desplegada.

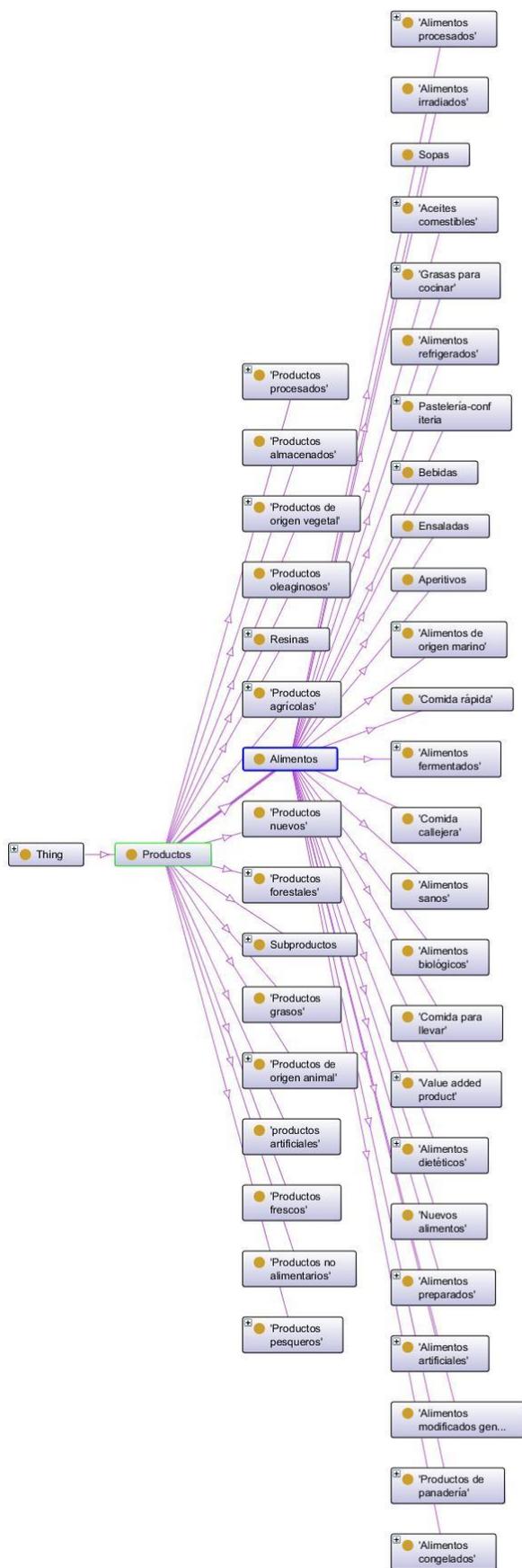


Figura 6.8. Representación gráfica de Productos AGROVOC. Subclase Alimentos desplegada.

Definición de clases y jerarquía de clases de la ontología nutricional.

A la hora de definir las clases de la ontología usamos como base la versión de AGROVOC reducida y podada en su rama “Productos” (figura 6.8), puesto que sus descendientes son equivalentes o relacionados con los conceptos nutricionales que forman parte del modelo conceptual de NutElCare. Siguiendo la sintaxis de los IRIs de los conceptos AGROVOC, se han definido los conceptos NutElCare, donde todo concepto se representa mediante un IRI con la siguiente sintaxis⁶⁰:

```
http://asistic.ugr.es/nutelcare/concept_name
```

Por tanto, el espacio de nombres de la ontología está definido como:

```
xmlns:nutelcare="http://asistic.ugr.es/nutelcare/"
```

y su prefijo en el documento OWL: &NUTELCARE;.

El motivo de utilizar una IRI propia, en lugar de reutilizar la de AGROVOC es facilitar posteriormente el tratamiento homogéneo de la ontología desde OWL API.

En el paso 3 se enumeraron los conceptos más importantes de la ontología a partir de la información recogida en la fase de elicitación de conocimiento. En este paso se decide cuáles de esos conceptos serán formalizados mediante clases OWL. Durante este proceso se realizan las siguientes operaciones básicas (recordar que la construcción de una ontología es un proceso iterativo donde varios pasos pueden repetirse sucesiva y alternativamente):

- **Eliminación de clases innecesarias.** Desde Protégé se pueden eliminar las clases que no interesan para sistema propuesto. Siempre antes de cada eliminación, es necesario comprobar la aparición de ese mismo concepto en otras partes de la ontología, ya que si ese concepto es superclase de otra/s clase/s que no se desea borrar y se eliminara, esa/s subclase/s se perdería o se podría situar en lugares indeseados de la jerarquía. En la figura 6.9 se muestra un ejemplo donde se pretende eliminar la clase C, que tiene como subclases a D y E, pero E es a su vez subclase de la clase G. Al borrar C, Protégé nos dará dos opciones: eliminar sólo C, o eliminar C y sus descendientes. Escogiendo la primera opción, la ontología se reestructura y resulta en la figura 6.9 (a) donde las subclases de C que se quedan sin superclase (D) aparecen como subclases de Thing y eligiendo la segunda opción resulta en la figura 6.9 (b), donde E ha desaparecido completamente de la ontología y D y sus subclases también han sido eliminadas.

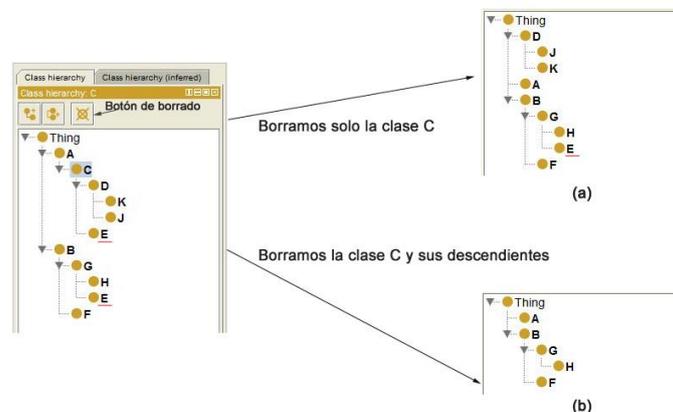


Figura 6.9. Ejemplo de eliminación de clases en Protégé.

⁶⁰ La ontología NutElCare está accesible en <http://asistic.ugr.es/nutelcare/nutelcare.owl>

- Adición de nuevas clases.** Lógicamente, es necesario añadir conceptos nuevos a la ontología, ya que no todos los que se van a utilizar se encuentran en AGROVOC. La adición de conceptos se realiza también manualmente en Protégé. Para todas las nuevas clases se han incluido también sus anotaciones correspondientes, tanto comentarios como etiquetas, en español e inglés. Reiterar que todas las adiciones, eliminaciones y modificaciones sobre los conceptos de la ontología, han estado siempre guiadas por los expertos en nutrición que colaboran en este trabajo. Se cita a continuación la definición OWL (recordar que nos referimos a OWL 2) de algunas de las nuevas clases mostrando, para resumir, únicamente su anotación de etiqueta (`rdfs:label`):

```

<owl:Class rdf:about="&NUTELCARE;dailyIntake">
  <rdfs:label xml:lang="en">Daily intake</rdfs:label>
  <rdfs:label xml:lang="es">Ingesta diaria</rdfs:label>
</Class>
<owl:Class rdf:about="&NUTELCARE;intakeDish">
  <rdfs:label xml:lang="en">Intake dish</rdfs:label>
  <rdfs:label xml:lang="es">Plato de ingesta</rdfs:label>
</owl:Class>
<owl:Class rdf:about="&NUTELCARE;homeServing">
  <rdfs:label xml:lang="en">Home serving</rdfs:label>
  <rdfs:label xml:lang="es">Ración casera</rdfs:label>
</owl:Class>
<Class rdf:about="&NUTELCARE;cookingStyle">
  <rdfs:label xml:lang="en">Cooking style</rdfs:label>
  <rdfs:label xml:lang="es">Modo de preparacion</rdfs:label>
</Class>
<Class rdf:about="&NUTELCARE;fatSolubleVitamins">
  <rdfs:label xml:lang="en">Fat Soluble Vitamins</rdfs:label>
  <rdfs:label xml:lang="es">Vitaminas Liposolubles</rdfs:label>
</Class>
<Class rdf:about="&NUTELCARE;waterSolubleVitamins">
  <rdfs:label xml:lang="en">Water Soluble Vitamins</rdfs:label>
  <rdfs:label xml:lang="es">Vitaminas Hidrosolubles</rdfs:label>
</Class>
<Class rdf:about="&NUTELCARE;energyFunction ">
  <rdfs:label xml:lang="en">Energy function</rdfs:label>
  <rdfs:label xml:lang="es">Función de energía</rdfs:label>
</Class>
<Class rdf:about="&NUTELCARE;constructionFunction ">
  <rdfs:label xml:lang="en">Construction function</rdfs:label>
  <rdfs:label xml:lang="es">Función de construcción</rdfs:label>
</Class>
<Class rdf:about="&NUTELCARE;regulationFunction ">
  <rdfs:label xml:lang="en">Regulation function</rdfs:label>
  <rdfs:label xml:lang="es">Función de regulación</rdfs:label>
</Class>

```

- Definición de la jerarquía de clases.** Como AGROVOC no está orientada a nutrición, su jerarquía de conceptos no se adapta bien a las necesidades de NutElCare. Por este motivo, se tuvieron que llevar a cabo algunos cambios en la misma, mediante la opción de Protégé que facilita mover nodos (con sus subnodos). Una de las primeras, y más importantes, modificaciones en la estructura ha consistido en el reordenamiento jerárquico de los alimentos considerados como básicos, según su función en el organismo, es decir, regulación, energía y construcción (ver figura 6.3), en lugar de cómo se encontraban clasificadas inicialmente, que era considerando únicamente si son origen animal o vegetal. Se muestra en la figura 6.10 la nueva estructura taxonómica, donde aparecen las nuevas clases “Función de Regulación”, “Función de Energía” y “Función de Construcción”, y se observa además que el primer nivel de la jerarquía separa los conceptos en dos grandes ramas: “Alimentos” (que contendrá los alimentos que el sistema recomendará) y “Nutrición” (que contendrá los conceptos relacionados con nutrición necesarios en el razonamiento, por ejemplo “Principales Nutrientes”)⁶¹.



Figure 6.10. Nueva jerarquía de la clase “Alimentos” en la ontología nutricional.

PASO 5 - Definir las propiedades (también llamadas relaciones o *slots*) de las clases. La potencia del razonamiento en el uso de ontologías se produce gracias a las relaciones semánticas entre los conceptos que la forman. Una vez creadas las clases y sus relaciones jerárquicas en la

⁶¹ Hasta ahora, en las figuras se han mostrado los términos de AGROVOC con la etiqueta de clase Protégé en español. A partir de este momento, los términos de la nueva ontología se mostrarán con la etiqueta en inglés para mantener la coherencia con las publicaciones internacionales realizadas en este trabajo.

ontología han de definirse las propiedades semánticas de las clases en el dominio. Del listado de los conceptos más importantes en el dominio (paso 3) se seleccionaron aquellos que sirven para dotar a las clases de dichas propiedades semánticas y que no aparecían en AGROVOC. Estas propiedades, que también pueden relacionarse jerárquicamente entre ellas, son las que van a aportar el nuevo valor nutricional a los conceptos de la ontología nutricional, ya que van a permitir el razonamiento sobre mediciones y los diversos factores nutricionales.

En primer lugar, se definieron las Propiedades de Datos (*Data Properties*). En la figura 6.11 se muestran parte de las mismas junto con su jerarquía.

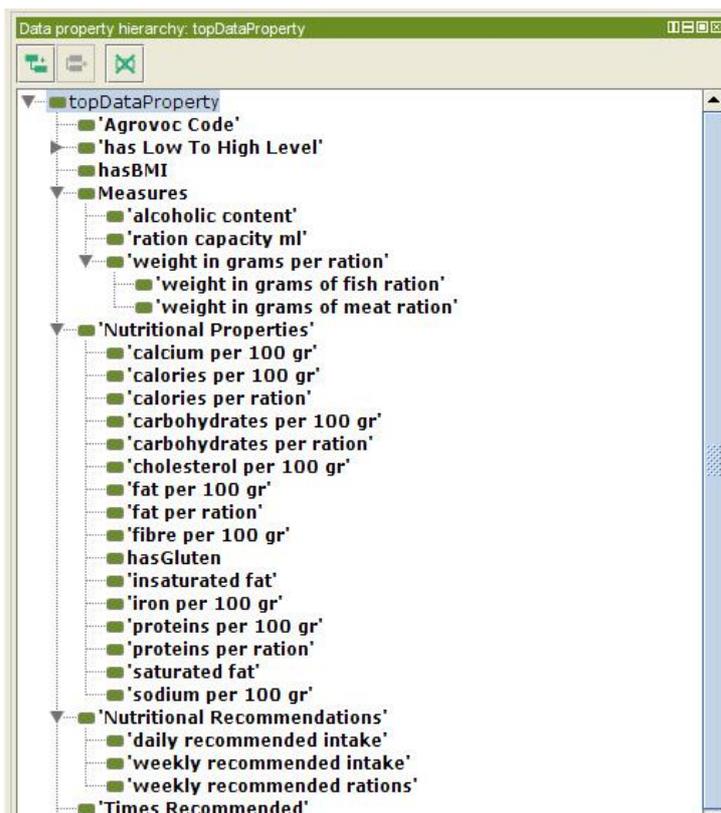


Figura 6.11. Extracto de las propiedades de tipo de datos de la ontología nutricional.

Además, pueden establecerse ciertas restricciones sobre el rango y el dominio de los slots, así como la cardinalidad que deban tener alguno de ellos. Por ejemplo, la propiedad “alcoholic content” tiene como dominio la clase “Alcoholic Beverages” y su rango es la definición del tipo de dato que en este caso es de tipo “float”. Todas las propiedades nutricionales se definen como funcionales, puesto que sólo pueden tomar un valor.

A continuación, se definieron las Propiedades de Objeto (*Object Properties*) que relacionan semánticamente dos clases de la ontología. En la figura 6.12 se muestra una parte de ellas y su jerarquía. En este caso, también puede restringirse las relaciones por dominio y rango. Por ejemplo, el dominio de la relación “cooked as” es la clase “Food Products” y el rango es la clase “Cooking Style”, indicando que un alimento está preparado mediante un modo de preparación. La definición de esta restricción en OWL sería:

```
<owl:ObjectProperty rdf:about="&NUTELCARE;cookedAs">
  <rdfs:label xml:lang="en">Cooked as</rdfs:label>
  <rdfs:label xml:lang="es">Cocinado como</rdfs:label>
```

```

<rdfs:domain rdf:resource="&NUTELCARE;Foods"/>
<rdfs:range rdf:resource="&NUTELCARE;CookingStyle"/>
</owl:ObjectProperty>

```

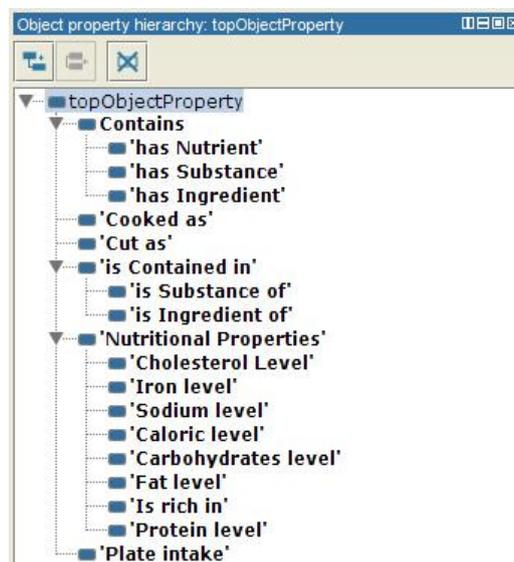


Figura 6.12. Algunas Propiedades de Objeto de la ontología nutricional.

En OWL se pueden caracterizar las propiedades mediante relaciones de equivalencia y relaciones inversas. Como ejemplo de propiedades en las que se ha definido una relación inversa encontramos “has Ingredient” y “is Ingredient Of”, indicando que si un plato tiene como ingrediente cierto alimento, el alimento, a su vez, es ingrediente de ese plato. Ambas propiedades son a su vez descendientes de “Contains” e “is Contained in” respectivamente. Así, si un alimento es ingrediente de un plato, puede inferirse que ese plato *contiene* ese ingrediente. A su vez, ambas son propiedades transitivas, ya que si un plato tiene un ingrediente X que a su vez contiene otro ingrediente Y, se infiere que el plato contiene al ingrediente Y. Como ejemplo, mostramos a continuación la definición OWL del slot o relación “has Ingredient”:

```

<owl:ObjectProperty rdf:about="&NUTELCARE;hasIngredient">
  <rdf:type rdf:resource="http://www.w3.org/2002/07/owl#TransitiveProperty" />
  <rdfs:label xml:lang="en">has Ingredient</rdfs:label>
  <rdfs:label xml:lang="es">tiene Ingrediente</rdfs:label>
  <owl:inverseOf rdf:resource="&NUTELCARE;isIngredientOf"/>
  <rdfs:subPropertyOf rdf:resource="&NUTELCARE;contains"/>
</owl:ObjectProperty>

```

PASO 7 – Definición de instancias. En este paso, se definieron las instancias de cada clase. Una parte significativa de ellas, aproximadamente el 75%, se encontraban ya incluidas en la ontología porque gran parte de esas instancias se obtuvieron de AGROVOC (actualmente la ontología tiene más de 800 instancias). Algunos ejemplos de instancias incluidas son:

- Alimentos que no estaban en AGROVOC (como algunos embutidos).
- Instancias de la clase “Meat Cut”: Lomo, Pierna, Filete, Pechuga, etc.

6.3.2 Creación de la Ontología de Dominio. Enfoque Bottom-up.

Desde la rama “Productos” de AGROVOC en su versión reducida se ha seguido un enfoque *top-down* para la definición de la ontología de aplicación, partiendo desde la definición de clases hasta la incorporación de instancias. A partir de aquí, el proceso de creación de la ontología se llevó a cabo siguiendo un enfoque *bottom-up* para completar el conocimiento utilizando las instancias en los siguientes pasos del proceso de representación.

Inicialmente, a cada instancia descendiente de la rama alimentos se le añaden sus características nutricionales mediante la instanciación de sus propiedades de datos y de objeto nutricionales. En la figura 6.13 se presenta a modo de ejemplo el caso de la instancia “Lemon”.

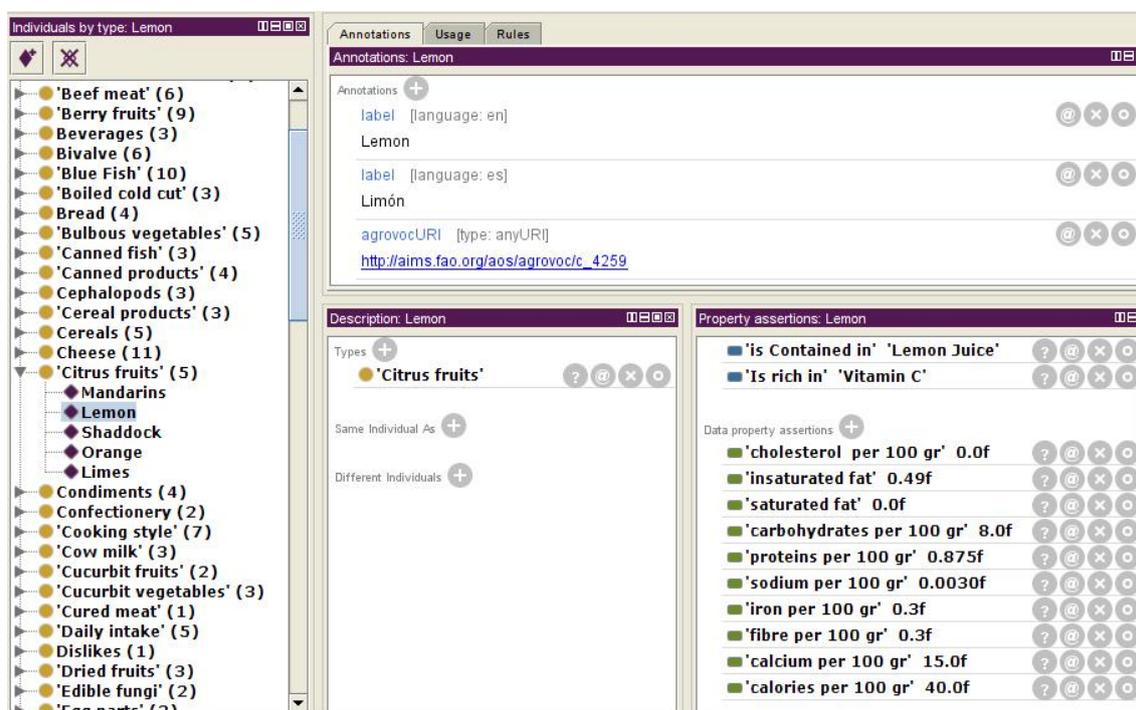


Figura 6.13 Representación de las propiedades nutricionales de la instancia “Lemon”.

Un ejemplo de restricción añadida es la limitación del número de raciones recomendadas semanalmente de los alimentos básicos de “función de construcción” (carne, pescado y huevos). Según las recomendaciones nutricionales debemos introducir las siguientes restricciones:

- Número recomendado de raciones de carne a la semana. Máximo 4, mínimo 3.
- Número recomendado de raciones de pescado a la semana. Máximo 4, mínimo 3.
- Número recomendado de raciones de huevo a la semana. Máximo 4, mínimo 3.

En este caso, se añadieron 6 restricciones de *propiedad objeto* a la clase “Recommended intake” (figura 6.14). En OWL quedaría expresado como (mostramos sólo la restricción para pescado):

```
<owl:Class rdf:about="&NUTELCARE;RecommendedIntake">
  <rdfs:label xml:lang="en">Recommended intake</rdfs:label>
  <rdfs:label xml:lang="es">Consumo recomendado</rdfs:label>
  <rdfs:subClassOf rdf:resource="&NUTELCARE;Nutrition"/><rdfs:subClassOf>
  <owl:Restriction>
```

```

<owl:onProperty rdf:resource="&NUTELCARE;RecommendedRationsAWeek"/>
<owl:onClass rdf:resource="&NUTELCARE;FishTypes"/>
<owl:maxQualifiedCardinality rdf:datatype="http://www.w3.org/2001/XMLSchema#nonNegativeInteger">4</owl:maxQualifiedCardinality>
</owl:Restriction>
</rdfs:subClassOf>
...
</owl:Class>

```

The screenshot displays a web-based ontology editor interface. On the left, a 'Class hierarchy' tree shows a path from 'Thing' to 'Nutrition' to 'Recommended intake'. The right side of the interface is divided into two main sections: 'Annotations' and 'Usage'. The 'Annotations' section shows two labels: 'Recommended intake' in English and 'Consumo recomendado' in Spanish, along with an 'agrovocURI' pointing to a specific resource. The 'Usage' section, titled 'Description: Recommended intake', lists several 'Equivalent To' and 'SubClass Of' relationships, including restrictions on 'Recommended rations a week' for 'Fish types', 'Fresh meat', and 'Eggs'.

Figura 6.14. Restricciones de propiedad sobre “Recommended intake” para delimitar las raciones semanales recomendadas.

Otro tipo de restricciones se han definido mediante relaciones de equivalencia. Por ejemplo, mediante la siguiente relación de equivalencia sobre la clase “Cereals” indicamos que los cereales contienen gluten:

EquivalentTo Contains value Gluten

Resultando así en OWL:

```

<owl:Class rdf:about="&NUTELCARE;Cereals">
  <rdfs:label xml:lang="en">Cereals</rdfs:label>
  <rdfs:label xml:lang="es">Cereales</rdfs:label>
  <owl:equivalentClass>
    <owl:Restriction>
      <owl:onProperty rdf:resource="&NUTELCARE;contains"/>

```

```

        <owl:hasValue rdf:resource="&NUTELCARE;Gluten"/>
    </owl:Restriction>
</owl:equivalentClass>
    <rdfs:subClassOf rdf:resource="&NUTELCARE;Funcion_de_energia"/>
</owl:Class>

```

De este modo se inferirá que todas las instancias descendientes de la clase “Cereals” contienen “Gluten”.

6.3.3 Representación Ontológica de Dietas

Las ontologías se usan también en el área de recuperación de información para indexar documentos, proporcionando una clasificación semántica al contenido de los documentos (Ghawi and Cullot, 2009). Puesto que las dietas a introducir en el sistema venían dadas mediante archivos de texto (ver Anexo A), las mismas pueden ser clasificadas en la ontología como descendientes de la clase “Diet Types”. Sin embargo, esta clasificación no es suficiente para poder razonar con los contenidos de las dietas. Con el fin de permitir el razonamiento sobre los contenidos de una dieta en el sistema, se decidió transferir dichos contenidos a la ontología. Para ello, se definió un proceso de transformación de las dietas que permitiera obtener un modelo ontológico de las mismas. En primer lugar, se obtuvo el modelo sintáctico de las dietas mediante anotaciones XML sobre todos los documentos del repositorio de dietas. La definición del metamodelo de dicho modelo sintáctico se especificó mediante un documento XSD (XML Schema) que asegura salvaguardar el modelo sintáctico en el sistema proporcionando además validación de *parseo* durante el proceso de anotación.

Una representación gráfica del modelo XSL completo de las dietas puede verse en la figura 6.15. Como se indica en la figura, una dieta se divide en diferentes días (“dayOfDiet”), que a su vez se componen de varias ingestas diarias, como desayuno, almuerzo, etc. (“dailyIntake”). Cada ingesta está formada por diferentes platos, como entrantes, primeros, postres, etc. (“Meal”) y cada plato tiene varios alimentos (“FoodItem”).

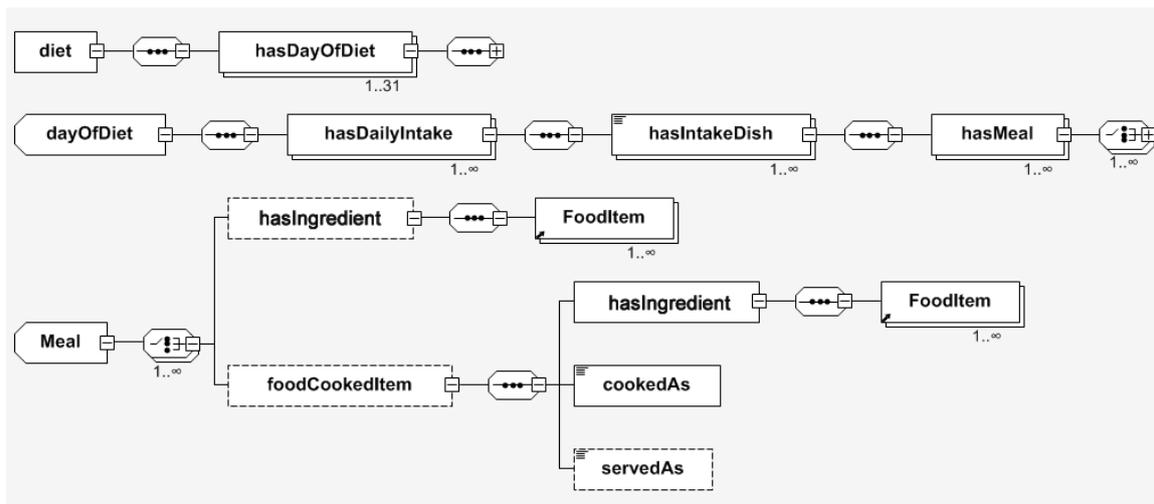


Figura 6.15. Modelo gráfico del XML Schema de una Dieta mediante XSDDiagram⁶².

En la figura 6.16 se presenta un extracto de la definición del XML Schema de una dieta, en concreto, la definición del elemento “Meal” correspondiente a un alimento de la misma. Es

⁶² XSDDiagram: <http://regis.cosnier.free.fr/?page=XSDDiagram>

importante notar que el elemento `FoodItem` de la figura 6.14 referencia a una instancia de la clase “Food Products” (IRI: <http://asisttic.ugr.es/nutelcare/Foods>) de la ontología nutricional. Dicho enlace se lleva a cabo mediante el atributo `ref="&nutelcare:Foods"` (como se aprecia en la figura 6.15), por lo que su espacio de nombres correspondiente debe ser importado al principio del documento incluyendo la siguiente línea:

```
<xs:import namespace="http://asisttic.ugr.es/nutelcare#" schemaLocation="nutelcare.xsd"/>
```

Así, los alimentos de las dietas se enlazan a las instancias de alimentos de la ontología.

```

▼<xs:complexType name="Meal">
  ▼<xs:choice maxOccurs="unbounded">
    ▼<xs:element minOccurs="0" name="hasIngredient">
      ▼<xs:complexType>
        ▼<xs:sequence>
          <xs:element maxOccurs="unbounded" minOccurs="1" ref="nutelcare:Foods"/>
        </xs:sequence>
        <xs:attribute name="value" type="foodItemType" use="required"/>
      </xs:complexType>
    </xs:element>
    ▼<xs:element minOccurs="0" name="foodCookedItem">
      ▼<xs:complexType>
        ▼<xs:sequence>
          ▼<xs:element minOccurs="1" name="hasIngredient">
            ▼<xs:complexType>
              ▼<xs:sequence>
                <xs:element maxOccurs="unbounded" minOccurs="1" ref="nutelcare:Foods"/>
              </xs:sequence>
              <xs:attribute name="value" type="foodItemType" use="required"/>
            </xs:complexType>
          </xs:element>
          ▼<xs:element minOccurs="1" name="cookedAs">
            ▼<xs:complexType>
              ▼<xs:simpleContent>
                ▼<xs:extension base="xs:string">
                  <xs:attribute name="value" type="cookingStyle" use="required"/>
                  <xs:attribute name="label_es" type="xs:string" use="required"/>
                </xs:extension>
              </xs:simpleContent>
            </xs:complexType>
          </xs:element>
          ▼<xs:element minOccurs="0" name="servedAs">
            ▼<xs:complexType>
              ▼<xs:simpleContent>
                ▼<xs:extension base="xs:string">
                  <xs:attribute name="value" type="homeServing" use="required"/>
                  <xs:attribute name="label_es" type="xs:string" use="required"/>
                </xs:extension>
              </xs:simpleContent>
            </xs:complexType>
          </xs:element>
        </xs:sequence>
      </xs:complexType>
    </xs:element>
  </xs:choice>
  <xs:attribute name="value" type="mealType" use="required"/>
  <xs:attribute name="label_es" type="xs:string" use="required"/>

```

Figura 6.16. Definición XML Schema del elemento “Meal” de una dieta.

Una vez definido el XML Schema se anotan las dietas para obtener su modelo sintáctico. Un ejemplo de anotación del *almuerzo* de una dieta se muestra en la figura 6.17.

El proceso consiste en un enriquecimiento de la ontología de dominio con los conceptos que componen el modelo sintáctico de las dietas. Con este fin, se establecen los siguientes pasos: definir reglas de mapeo que constituyan la relación entre los elementos de XML-Schema y axiomas OWL, especificar una transformación XSLT para construir un modelo OWL a partir del modelo sintáctico y, finalmente, añadir dicho modelo OWL a la ontología de dominio, adquiriendo todo el potencial semántico y de razonamiento de dicha ontología.

El mapeo entre los elementos XML Schema y los conceptos OWL se establece mediante una transformación XSLT que utiliza una hoja de estilos XSL siguiendo las reglas resumidas en la Tabla 6.1. Destacar que estas reglas están definidas siguiendo el propósito de nuestra aplicación y la representación de la ontología de dominio, por lo que difiere en algunos aspectos de los propuestos por otros autores en sus trabajos (Ferdinand et al., 2004), (Bohring and Auer, 2005), (Hacherouf et al., 2015).

Tabla 6.1. Reglas de mapeo entre los nodos del modelo de dieta XSD definido a conceptos OWL.

Nodos XSD	Conceptos OWL
xsd:element, with nested elements or at least one attribute	owl:Class + owl:ObjectProperty
other xsd:element	owl:ObjectProperty
named xs:complexType	owl:Class
named xs:simpleType	owl:DatatypeProperty
xs:minOccurs, xs:maxOccurs	owl:minCardinality, owl:maxCardinality
xs:choice	owl:unionOf
xs:sequence	owl:intersectionOf

Para llevar a cabo la transformación XSLT se utiliza el código JSP de la figura 6.19. En ella se observa que inicialmente se importa la ontología y la hoja de estilos definida XSL.

```
<%@ page language="java" contentType="text/html; charset=ISO-8859-1"
pageEncoding="ISO-8859-1"%>
<%@ taglib uri="http://java.sun.com/jsp/jstl/xml" prefix="x" %>
<%@ taglib uri="http://java.sun.com/jsp/jstl/core" prefix="c" %>
<c:import url="http://localhost:8080/NutElCare/diets/diet_schema.xsd"
var="inputDoc" />

<c:import url="http://localhost:8080/NutElCare/diets/transformDiets.xsl"
var="stylesheet" />

<c:set var="doc" scope="page">
<x:transform xml = "${inputDoc}" xslt = "${stylesheet}" />
</c:set>
<x:transform xml = "${inputDoc}" xslt = "${stylesheet}" />
```

Figura 6.19. Página JSP que realiza la transformación XSLT.

Un extracto de la definición de la hoja de estilos XSL puede verse en la figura 6.20.

```

<xsl:template name="elements" match="xs:element">
  <xsl:if test="./@name">
    <xsl:element name="owl:ObjectProperty">
      <xsl:attribute name="rdf:about">
        <xsl:text disable-output-escaping="yes">&nutelcare;</xsl:text><xsl:value-of select="./@name"/>
      </xsl:attribute>
      <xsl:if test="./@type">
        <xsl:element name="rdfs:range">
          <xsl:attribute name="rdf:resource">
            <xsl:text disable-output-escaping="yes">&nutelcare;</xsl:text><xsl:value-of select="./@type"/>
          </xsl:attribute>
        </xsl:element>
      </xsl:if>
    </xsl:element>
  </xsl:if>
  <xsl:if test="./@ref">
    <xsl:element name="owl:Class">
      <xsl:attribute name="rdf:about">
        <xsl:text disable-output-escaping="yes">&nutelcare;</xsl:text><xsl:value-of select="substring-after(./@ref,':')"/>
      </xsl:attribute>
    </xsl:element>
  </xsl:if>
</xsl:template>

```

Figura 6.20. Extracto de la hoja de estilos XSL para la definición del mapeo de reglas.

Aplicando la transformación se obtiene un archivo con el modelo semántico en OWL; como ejemplo del modelo semántico resultante, en la figura 6.21 se muestra el fragmento que corresponde a la clase “Meal”.

```

<!-- http://asistic.ugr.es/nutelcare/Meal -->
<owl:Class rdf:about="&nutelcare;Meal">
  <owl:equivalentClass>
    <owl:Class>
      <owl:unionOf rdf:parseType="Collection">
        <owl:Restriction>
          <owl:onProperty rdf:resource="&nutelcare;hasFoodCookedItem"/>
          <owl:someValuesFrom rdf:resource="&nutelcare;foodCookedItem"/>
        </owl:Restriction>
        <owl:Restriction>
          <owl:onProperty rdf:resource="&nutelcare;hasFoodItem"/>
          <owl:someValuesFrom rdf:resource="&nutelcare;Foods"/>
        </owl:Restriction>
      </owl:unionOf>
    </owl:Class>
  </owl:equivalentClass>
  <rdfs:subClassOf rdf:resource="&nutelcare;dietComposition"/>
  <owl:ObjectProperty></owl:ObjectProperty>
</owl:Class>

```

Figura 6.21. Resultado de aplicación de la transformación XSLT sobre el nodo “Meal” a clase OWL.

Finalmente, el modelo OWL obtenido se importó a la ontología de dominio utilizando como nodo raíz la clase “Diets Composition”. En la figura 6.22 puede verse el resultado de dicha importación mostrando como ejemplo la definición de la clase “Meal” y en negrita todos los conceptos utilizados en la definición de una dieta.

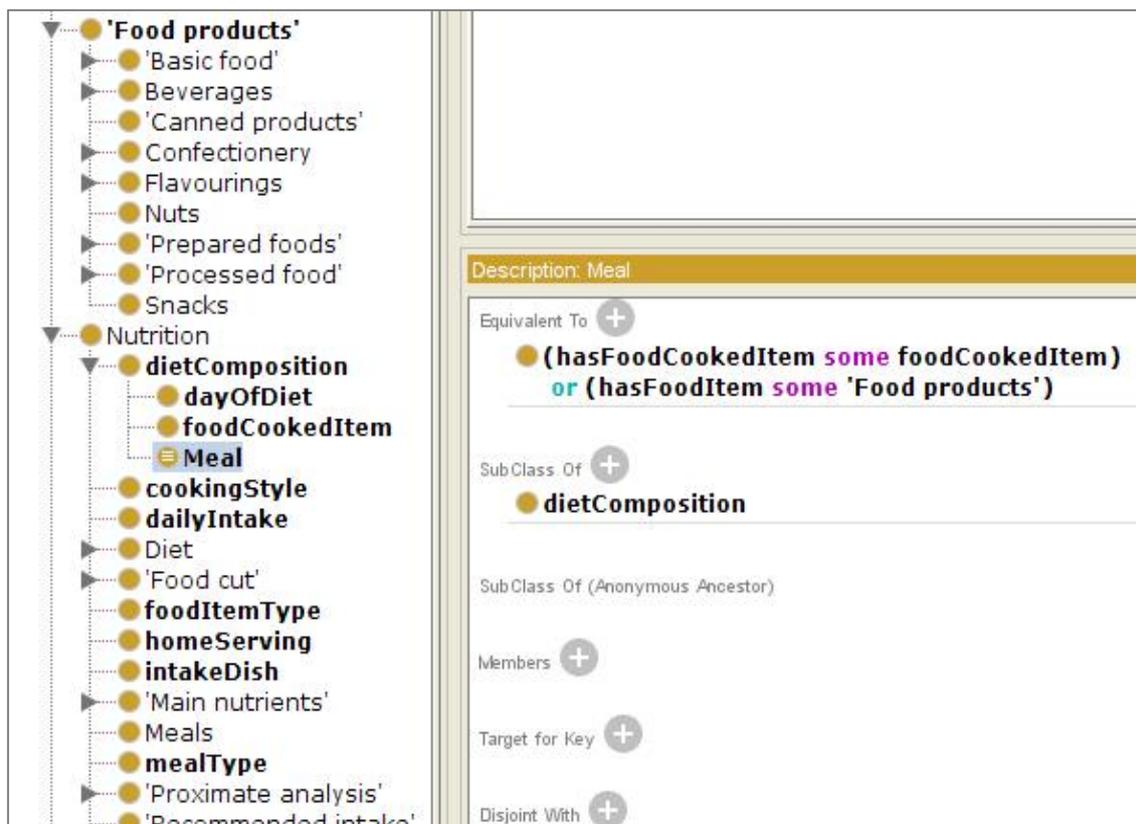


Figura 6.22. Resultado del enriquecimiento de la ontología mediante la transformación XSLT del modelo sintáctico XML de las dietas.

6.4. Adquisición de Conocimiento para la Representación del Perfil de Usuario

Una vez construida la ontología de dominio nutricional y tras el proceso de enriquecimiento con la definición de dietas propuesto, es necesario establecer qué características del usuario deberán representarse en su perfil de usuario y cómo será el proceso para obtenerlas.

Cuando un usuario accede al sistema, lo hace con la finalidad de obtener una dieta adecuada a sus características y que éste le ofrezca la posibilidad de modificar esa dieta, de modo que, sin perder el valor nutritivo y saludable, pueda ajustarse mejor a sus intereses individuales y resultar más de su agrado. Para ello, este sistema de recomendación debe conocer, no sólo las características físicas y demográficas del usuario, sino también cuáles son sus preferencias. Por este motivo, la primera vez que el usuario se conecta a NutElCare, durante el proceso de registro, debe cumplimentar un formulario diseñado para aportar información sobre sí mismo, que es relevante para el funcionamiento del sistema.

En esta sección se ha realizado una revisión y análisis de los requisitos relevantes para los procesos de recomendación de dietas y alimentos a partir de la información aportada por los expertos. Recordemos que en la estrategia de recomendación propuesta en el capítulo 5 se definió una primera fase de recomendación en la cual se obtiene un tipo de dieta adecuado a los requisitos nutricionales del usuario. Dichos requisitos se especifican como:

▪ **Factores físicos:**

- El peso y la altura del usuario. Se obtienen del formulario inicial y a partir de ellos se calcula el Índice de Masa Corporal (IMC) según la ecuación (1):

$$IMC = \frac{Peso (kg)}{Altura (m)^2} \quad (11)$$

No existe un criterio uniforme para delimitar los intervalos de *bajopeso* y *sobrepeso* según los valores del IMC. Se tienden a aceptar como puntos de corte para definir el peso insuficiente, valores del IMC menores de 18 y para la obesidad, valores del IMC mayores de 30. La OMS (Organización Mundial de la Salud) y la SEEDO (Sociedad Española para el Estudio de la Obesidad) han propuesto una clasificación del estado nutricional basado en el IMC. En el documento de consenso de la SENPE (Sociedad Española de Nutrición Parenteral y Enteral) sobre valoración nutricional en el anciano (SENPE and SEGG, 2007) se muestra una clasificación para el estado nutricional del anciano según su IMC. En la Tabla 6.2 se recogen las clasificaciones citadas.

Tabla 6.2. Valoración del Estado Nutricional según IMC. Fuente: (SENPE and SEGG, 2007).

ÍNDICE DE MASA CORPORAL (IMC) = PESO / TALLA ²			
Valoración Nutricional	OMS	SEEDO	Ancianos
Desnutrición			<18.4 kg/m ²
Peso insuficiente	<18.5 kg/m ²	<18.5	18.5 – 22 kg/m ²
Normopeso	18.5 - 24.9 kg/m ²	18,5 - 21,9 kg/m ²	22 - 19,9 kg/m ²
Sobrepeso	25 - 29,9 kg/m ²	25 - 26,9 kg/m ²	27 - 29,9 kg/m ²
Obesidad	>30 kg/m ²	>30 kg/m ²	>30 kg/m ²

Partiendo de las mismas y tomando como umbral la edad de 75 años para la aplicación de la columna *Ancianos*, definimos la siguiente clasificación:

Si el usuario es menor de 75 años (siguiendo el criterio de la OMS):

$$IMC (u_{edad < 75}) = \begin{cases} < 18.5 & \text{bajo - peso} \\ 18.5 - 24.9 & \text{normo - peso} \\ 25 - 29.9 & \text{sobre - peso} \\ > 30 & \text{obesidad} \end{cases} \quad (12)$$

Si el usuario es mayor de 75 años (siguiendo el criterio de la SEEDO):

$$IMC (u_{edad > 75}) = \begin{cases} < 18.5 & \text{desnutrición} \\ 18.5 - 21.9 & \text{bajo - peso} \\ 22 - 26.9 & \text{normo - peso} \\ 27 - 29.9 & \text{sobre - peso} \\ > 30 & \text{obesidad} \end{cases} \quad (13)$$

Si el usuario se clasifica en la categoría *desnutrición* u *obesidad*, el sistema le recomienda acudir al médico para una dieta más estricta. Por tanto, las dietas para un

usuario con obesidad o desnutrición quedan excluidas de este trabajo. Lo mismo ocurre si se detecta alguno de estos casos en la aplicación de los métodos de *screening* durante el proceso de uso del sistema.

- Nivel de deglución y masticación. Para distinguir si el usuario necesita dietas normales, blandas, semi-líquidas o de fácil digestión.
- **Factores ambientales o de contexto.** Época del año y entorno geográfico en que se realizan las recomendaciones. Se distinguirán entre dietas de primavera-verano y otoño-invierno. El entorno geográfico se utiliza para determinar el tipo de clima de la zona en que se encuentra el usuario, como clima seco o húmedo.
- **Factor de actividad.** Es necesario conocer la cantidad de ejercicio que el usuario realiza en su vida diaria. Se distingue entre:

$$AF = \begin{cases} > 3h & \text{activo} \\ 1 - 3h \text{ (tareas hogar y paseos)} & \text{estandar} \\ nada & \text{sedentario} \end{cases} \quad (14)$$

- **Otras restricciones.** Religión, factores económicos, vegetariana, etc. Este tipo de restricciones también se tienen en cuenta en la segunda fase de recomendación.
- **Contraindicaciones y gustos.** La segunda fase de recomendación consiste en la modificación de la dieta para adaptarse a los gustos y requisitos personales del usuario. Esta fase puede llevarse a cabo en dos momentos diferentes: en el paso previo a la recomendación de la dieta (donde se comprueba que no aparece ningún alimento contraindicado o que no sea del agrado del usuario y si aparece, debe sustituirse por otro nutricionalmente similar) o, una vez el usuario recibe la dieta recomendada (donde se le ofrece la posibilidad de modificar alguno de los alimentos, dándole la opción de elegir entre diferentes alternativas con aporte nutricional parecido). Para ello, es necesario recoger en el formulario inicial sus contraindicaciones o alergias y qué alimentos no son de su agrado (gustos explícitos), estos alimentos o sustancias no deben aparecer en las recomendaciones.

Todos aquellos alimentos que no son recomendables serán identificados/clasificados/marcados en el sistema como *no-interesantes*. A medida que el usuario va usando el sistema, va proporcionando *feedback relevante* (capítulo 1, apartado 1.4). De este modo el sistema puede tener constancia de las recomendaciones ofrecidas y las selecciones realizadas por el usuario, infiriendo así sus gustos implícitos, es decir, el grado de *interés* de un usuario por un alimento. Los intereses de los usuarios son la parte más importante del perfil de usuario en los SR (Brusilovsky and Millán, 2007). Las selecciones sobre las recomendaciones representarán los *ratings* sobre los ítems del dominio que deben contener los perfiles de usuario de los SR para usarse en la inferencia de intereses en los procesos de recomendación (Resnick and Varian, 1997).

En la siguiente sección se presenta la representación ontológica del perfil de usuario propuesta para recibir recomendaciones en el dominio nutricional y en el capítulo 7 se muestra el proceso de aprendizaje de los gustos implícitos inferidos a partir de la selección en las recomendaciones.

6.5. Representación del Perfil de Usuario

El modelado del usuario en el sistema se ha realizado mediante perfiles de usuario ontológicos enriqueciendo el razonamiento junto a la ontología nutricional y potenciando la reutilización. Cuando el usuario entra al sistema, se instancia su perfil de usuario en la ontología de la base de conocimiento para el razonamiento durante los procesos de recomendación.

6.5.1 Reutilización de Modelos de Usuario Ontológicos

Al igual que ocurría en el caso de la ontología nutricional, es importante tener en cuenta que podemos beneficiarnos de modelos de usuario ontológicos ya existentes. Según el grado de reutilización, generalización y dependencia del dominio se puede distinguir entre distintos tipos de ontologías para representar al usuario en sistemas de recomendación:

- Ontología *top-level* o general. Contiene conceptos genéricos para representar a los usuarios. Su información es reutilizable en cada dominio puesto que es independiente del mismo.
- Ontología a nivel de dominio. Extiende a la anterior. Representa los conceptos dependientes del dominio de los usuarios, por lo que es reutilizable en el mismo dominio e independiente de la aplicación.
- Ontología a nivel de aplicación. Extiende a la ontología de dominio. Es dependiente del dominio y contiene la información del usuario relativa a la aplicación particular. Sobre ella se aplicará el razonamiento para la inferencia de nuevo conocimiento y producción de recomendaciones.

El primer paso en la toma de decisiones de diseño de la ontología ha sido el estudio de otras propuestas de perfil de usuario ontológico para ser reutilizadas en el ámbito de recomendación nutricional, es decir, una ontología de dominio. La propuesta encontrada más cercana a nuestro objetivo ha sido la aportada por Al-Nazer y colaboradores (Al-Nazer et al., 2014). Sin embargo, esta propuesta carece de ciertas características que los expertos recomiendan tomar en cuenta en los procesos de recomendación nutricionales como información relevante de contexto, ingestas previas o el grupo de edad de los usuarios. Además, no se considera la reutilización de una ontología general, lo que dificulta la interacción con otras aplicaciones o perfiles de usuario semánticos.

Para construir un modelo ontológico del usuario es posible partir de una ontología general existente (o *top-level*) y seguir un proceso de extensión hacia una ontología de dominio para ser reutilizada en la aplicación donde se llevará a cabo el razonamiento (Andrejko et al., 2007). Al no encontrarse una ontología de dominio adecuada, nuestra propuesta de perfil de usuario ha seguido este enfoque partiendo de la reutilización de una ontología general *top-level*.

En el capítulo 3 (apartado 3.2.2.) se estudiaron los diferentes planteamientos de modelos ontológicos de usuario generales encontrados en la literatura. De entre ellos, se ha escogido reusar la ontología GUMO (Heckmann et al., 2005) como *top-level* ya que, además de estar desarrollada en OWL, se ha considerado lo suficientemente completa y extensible para nuestro propósito.

GUMO o sus derivados ya han sido utilizados para definir perfiles de usuarios en diferentes sistemas como (Basit and Matskin, 2011) y (Skillen et al., 2012). GUMO considera cuatro dimensiones de usuario: *Básica* (conteniendo los datos demográficos del usuario y de personalidad), *de Contexto* (para representar el contexto actual del usuario), *de Sensores* (para medidas de sensores) y *Dependiente del Dominio* (donde se almacenan los intereses, conocimiento y preferencias en los diferentes dominios). Su diseño es concordante con la especificación UserML (Heckmann and Krueger, 2003) que divide las dimensiones del modelo de usuario en tres partes: *auxiliar*, *predicado* y *rango*. Por ejemplo, una aserción como “interés en alimentación” puede ser dividida en *auxiliar=tieneInterés*, *predicado=alimentación* y en el *rango=poco-medio-bueno-excelente*. Esta aserción puede ser utilizada para conocer el grado de interés del usuario en recibir recomendaciones sobre alimentación en diferentes entornos semánticos.

GUMO por sí misma puede usarse como ontología *top-level* para preservar la reutilización y permitir interacciones con modelos de usuario externos, pero se precisa extenderla para representar preferencias y requerimientos nutricionales. A continuación se presenta el proceso de extensión de GUMO para obtener un modelo de usuario tal que sea lo suficientemente general para satisfacer un amplio espectro de entornos nutricionales orientados a la salud cubriendo la mayoría de los requerimientos nutricionales de los usuarios.

6.5.2 Desarrollo de la Ontología de Perfil de Usuario

El perfil del usuario se ha diseñado para que las recomendaciones se adapten tanto a los requerimientos nutricionales como a las preferencias de los usuarios (apartado 6.4). De acuerdo a (Middleton et al., 2004), la definición de perfiles de usuarios en SR está típicamente basado en conocimiento o en comportamiento. El enfoque basado en conocimiento construye modelos estáticos de los usuarios y hace *matching* dinámicamente al modelo más cercano o similar. El enfoque basado en comportamiento, usa el comportamiento como un modelo y normalmente emplea técnicas de inferencia automática para descubrir patrones de comportamiento útiles. En nuestra propuesta de perfil de usuario nos interesan ambos enfoques por lo que será un híbrido de ambos. A continuación se explica cómo se ha construido cada uno de ellos.

Perfil de usuario basado en conocimiento

Esta parte del perfil se modela principalmente mediante las dimensiones básicas de GUMO, en particular, reusando las clases: *Contact Information*, *Demographics*, *Nutrition*, *Physiological State* y algunas clases de la dimensión de contexto como, *Physical Environment* y sus descendientes *Humidity*, *Temperature* y *Weather*. Como subclases de *Physiological State* se han añadido las clases *Swallowing*, *Chewing* y *Nutritional State* para recoger los niveles de deglución y masticación junto con el estado nutricional del usuario. Además, se ha incluido una nueva clase *Physical State* con subclases *weight*, *height*, *activity level*.

Las figuras 6.23 y 6.24 muestran las partes más significativas de la ontología que contienen los conceptos del perfil de usuario basado en conocimiento.

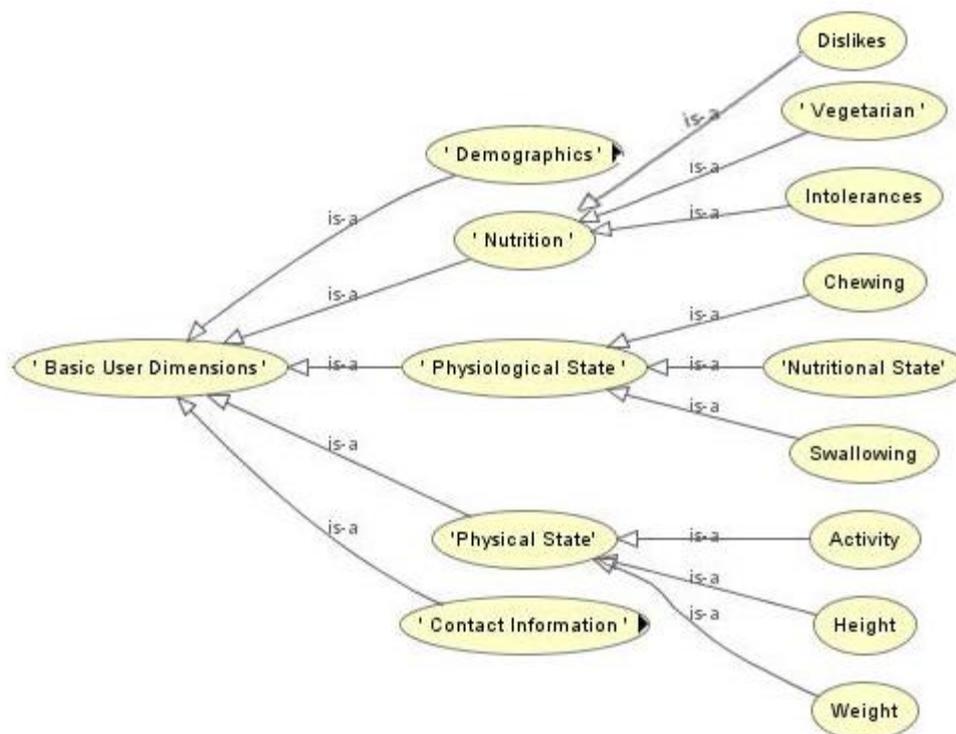


Figura 6.23. Extensión de la dimensión básica del perfil de usuario GUMO con requerimientos nutricionales.

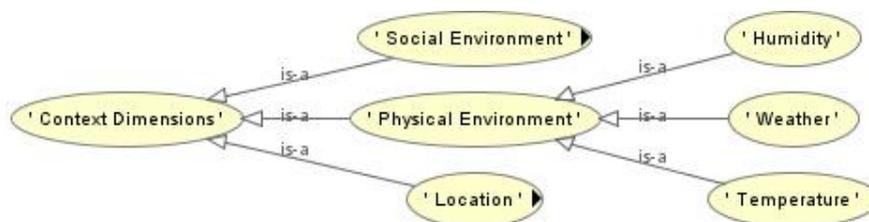


Figura 6.24. Dimensión de contexto de GUMO reutilizada en el perfil de usuario.

GUMO contiene un vocabulario sin restricciones ni axiomas. Además, las propiedades están modeladas como subclases de la clase Auxiliary, lo cual no permite capturar el significado inherente a dicho vocabulario. En la ontología que se propone las propiedades han sido modeladas mediante propiedades de datos y de objeto OWL y se ha añadido los axiomas y restricciones necesarios para permitir inferencias más precisas.

Como ejemplo de propiedades añadidas podemos citar las siguientes:

- *hasLowToHighLevel*: usada para clasificar ciertas clases como *Chewing* y *Swallowing*, pudiendo tomar un valor de la clase *LowToHighValues* ("VERY HIGH", "HIGH", "MEDIUM", "LOW", "VERY LOW"). A ella se le añaden las subpropiedades de este tipo, por ejemplo:
 - o *hasChewLevel* que clasifica el nivel de masticación del usuario.
 - o *hasSwallowLevel* que clasifica el nivel de deglución del usuario.

- *hasBMI*: representa el índice de masa corporal del usuario. Se usa para inferir la propiedad *hasNutritionalState*.
- *hasNutritionalState*: los valores de esta propiedad de objeto se infieren desde la propiedad *hasBMI* y la edad del usuario (mayor o menor de 75 años) y puede contener uno de los valores de la clase *NutritionalState*: *malnourished*, *underweight*, *normalweight*, *overweight*, *obese*.
- *canEat*: esta propiedad afirma si un usuario puede ingerir algún alimento o sustancia determinado.

Perfil de usuario basado en comportamiento

Aunque las técnicas empleadas para inferir las preferencias de los usuarios se explicarán en el siguiente capítulo, en esta sección se presenta someramente la representación de los conceptos que se han introducido en la ontología para permitir el aprendizaje del comportamiento.

Las técnicas de recomendación para dicho aprendizaje serán técnicas basadas en contenido las cuales recomendarán un ítem a un usuario basándose en los *ratings* de los usuarios sobre los ítems y empleando cálculos de similaridad semántica. Esta parte del perfil se obtendrá a partir de la interacción del usuario con los alimentos a recomendar. Debe actualizarse automáticamente en cada selección en respuesta al *feedback* acerca del interés de los ítems que se le han recomendado (Pazzani and Billsus, 2007).

GUMO incluye el concepto *hasRated* para registrar los ítems que han sido valorados por el usuario. En NutElCare, sin embargo, a los usuarios no se les pide que valoren ítems, si no que se tiene en cuenta las selecciones previas sobre los alimentos recomendados. Puesto que el *feedback* es implícito, se realiza de forma transparente, por lo que no resulta intrusivo al usuario. Para sustituir a *hasRated* se han introducido dos nuevas propiedades de datos, que se irán actualizando automáticamente y que permitirán obtener preferencias del usuario a partir de su comportamiento:

- *timesRecommended*: número de veces que se ha recomendado un cierto alimento al usuario.
- *timesSelected*: número de veces que ha seleccionado ese alimento en las recomendaciones previas.

Finalmente, para realizar el seguimiento de los alimentos ingeridos durante la semana se incluye la clase *IngestedFood* cuyos descendientes serán instancias de la ontología de alimentos.

6. 6. Enlazado de la Ontología Nutricional y el Perfil de Usuario

En las secciones previas se han presentado la ontología nutricional y la ontología de perfil de usuario por separado. Sin embargo, para llevar a cabo el razonamiento de los procesos de recomendación ambas ontologías deben razonar juntas. Existen diferentes modos para combinar diferentes ontologías en un sola:

- Fusión de ontologías (*merge*). Es el proceso de generar una sola ontología coherente a partir de dos o más ontologías diferentes relacionadas con un mismo dominio. La ontología resultante incluye la información de todas las ontologías fuente, pero sin apenas cambios. Requiere revisiones posteriores menores de la ontología final.

- Alineamiento de ontologías (*alignment*). Es la tarea de crear enlaces entre dos ontologías originales. Se consigue el alineamiento de ontologías si se logra hacerlas consistentes entre sí, pero se mantienen separadas. Se establece por lo general cuando las ontologías tienen dominios complementarios.
- Integración de ontologías (*integration*). Es el proceso de generar una sola ontología sobre una materia a partir de dos o más ontologías con diferentes materias, las cuales pueden estar relacionadas. En una ontología integrada se requerirán cambios posteriores.

Puesto que ambas ontologías a fusionar en la base de conocimiento son del mismo dominio y han seguido un proceso de diseño similar, se llevó a cabo una operación de fusión entre ellas. Protégé permite efectuar esta fusión mediante la operación MERGE de su interfaz de usuario. Posteriormente, sólo se necesitaron mínimas revisiones en la ontología obtenida.

Esta ontología resultante permite la aplicación de axiomas y restricciones que combinan conceptos de la ontología nutricional y la ontología de perfil de usuario. A modo ilustrativo se muestra el proceso de inclusión de la restricción “Los vegetarianos no pueden comer carne”. Seleccionamos la clase *Vegetarian* y le añadimos la siguiente equivalencia de clase:

EquivalentTo not (canEat some (Meals and (hasIngredient some MeatProducts))).

Cuya representación en OWL es:

```
<owl:Class rdf:about="http://gumo.org/2.0/#Vegetarian.801101">
  <rdfs:label xml:lang="en">Vegetarian</rdfs:label>
  <owl:equivalentClass>
    <owl:Class>
      <owl:complementOf>
        <owl:Restriction>
          <owl:onProperty rdf:resource="#canEat"/>
          <owl:someValuesFrom>
            <owl:Class>
              <owl:intersectionOf rdf:parseType="Collection">
                <rdf:Description rdf:about="#Meals"/>
                <owl:Restriction>
                  <owl:onProperty rdf:resource="#hasIngredient"/>
                  <owl:someValuesFrom rdf:resource="#Carne"/>
                </owl:Restriction>
              </owl:intersectionOf>
            </owl:Class>
          </owl:someValuesFrom>
        </owl:Restriction>
      </owl:complementOf>
    </owl:Class>
  </owl:equivalentClass>
<rdfs:subClassOf rdf:resource="#Person"/>
</owl:Class>
```

6.7. Razonamiento mediante Reglas SWRL

Como se explicó en el capítulo 2, el uso de reglas SWRL (Horrocks et al., 2004) extiende los axiomas OWL con cláusulas de Horn aportando mayor expresividad y capacidad de razonamiento a la base de conocimiento. En NutElCare se utilizan este tipo de reglas con distintos propósitos:

- Clasificaciones de propiedades nutricionales.
- Cálculos de valores en propiedades de datos.
- Refuerzo de restricciones en ingestas.

A continuación se explica cada uno de estos tipos y se muestran algunos ejemplos.

Clasificación de propiedades nutricionales

En la base de conocimiento de NutElCare interesa clasificar ciertas propiedades nutricionales usando la clase *LowToHighLevel*, por ejemplo, para clasificar la propiedad *contenidoCalórico* según la siguiente norma (ecuación 5):

$$CaloricLevel(f) = \begin{cases} c \geq 300 & VERY\ HIGH \\ 200 \geq c \geq 299 & HIGH \\ 100 \geq c \geq 199 & MEDIUM \\ 50 \geq c \geq 99 & LOW \\ 0 \geq c \geq 49 & VERY\ LOW \end{cases} \quad (15)$$

Donde f es un determinado alimento, c es el valor de la propiedad “calorías por 100 gramos” y VERY HIGH, HIGH, MEDIUM, LOW y VERY LOW son instancias de la clase *LowToHighLevel*. Las reglas añadidas para esta clasificación son⁶³:

'Food products'(?c), 'calories per 100 gr'(?c, ?x), greaterThan(?x, 299) -> 'Caloric level'(?c, VERY_HIGH)

'Food products'(?c), 'calories per 100 gr'(?c, ?x), greaterThan(?x, 199), lessThan(?x, 300) -> 'Caloric level'(?c, HIGH)

'Food products'(?c), 'calories per 100 gr'(?c, ?x), greaterThan(?x, 99), lessThan(?x, 200) -> 'Caloric level'(?c, MED)

'Food products'(?c), 'calories per 100 gr'(?c, ?x), greaterThan(?x, 49), lessThan(?x, 100) -> 'Caloric level'(?c, LOW)

'Food products'(?c), 'calories per 100 gr'(?c, ?x), lessThan(?x, 50) -> 'Caloric level'(?c, VERY_LOW)

Cálculos de valores en propiedades de datos

Este tipo de reglas permite el cálculo de valores de propiedades de datos a partir de otras propiedades. Por ejemplo, la siguiente regla SWRL indica que el valor de la propiedad “grasa por 100 gramos” es la suma de los valores de las propiedades “grasas saturadas” y “grasas insaturadas”:

'Food products'(?c), 'insaturated fat'(?c, ?i), 'saturated fat'(?c, ?s), add(?f, ?i, ?s) -> 'fat per 100 gr'(?c, ?f)

El valor de la propiedad “proteínas por ración” puede obtenerse a partir de operaciones con las propiedades “peso en gramos de una ración” y “proteínas por 100 gramos” mediante la regla:

'Food products'(?c), 'weight in grams per ration'(?c, ?y), 'proteins per 100 gr'(?c, ?z), divide(?x, ?m, 100.0f), multiply(?m, ?y, ?z) -> 'proteins per ration'(?c, ?x)

Refuerzo de restricciones en ingestas

La inclusión de este tipo de reglas proporciona mayor expresividad en las restricciones alimenticias. La idea principal es detectar inconsistencias en la ontología en los procesos de recomendación. Por ejemplo, mediante las siguientes dos reglas expresamos que un mismo pescado o una misma carne no puede aparecer más de dos veces en la dieta semanal. Es decir, un mismo pescado o carne no puede aparecer una tercera vez en la dieta, si ocurriera, la ontología sería inconsistente. Esta restricción puede expresarse mediante las siguientes reglas SWRL:

⁶³ Para distinguir entre axiomas OWL y reglas SWRL se utiliza diferente tipo de tipografía.

'Fish'(?f), dietComposition(?m1), dietComposition (?m2), dietComposition (?m3), Contains(?m1, ?f), Contains(?m2, ?f), Contains(?m3, ?f), DifferentFrom (?m1, ?m2), DifferentFrom (?m1, ?m3), DifferentFrom (?m2, ?m3) -> (not (dietComposition))(?m3)

'Meat'(?f), dietComposition(?m1), dietComposition (?m2), dietComposition (?m3), Contains(?m1, ?f), Contains(?m2, ?f), Contains(?m3, ?f), DifferentFrom (?m1, ?m2), DifferentFrom (?m1, ?m3), DifferentFrom (?m2, ?m3) -> (not (dietComposition))(?m3)

Pero además, este hecho también ha de tenerse en cuenta en los demás procesos de recomendación, puesto que la dieta se va modificando conforme el usuario realiza selecciones. Para que un alimento que ya ha sido ingerido dos veces en esa semana no pueda aparecer en futuras recomendaciones, es necesario añadir también las dos siguientes reglas:

'Fish'(?f), ingestedFood(?m1), ingestedFood (?m2), ingestedFood (?m3), Contains(?m1, ?f), Contains(?m2, ?f), Contains(?m3, ?f), DifferentFrom (?m1, ?m2), DifferentFrom (?m1, ?m3), DifferentFrom (?m2, ?m3) -> (not (ingestedFood))(?m3)

'Meat'(?f), ingestedFood (?m1), ingestedFood (?m2), ingestedFood (?m3), Contains(?m1, ?f), Contains(?m2, ?f), Contains(?m3, ?f), DifferentFrom (?m1, ?m2), DifferentFrom (?m1, ?m3), DifferentFrom (?m2, ?m3) -> (not (ingestedFood))(?m3)

Por último, para indicar que a un usuario sólo pueden recomendársele alimentos que pueda comer, se añade la siguiente regla:

Meals(?m), Person(?p), 'has Recommended Meals'(?p, ?m) -> 'Can Eat'(?p, ?m)

Esta regla se complementa con una restricción OWL en la clase “Persona” que indica que una persona no puede comer alimentos que contengan instancias de la clase “Intolerancias”:

Person EquivalentTo not ('Can Eat' some (Meals and (Contains some Intolerances))).

Y finalmente, se concluye con la restricción de la clase “Intolerancias” (que contiene aquellos alimentos o sustancias que el usuario no puede ingerir):

Intolerances EquivalentTo not ('Can Be Ingested By' some Person).

Referencias del capítulo

- AL-NAZER, A., HELMY, T. & AL-MULHEM, M. 2014. User's Profile Ontology-based Semantic Framework for Personalized Food and Nutrition Recommendation. *Procedia Computer Science*, 32, 101-108.
- ANDREJKO, A., BARLA, M. & BIELIKOVÁ, M. 2007. Ontology-based user modeling for web-based information systems. *In Advances in Information Systems Development* Springer US.
- ARBONÉS, G., CARBAJAL, A., GONZALVO, B., GONZÁLEZ-GROSS, M., JOYANES, M., MARQUES-LOPES, I. & PUIGDUETA, I. 2003. Nutrición y recomendaciones dietéticas para personas mayores: Grupo de trabajo "Salud pública" de la Sociedad Española de Nutrición (SEN). *Nutrición hospitalaria*, 18, 109-137.
- BAADER, F., HORROCKS, I. & SATTLER, U. 2005. Description logics as ontology languages for the semantic web. *Mechanizing Mathematical Reasoning*. Springer Berlin Heidelberg.
- BASIT, K. A. & MATSKIN, M. GUMO inspired ontology to support user experience based Citywide Mobile Learning. International Conference on User Science and Engineering (i-USEr) 2011 2011. IEEE, 195-200.
- BOHRING, H. & AUER, S. 2005. Mapping XML to OWL Ontologies. *Leipziger Informatik-Tage*, 72, 147-156.

- BRUSILOVSKY, P. & MILLÁN, E. 2007. User models for adaptive hypermedia and adaptive educational systems. *The adaptive web*. Springer-Verlag.
- BÜHMANN, L. & LEHMANN, J. 2012. Universal OWL axiom enrichment for large knowledge bases. *Knowledge Engineering and Knowledge Management*. Springer Berlin Heidelberg.
- CARACCILO, C., STELLATO, A., MORSHED, A., JOHANNSEN, G., RAJBHANDARI, S., JAQUES, Y. & KEIZE, J. 2013. The Agrovoc linked dataset. *Semantic Web*, 4, 341-348.
- CHANDRASEKARAN, B., JOSEPHSON, J. R. & BENJAMINS, V. R. 1999. What are ontologies, and why do we need them? *IEEE Intelligent Systems*, 1, 20-26.
- DAPCICH, V., CASTELL, G. S., BARBA, L. R., RODRIGO, C. P., BARTRINA, J. A. & MAJEM, L. S. 2004. Necesidades Nutricionales en el Envejecimiento. *Guía de la Alimentación Saludable*. Sociedad Española de Nutrición Comunitaria, Madrid, Everest, 96-98.
- ESPÍN, V., NOGUERA, M. & HURTADO, M. V. 2015. A Mixed Approach for the Representation of Nutritional Information Through XML-to-OWL Mappings. *Languages, Applications and Technologies*. Springer International Publishing.
- FERDINAND, M., ZIRPINS, C. & TRASTOUR, D. Lifting XML schema to OWL. *Web Engineering*, 2004. Springer Berlin Heidelberg, 354-358.
- GHAWI, R. & CULLOT, N. 2009. Building Ontologies from XML Data Sources. *DEXA Workshops*, 480-484.
- GRAU, B. C., HORROCKS, I., MOTIK, B., PARSIA, B., PATEL-SCHNEIDER, P. & SATTLER, U. 2008. OWL 2: The next step for OWL. *Web Semantics: Science, Services and Agents on the World Wide Web*, 6, 309-322.
- GRUBER, T. R. 1995. Toward principles for the design of ontologies used for knowledge sharing? *International journal of human-computer studies*, 43, 907-928.
- GRUNINGER, M. & FOX, M. S. Methodology for the design and evaluation of ontologies. Proceedings of the Workshop on Basic Ontological Issues in Knowledge Sharing IJCAI, 1995. AAAI Press, 1-10.
- GUARINO, N. 1998. *Formal Ontology in Information Systems: Proceedings of the 1st International Conference June 6-8, 1998, Trento, Italy*, IOS Press.
- HACHEROUF, M., BAHLOUL, S. N. & CRUZ, C. 2015. Transforming XML documents to OWL ontologies: A survey. *Journal of Information Science*, 41, 242-259.
- HECKMANN, D. & KRUEGER, A. 2003. A user modeling markup language (UserML) for ubiquitous computing. *User Modeling*. Springer Berlin Heidelberg.
- HECKMANN, D., SCHWARTZ, T., BRANDHERM, B., SCHMITZ, M. & VON WILAMOWITZ-MOELLENDORFF, M. 2005. Gumo—the general user model ontology. *User Modelling*. Springer Berlin Heidelberg.
- HORROCKS, I., PATEL-SCHNEIDER, P. F., BOLEY, H., TABET, S., GROSOFF, B. & DEAN, M. 2004. *SWRL: A semantic web rule language combining OWL and RuleML* [Online]. W3C Member submission. Available: <http://www.w3.org/Submission/SWRL/> [Accessed 30 November 2015].
- MIDDLETON, S. E., SHADBOLT, N. R. & DE ROURE, D. C. 2004. Ontological user profiling in recommender systems. *ACM Transactions on Information Systems (TOIS)*, 22, 54-88.
- NOY, N. F. & MCGUINNESS, D. L. 2001. *Ontology development 101 a guide to creating your first ontology*. Technical Report. Stanford Knowledge Systems Laboratory: Stanford University, Stanford, AC.
- PAZZANI, M. & BILLSUS, D. 2007. Content-based recommendation systems. *The Adaptive Web*. Lecture Notes in Computer Science, Springer Berlin, Heidelberg.
- PINILLA, M. F. & DE TORRES AURED, M. L. 2007. La Dieta equilibrada: guía para enfermeras de Atención Primaria. *Sociedad Española de Dietética y Ciencias de la Alimentación*.

- PROTÉGÉ. 2013. *Protégé ontology editor* [Online]. Available: <http://protege.stanford.edu/> [Accessed 30 Nov 2015].
- RESNICK, P. & VARIAN, H. R. 1997. Recommender systems. . *Communications of the ACM*, 40, 56-58.
- SENPE & SEGG 2007. *Valoración nutricional en el anciano. Documento de Consenso.*, Galenitas-Nigra Trea.
- SIRIN, E., PARSIA, B., GRAU, B. C., KALYANPUR, A. & KATZ, Y. 2007. Pellet: A practical owl-dl reasoner. *Web Semantics: science, services and agents on the World Wide Web*, 5, 51-53.
- SKILLEN, K., CHEN, L., NUGENT, C. D., DONNELLY, M. P. & SOLHEIM, I. A user profile ontology based approach for assisting people with dementia in mobile environments. Annual International Conference on Engineering in Medicine and Biology Society (EMBC) 2012 2012. IEEE, 6390-6393.
- TRAJKOVA, J. & GAUCH, S. 2004. Improving Ontology-Based User Profiles. *RIAO*, 2004.

CAPÍTULO 7

EXPLOTACIÓN DEL CONOCIMIENTO SEMÁNTICO Y PRODUCCIÓN DE RECOMENDACIONES

Una vez instanciado el marco conceptual del sistema mediante la representación ontológica del conocimiento nutricional y del perfil de usuario en la base de conocimiento, es posible explotar la información recogida en la base de conocimiento resultante mediante el uso de razonadores semánticos como Pellet, infiriendo así nuevo conocimiento que no se encuentra representado explícitamente en las ontologías. En este capítulo se presenta la estrategia de recomendación que explota el conocimiento semántico en el sistema para ofrecer a los usuarios recomendaciones nutricionales adecuadas a sus gustos, necesidades y perfil. Para una mejor comprensión de los procesos que intervienen en dicha estrategia, a lo largo del capítulo se presentan ejemplos de razonamiento. También se realiza un análisis de aplicación de técnicas de similitud semántica de forma que se adapten a las necesidades habituales en el dominio nutricional. Finalmente, se valida la aplicabilidad de la estrategia propuesta mediante un ejemplo de uso en NutEiCare.

7.1. Introducción

Este capítulo se centra en la explotación del conocimiento semántico cuya representación se explicó en el capítulo anterior. Gracias a esta representación, empleando ontologías OWL 2 (Grau et al., 2008) es posible la inferencia de nuevo conocimiento mediante un razonador semántico. La inferencia de conocimiento en el ámbito ontológico consiste básicamente en el descubrimiento de nuevas relaciones entre sus componentes.

En la primera parte del capítulo se explica, a través de ejemplos, diversos procesos de explotación de conocimiento semántico mediante razonamiento que pueden tener lugar en la ontología, facilitando así la comprensión de los procesos de recomendación.

A continuación, se lleva a cabo la formalización de la estrategia de recomendación propuesta donde se muestran los procesos principales que intervienen en ella y que se analizan en este capítulo. En primer lugar, se aborda la instanciación del usuario en el sistema y de ahí, se procede a la recomendación de dietas y alimentos.

La instanciación del usuario consiste en un proceso de población de la ontología con los datos del usuario almacenados en la base de datos de respaldo. Durante este proceso de población se disparan reglas y restricciones que van a determinar las recomendaciones posteriores. Asimismo, la selección de la dieta más adecuada al usuario conlleva un proceso de población de la ontología con los diferentes alimentos y relaciones dietéticas que intervienen.

NutEiCare recomienda dos tipos de ítems: dietas y alimentos, aunque también se favorece el uso de recetas o platos que permite el modelo de representación propuesto. La estrategia de recomendación es una estrategia híbrida que combina el paradigma *basado en conocimiento* y el paradigma *basado en contenido*. Puesto que el sistema no contempla de momento el paradigma colaborativo, los ítems se recomiendan basándose en la información del ítem en sí, en lugar de las preferencias de otros usuarios, lo que permite ofrecer a los usuarios explicaciones sobre las recomendaciones y aumentar la confianza del usuario en las mismas (Mooney and Roy, 2000).

Por un lado, se estudian las características de los ítems para que compartan una descripción común mediante un conjunto de características denominadas *propiedades* o *atributos* (Lops et al., 2011). En NutElCare, esta descripción se materializa mediante el uso de propiedades, relaciones jerárquicas (is-a), restricciones y reglas SWRL de apoyo. Por otro lado, los perfiles de usuario incluyen sus intereses o gustos y actividades pasadas con respecto a los ítems, es decir, se observa el comportamiento del usuario a través del *feedback relevante* proporcionado.

Una técnica fundamental en los paradigmas de recomendación es el cálculo de la *similaridad semántica* entre ítems o entre ítems y usuarios. En este capítulo se analizan las técnicas clásicas empleadas de cálculo de similaridad y las debilidades detectadas por diferentes autores, tratando a su vez de proponer alternativas o mejoras. En versiones previas de este trabajo, publicadas en (Espín et al., 2013) y (Espín et al., 2016) se han propuesto diferentes métodos de cálculo de similaridad e inclusión de intereses de usuario que han ido evolucionando hasta la versión que hoy nos ocupa.

Al final del capítulo, se valida la aplicabilidad de la estrategia de recomendación propuesta mediante un ejemplo de recomendación nutricional completo.

Para ilustrar las capacidades del razonamiento basado en lógica descriptiva se ha seleccionado el razonador Pellet (Sirin et al., 2007), explicado en el capítulo 2. Otros ejemplos de razonadores son Hermit (Shearer et al., 2008) y Fact++ (Tsarkov and Horrocks, 2006), presentados también en el mismo capítulo, que difieren en algunas de las inferencias que proporcionan, si bien, en general, suelen ser capaces de producir las mismas inferencias. Pellet es un razonador semántico *open source* desarrollado en Java que permite adquirir conocimiento no representado de forma explícita en la ontología y que, además, soporta el uso de reglas SWRL (Horrocks et al., 2004) que aportan otro nivel expresivo más allá del permitido por OWL.

La gestión y manipulación de las ontologías durante el funcionamiento del sistema se lleva a cabo con la librería java OWL API (Horridge and Bechhofer) que, a su vez, permite la interacción con el razonador Pellet.

7.2. Explotación del Conocimiento Semántico

En el capítulo 6 se describió el diseño de la base de conocimiento, representada mediante ontologías OWL. Gracias al uso de razonadores como Pellet, es posible la inferencia de conocimiento que no se encuentra representado explícitamente en una ontología.

A continuación, y a partir de la representación de conocimiento del capítulo anterior, se muestran ejemplos de algunos de los diferentes tipos de inferencia que se producen en el sistema durante los procesos de razonamiento. Aunque es OWL API el que invoca al razonador desde la aplicación, los resultados de ejecución de los ejemplos se presentan a través de capturas la interfaz gráfica de Protégé para una mejor comprensión.

7.2.1 Inicialización de OWL API

El *back-end* de NutElCare está implementado en Java y el soporte para la gestión de ontologías se realiza mediante OWL API, que carga la ontología en memoria al iniciarse el servidor (Tomcat) mediante un *Listener* de contexto (ver figura 7.1).

Cuando el usuario se identifica, se instancia su perfil durante su sesión, el cual se libera de memoria al finalizar dicha sesión. Este cambio en la ontología, como todos los demás, debe ser

notificado al razonador para que éste se actualice durante la ejecución del programa, resultando en la población de la ontología con la información del usuario actual y la dieta seleccionada previamente, si la hubiera. Para permitir esta notificación, el razonador se carga como un *Listener* al iniciarse la sesión del usuario (figura 7.2). Así, al cambiar la ontología, el razonador tiene disponible la actualización en memoria para la siguiente consulta.

```
public void contextInitialized(ServletContextEvent servletContextEvent) {
    ServletContext ctx = servletContextEvent.getServletContext();
    OWLOntologyManager OntManager = OWLManager.createOWLOntologyManager();

    // Creates the Pellet reasoner
    PelletReasonerFactory reasonerFactory = PelletReasonerFactory.getInstance();
    final IRI nutelcareIRI = IRI.create("http://asistic.ugr.es/nutelcare/Nutelcare-MERGED.owl");
    OWLDataFactory OntFactory = OntManager.getOWLDataFactory();
    try {

        ontology = OntManager.loadOntologyFromOntologyDocument(nutelcareIRI);
        ctx.setAttribute("OntManager", OntManager);
        ctx.setAttribute("nutelcareIRI", nutelcareIRI);
        ctx.setAttribute("reasonerFactory", reasonerFactory);
        ctx.setAttribute("OntFactory", OntFactory);
        ctx.setAttribute("ontology", ontology);

    } catch (OWLOntologyCreationException e) {
        // TODO Auto-generated catch block
        e.printStackTrace();
    }
}
```

Figura 7.1 Inicialización de OWL API con la ontología nutricional.

```
//Load reasoner
PelletReasoner pelletReasoner = reasonerFactory.createNonBufferingReasoner(ontology);
//Add the reasoner as an ontology change listener
OntManager.addOntologyChangeListener( pelletReasoner );
```

Figura 7.2 Carga del razonador Pellet como Listener de cambios en la ontología.

7.2.2 Razonamiento en NutElCare

El razonamiento en NutElCare tiene como objetivo la recomendación de dietas saludables adecuadas para el perfil nutricional de cada usuario, las cuales pueden ser personalizadas por ellos mismos para ajustarse a sus gustos o necesidades con la ayuda del sistema. La base de conocimiento ontológica en que se fundamentan todas las recomendaciones nutricionales se ha construido sobre las bases de la lógica descriptiva donde se definen dos tipos de axiomas:

- Axiomas terminológicos (o Tbox): incluyen el conjunto de conceptos y roles junto con sus relaciones de subsunción.
- Axiomas asertivos (o Abox): incluyen el conjunto de instancias, su pertenencia a clases y sus relaciones y propiedades.

Cada vez que se añade un axioma nuevo a la base de conocimiento, ya sea Tbox o Abox, ésta debe seguir siendo consistente. Mediante los razonadores semánticos como Pellet, se consigue detectar estados inconsistentes de la ontología al añadir nuevos axiomas, lo que implica que el axioma no puede incluirse en la base de conocimiento. Otro tipo de razonamiento es el descubrimiento de nuevas relaciones en los conjuntos Tbox o Abox, como las relaciones de subsunción entre clases o entre clases e instancias. Las reglas SWRL permiten también la inferencia de nuevas clasificaciones a partir del conocimiento que expresan.

A continuación, se presentan ejemplos de diferentes tipos de razonamiento que tienen lugar durante el funcionamiento de NutElCare. En primer lugar, la instanciación del estado nutricional

del usuario mediante reglas SWRL a partir de la información contenida en su perfil de usuario. En segundo lugar, se muestra un ejemplo de razonamiento donde se infiere conocimiento al alcanzar un estado inconsistente de la ontología. Para terminar, se presenta la justificación de inconsistencia del ejemplo anterior.

Obtención del Estado Nutricional del Usuario mediante reglas SWRL

Para comenzar con los ejemplos, se presenta la obtención del estado nutricional del usuario según las ecuaciones (6.1), (6.2) y (6.3) del capítulo anterior. En este ejemplo concreto, el perfil de usuario de una persona, Mary⁶⁴, indica que su edad es 77 años, su peso 53 kilogramos y su altura 1.70 metros. En el proceso de razonamiento, el sistema detecta la condición de malnutrición de Mary tras dispararse las siguientes reglas SWRL.

- Cálculo del índice de masa corporal a partir del peso y la altura:

```
Person(?P), 'has Height'(?P, ?h), 'has Weight'(?P, ?w), divide(?y, ?w, ?x), multiply(?x, ?h, ?h) -> 'has BMI'(?P, ?y)
```

- Detección del estado de malnutrición a partir de la edad y el índice de masa corporal:

```
Person(?p), 'has Age'(?p, ?a), 'has BMI' (?p, ?bmi), greaterThan(?a, 75), lessThan(?bmi, 18.5) -> 'has Nutritional State'(?p, malnutrition)
```

En la figura 7.3 se muestra el conocimiento inferido tras el razonamiento (Protégé expresa dicho conocimiento en color sepia). El índice de masa corporal (18.339 kg/m^2) y el subsecuente estado nutricional (*malnutrition*) se marcan usando un rectángulo de color rojo. La detección del estado de malnutrición envía una alerta al usuario y al cuidador o nutricionista correspondiente.

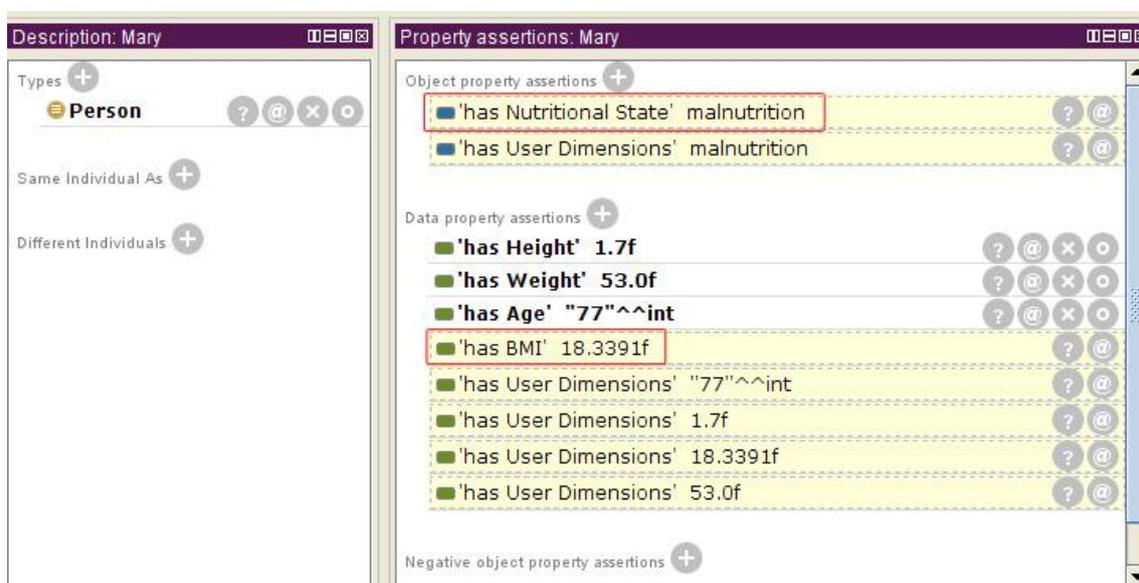


Figura 7.3. Inferencia del estado nutricional del usuario mostrado en Protégé.

⁶⁴ En NutEICare, para preservar la privacidad de los datos personales, las instancias de la clase persona se describen mediante identificadores únicos. En el ejemplo se muestra un nombre propio por claridad.

Introducción de Nuevos Axiomas y Detección de Inconsistencia en la Ontología

Durante el funcionamiento del sistema, se suceden modificaciones de la ontología que se traducen en la introducción de nuevos axiomas. Cada vez que se inserta un nuevo axioma, el razonador comprueba si la ontología sigue siendo consistente, y si no lo es, debe eliminarse el axioma introducido y volver al estado anterior. Este caso supone otro tipo de razonamiento consistente en el rechazo de una acción determinada a ejecutar en la base de conocimiento. Veamos ahora un ejemplo en el cual, se llega a un estado inconsistente de la ontología, indicando que la acción que se intenta ejecutar no puede llevarse a cabo.

Por ejemplo, supongamos que Mary es alérgica al gluten. Aunque el sistema le recomienda una dieta adecuada para celíacos, si Mary desea modificar un alimento de la dieta, entre los alimentos recomendados como alternativa, no puede aparecer ninguno que contenga gluten. Previamente se ha indicado que los cereales contienen gluten, y que la propiedad *Contains* es inversa de *is Contained in*.

En la figura 7.4 se muestra la instancia *Gluten* y sus propiedades *is Contained in* inferidas, resultando en todos los alimentos que contienen gluten. Además, puede verse que es de la clase *Intolerances* (cuyas instancias son las intolerancias nutricionales de Mary).

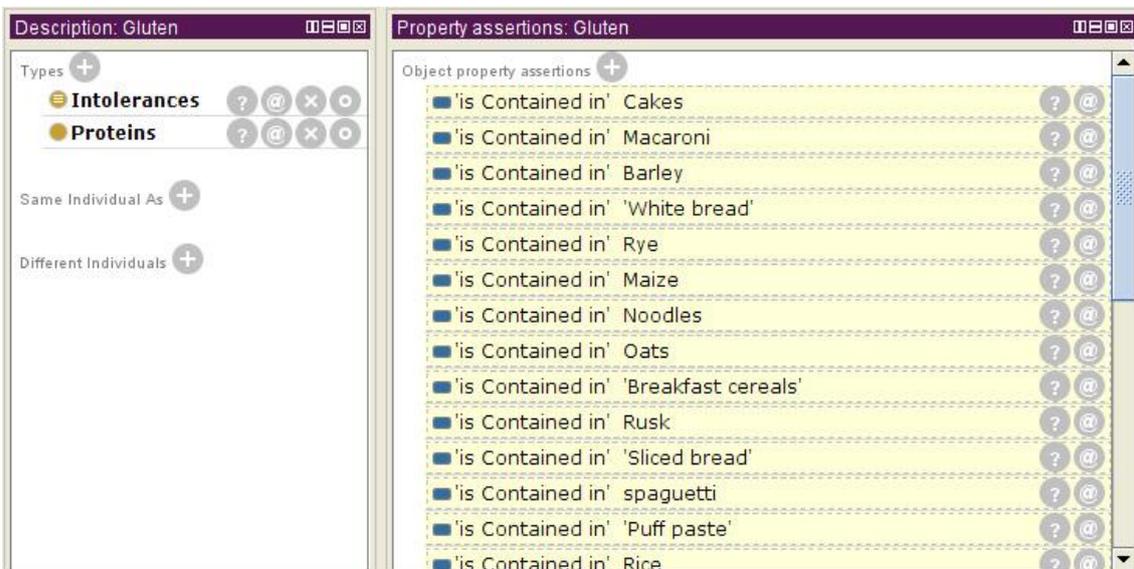


Figura 7.4. Propiedades de objeto inferidas de la instancia “Gluten”.

Ahora consideremos que Mary desea que se le recomienden alternativas a cierto plato de su dieta y que entre estas alternativas aparece el plato *spaguetti Bolognaise*, que no debería recomendarse a Mary, ya que la pasta contiene gluten. Veamos el proceso de razonamiento. Al introducir en la ontología el plato, se instancian sus ingredientes mediante la propiedad *Has Ingredient* como se muestra en la figura 7.5.

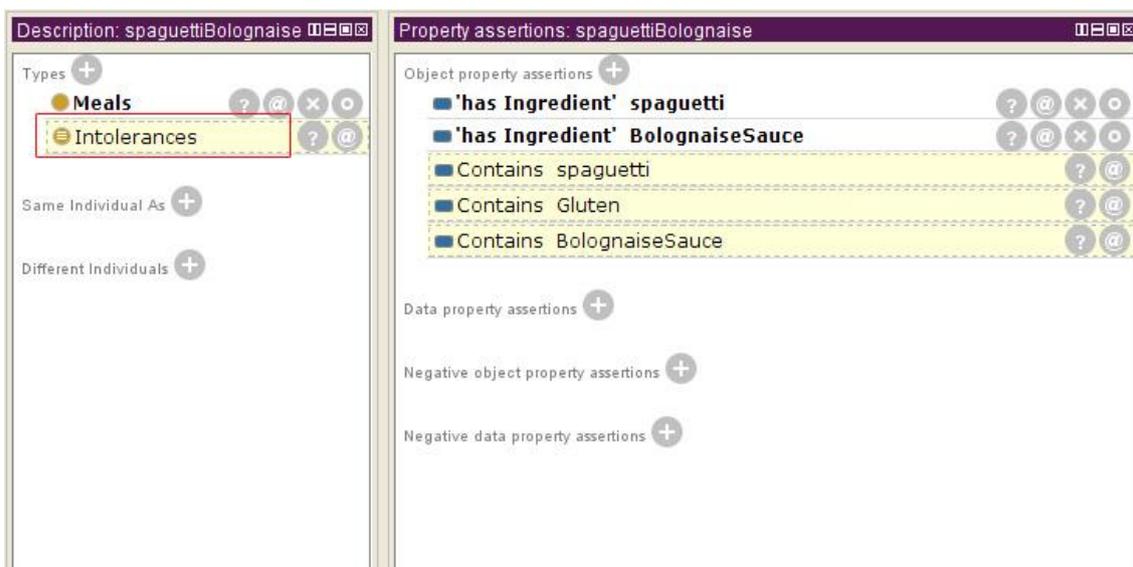


Figura 7.5. Propiedades del plato “spaguettiBolognaise”.

En la figura 7.5 se aprecia que uno de sus ingredientes es *spaguetti* que, a su vez, es instancia de la clase *Pasta* cuya superclase es *Cereals*, por lo que se infiere que contiene *Gluten* (ver figura 7.6) y, por lo tanto, es clasificado en la clase *Intolerances* (figura 7.5 recuadro rojo).

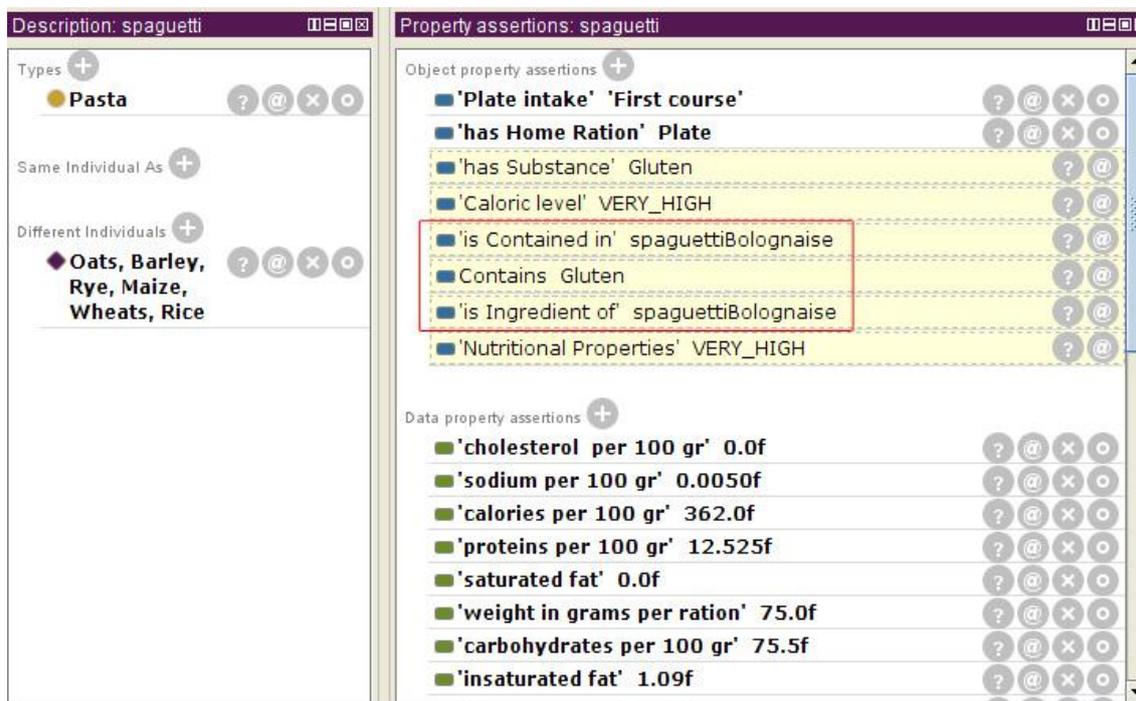


Figura 7.6. Descripción de la instancia “spaguetti”.

Si el sistema intenta recomendar ese plato a Mary, es decir, se intenta añadir el axioma *has Recommended Meal (Mary, spaguettiBolognese)*, se produciría una inconsistencia en la base de conocimiento. Por lo tanto, este axioma no puede ser introducido en la ontología y, de este modo, el sistema no puede recomendar a Mary el plato *spaguetti Boloñesa*.

Justificación de Inconsistencias

Cada vez que se añade un axioma nuevo a la ontología, por medio de OWL API y el razonador se chequea la consistencia con el nuevo axioma. Si como resultado de la inclusión de este nuevo axioma la ontología quedara en un estado inconsistente, se procede capturando la excepción y eliminando dicho axioma añadido volviendo la base de conocimiento al estado consistente anterior. Mediante la interfaz *ExplanationGenerator* de OWL API, es posible conocer el motivo de la inconsistencia, obteniendo las mismas justificaciones que las proporcionadas por Protégé. En la figura 7.7 se muestra la justificación de inconsistencia suministrada por Protégé cuando se intenta recomendar *spaguetti Boloñesa* a Mary.

Explanation for: Thing SubClassOf Nothing	
1)	Mary 'has Recommended Meal' spaguettiBolognaise
2)	Pasta SubClassOf Cereals
3)	Meals(?m), Person(?p), 'has Recommended Meal'(?p, ?m) -> 'Can Eat'(?p, ?m)
4)	Cereals(?c) -> Contains(?c, Gluten)
5)	Gluten Type Intolerances
6)	Transitive: Contains
7)	Mary Type Person
8)	Person EquivalentTo not ('Can Eat' some (Meals and (Contains some Intolerances)))
9)	spaguettiBolognaise Type Meals
10)	'has Ingredient' SubPropertyOf Contains
11)	spaguetti Type Pasta
12)	spaguettiBolognaise 'has Ingredient' spaguetti

Figura 7.7. Explicación de inconsistencia al recomendar a Mary un alimento con gluten.

7.3. Formalización de la Estrategia de Recomendación

En los capítulos 5 y 6 se introdujo la estrategia de recomendación propuesta para proporcionar dietas saludables a los usuarios. En este apartado se formaliza dicha propuesta y se define cada uno de los diferentes procesos que intervienen.

En la figura 7.8 se presenta el diagrama de flujo que establece los pasos a seguir en las diferentes fases de recomendación de NutElCare, desde que el usuario se identifica en el sistema hasta que se desconecta (el paso de desconexión no se refleja).

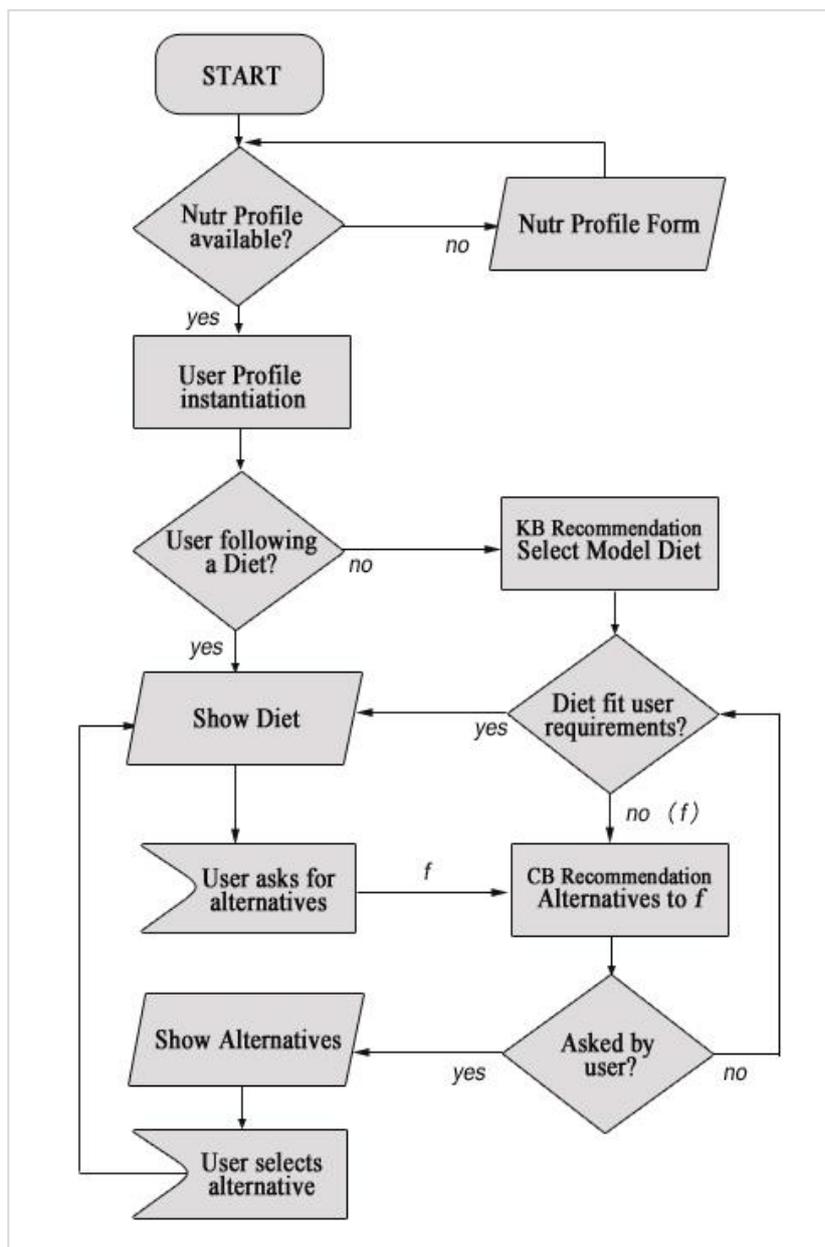


Figura 7.8 Diagrama de flujo de la estrategia de recomendación de NutEiCare.

El diagrama utiliza los elementos de representación de UML (Lenguaje Unificado de Modelado)⁶⁵. Los tres procesos principales mostrados en el diagrama y que rigen el funcionamiento del sistema para la producción de recomendaciones son:

- instanciación del perfil de usuario,
- selección del modelo de dieta (recomendación basada en conocimiento),
- recomendación de alternativas (recomendación basada en contenido).

En posteriores apartados se profundiza en cada uno de estos procesos y en el apartado 7.7 se presenta un ejemplo de aplicación de dicha estrategia.

⁶⁵ UML <http://www.uml.org/>

7.4. Instanciación del Perfil de Usuario

Cuando el usuario se identifica en el sistema se comprueba si existe su perfil nutricional en la base de datos de respaldo. Si no existe, se muestra el formulario inicial que el usuario debe rellenar con la información nutricional necesaria (peso, altura, ejercicio semanal, masticación, alergias, etc.).

Una vez los datos nutricionales están almacenados en la base de datos se procede a la población de la ontología de perfiles de usuario con dichos datos invocando el servicio web correspondiente.

Puesto que en la fase de diseño de la base de conocimiento se realizó el enlazado del perfil de usuario con la ontología nutricional, los conceptos y axiomas del perfil de usuario ya forman parte del sistema. Por lo tanto, es posible automatizar el proceso de población de la ontología mediante un mapeo simple en Java entre las columnas de la base de datos con las clases y propiedades de la ontología.

Si el usuario ha realizado selecciones previas de alimentos, estos valores también están almacenados en base de datos: así, por cada alimento que el usuario ha seleccionado en un proceso de recomendación y por cada alimento que se le ha recomendado, las propiedades *timesSelected* y *timesRecommended* toman los valores correspondientes respectivamente. Para tratar de equilibrar la eficiencia y la precisión al incluir selecciones previas durante los procesos de recomendación, previamente al uso de dichos valores es necesario llevar a cabo una decisión de diseño según los siguientes casos:

1. ¿La aparición de un alimento en la dieta debe incrementar el número de veces de recomendado?.
2. Si el usuario sigue la recomendación inicial de dicho alimento, ¿debe incrementarse el número de veces seleccionado?.
3. Si el usuario decide cambiar ese alimento, ¿cómo debe afectar a ambas propiedades?.

Parece claro que la respuesta a la pregunta número 1 afecta a la respuesta de la número 2, puesto que si un alimento que aparece en la dieta se clasifica como recomendado y el usuario sigue dicha recomendación, debe considerarse también como seleccionado. Lo mismo ocurre si no se considera como recomendado, en cuyo caso no tiene sentido incrementar el número de veces que el usuario lo ha seleccionado. La principal desventaja de mantener la lista con los alimentos recomendados en la dieta es el espacio en memoria, ya que partirían como instancias de la propiedad *has Recommended Food*. Sin embargo, un beneficio es el aumento de la precisión en los cálculos de las recomendaciones ya que se cuenta con más datos de partida en el proceso de aprendizaje. Por motivos de eficiencia en los procesos de razonamiento, se ha optado por no considerar los alimentos de la dieta inicial como recomendados, lo cual puede volver a reconsiderarse en un futuro. La respuesta a la pregunta 3, va a suponer una excepción a la respuesta de la pregunta 1. El que el usuario quiera modificar un alimento de la dieta indica un hecho relevante en el sistema y supuestamente menos asiduo. En este caso, el número de veces recomendado, a diferencia de como ocurre con las preguntas 1 y 2, se incrementa, permaneciendo igual el número de veces seleccionado.

Siguiendo el mapeo definido y el criterio de modificación de los valores de las propiedades *timesRecommended* y *timesSelected*, OWL API instancia los individuos de cada clase y valores de sus propiedades, pasando el perfil de usuario a formar parte de la base de conocimiento del

sistema. A continuación, el razonador realiza los procesos de inferencia necesarios (obteniendo, por ejemplo, el estado nutricional a partir de su edad y su índice de masa corporal).

7.5. Recomendación Basada en Conocimiento. Selección del Modelo de Dieta.

La primera recomendación que se lleva a cabo en el sistema se basa en conocimiento que, como se vio en el capítulo 1, establece una relación entre una necesidad y una recomendación centrándose en cómo un ítem satisface una necesidad del usuario (Burke, 2000). En este caso, los ítems son las dietas del repositorio que aparecen ya clasificadas como instancias en la ontología nutricional (capítulo 6). La necesidad del seguimiento por parte de un usuario de un tipo de dieta u otra viene determinada por diversos factores, principalmente los requisitos recogidos en su perfil nutricional y las deficiencias derivadas de sus hábitos dietéticos, como un déficit protéico que se debería corregir.

En los capítulos 1 y 3 se presentaron los enfoques más utilizados de *matching* ítem-usuario de la recomendación basada en conocimiento. El enfoque basado en casos utiliza técnicas de similitud con casos anteriores y entre requisitos del usuario y características de los ítems. El enfoque basado en restricciones realiza *mappings* entre los requisitos del usuario y las propiedades de los ítems mediante una definición explícita de reglas, además de seleccionar aquel ítem que ofrece mayor grado de *satisfacibilidad* de requisitos del usuario. Entre ambos enfoques seleccionamos el basado en restricciones, puesto que se ajusta mejor a la representación mediante reglas utilizada en la base de conocimiento propuesta.

Recordemos que una dieta puede pertenecer a más de una categoría (herencia múltiple), por lo que puede ser miembro de varias clases. Por cada categoría a la que pertenece una dieta existe una regla R_i que la relaciona con un requisito de la clase *Person* mediante la propiedad de objeto *has Recommended Diets*. Como ejemplo, supongamos que una persona P tiene estado nutricional *bajo-peso* y nivel de masticación *bajo*. Si existiera una dieta hipercalórica $d1$ y una dieta blanda $d2$, se dispararían las siguientes reglas:

Person(?P), 'has Nutritional State'(P, underweight), 'Hypocaloric diet'(d1) -> 'has Recommended Diets'(P, d1)

Person(?P), 'has Chewing Level'(P, 'low chewing level'), 'Blend diet'(d2) -> 'has Recommended Diets'(P, d2)

Si cada una de esas reglas sólo se disparara una vez y $d1=d2$ (es decir, las dietas son la misma), se recomendaría una única dieta, ya que cumpliría ambos requisitos. Sin embargo, si alguna de esas reglas se dispara varias veces (hay varias dietas que cumplen los requisitos) o bien $d1 \neq d2$, resultarían varias dietas a recomendar, por lo que habría que elegir la más adecuada. Con este fin, se define una variable que denominaremos *dietRating*, que nos indicará el valor de adecuación al usuario de cada dieta recomendada tras dispararse las reglas. Este valor se obtiene a partir del número de reglas que se han disparado (es decir, número de restricciones satisfechas) y la importancia de cada requisito del usuario que ha disparado la regla. Esta importancia se representa mediante *pesos* y se utiliza para relajar el cumplimiento de ciertas restricciones. Los pesos se han asignado explícitamente y toman valores en el intervalo $[0,1]$.

Además de considerar las restricciones satisfechas, las que no se cumplen también deben influenciar el *dietRating* de cada dieta, puesto que deberían decrementar su grado de satisfacción. En esta propuesta se consideran dos tipos de *no-satisfacción* de restricciones:

- no aparición de la propiedad que instancia el valor del requisito. Por ejemplo, un requisito es que la dieta sea baja en grasa, pero en la dieta contemplada no aparece la propiedad “contenido en grasas”.
- la aparición de la propiedad, pero con un valor diferente al requisito. Por ejemplo, en el caso del requisito anterior, la dieta contiene la propiedad “contenido en grasas” pero su valor es diferente a “bajo”.

La posición en la ontología determina, en el caso de la clasificación de dietas, que los diferentes valores de cada requisito son opuestos, por ejemplo, las dietas clasificadas como *restringidas según calorías* pueden ser *hipercalóricas* o *hipocalóricas*, que son contrapuestas entre sí.

Sea D el vector de satisfacción de restricciones de una dieta d :

$$D = \{ [S_{C_1}, \dots, S_{C_i}, \dots, S_{C_N}] \mid 1 \leq i \leq N, S_{C_i} \in [-1,1], d \xrightarrow{\text{hasType}} D_i, D_i \xrightarrow{\text{subclassOf}} C_i \}$$

donde:

- N es el número total de restricciones de usuario mapeadas a la taxonomía de dietas,
- D_i es la clase a la que pertenece la dieta, indicada con $\xrightarrow{\text{hasType}}$, que representa el valor de una restricción de usuario
- C_i es la restricción mapeada que a su vez es superclase de D_i y $\xrightarrow{\text{subclassOf}}$ denota la relación de subclase no-recursiva
- $S_{C_i} = R_i \cdot w_i$, es el grado de satisfacción de C_i con:
 - $R_i = 1$ si se dispara la regla que indica que D satisface la restricción de usuario en C_i o si no existe la restricción ni en el usuario ni en la dieta.
 - $R_i = 0$ si existe la restricción C_i , pero no se dispara ninguna regla que incluya descendientes de C_i .
 - $R_i = -1$ si se dispara una regla que indica que $d \xrightarrow{\text{hasType}} \text{Superclass}(D_i)$, pero no satisface la restricción de usuario C_i o no existe la restricción de usuario C_i , pero la dieta es descendiente de C_i .
 - w_i es el peso o factor de importancia de la restricción de usuario C_i .

En la figura 7.9 se muestra la estructura jerárquica de clasificación de dietas y cada uno de los elementos que intervienen en el cálculo del *dietRating*.

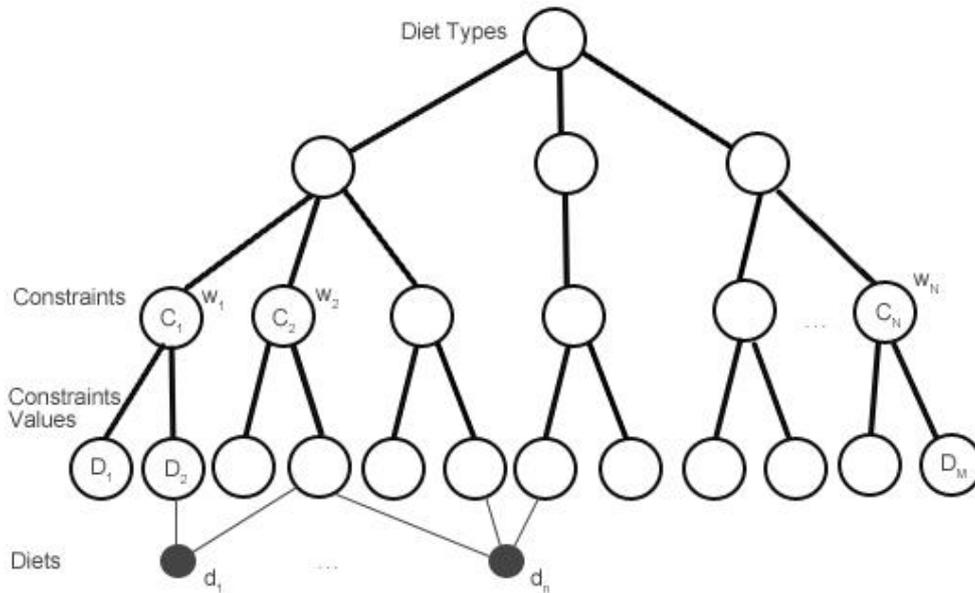


Figura 7.9. Estructura taxonómica de tipos de dietas y elementos considerados en el enfoque basado en restricciones.

A partir del vector D se define el valor *dietRating* para cada modelo de dieta d como la sumatoria del grado de satisfacción de cada requisito de usuario C_i , normalizado en el intervalo $[-1,1]$:

$$dietRating(d) = \frac{\sum_{i=1}^N S_{id}}{N} \quad (16)$$

Y finalmente, se recomendará el modelo de dieta con mayor *dietRating* obtenido:

$$RecommendedDiet(d) = \max \{dietRating(d)\} \quad (17)$$

Una vez seleccionado el modelo de dieta, el sistema chequea que no aparezca algún alimento o sustancia no recomendable, es decir, entre los considerados como *no-interesantes*, que, recordemos, eran aquellas instancias de las clases *Dislikes* e *Intolerances*. Si alguno de ellos apareciera, debe ser sustituido por el alimento nutricionalmente más similar que no pertenezca a estas clases. El proceso de obtención de la alternativa más similar se lleva a cabo mediante la metodología de recomendación basada en contenido que se explica en el siguiente sub-apartado. Tras haber sido sustituidos los alimentos no recomendables por sus alternativas más idóneas se instancia la dieta completa en la ontología.

7.6. Recomendación basada en Contenido. Obtención de Alternativas.

Cuando el usuario recibe la dieta recomendada a través de la interfaz puede solicitar alternativas a diferentes alimentos de la dieta que, por cualquier circunstancia (disponibilidad, apetencia, etc.), no desea tomar. El proceso de obtención de las alternativas más saludables, ajustadas a los gustos de los usuarios, se realiza mediante la técnica de recomendación basada en contenido que se fundamenta en la descripción del ítem y el perfil de intereses de los usuarios.

En la ontología propuesta, los ítems pueden ser instancias de dietas, platos o alimentos. En concreto, en este tipo de recomendación nos centramos en los alimentos contenidos en la

ontología, que vienen descritos, tanto por su posición taxonómica, como por sus propiedades y relaciones. Las recomendaciones se basan en la similaridad semántica entre los alimentos, rigiéndose además por las restricciones nutricionales que se contemplan en la confección de dietas saludables y por los requisitos y gustos (explícitos e implícitos) de los usuarios. Es importante aclarar que en la estrategia de recomendación definida podrían recomendarse alimentos (ingredientes) o platos completos. En esta propuesta, al tratarse de una alimentación más estricta dirigida a personas mayores, las recomendaciones se centran en los alimentos específicos de la ontología y no en los platos completos, sin embargo, la definición de la ontología, gracias al uso de propiedades que recogen la inclusión de ingredientes en los diferentes platos, permite adaptar esta recomendación de forma sencilla a platos completos del menú ajustando el método de recomendación.

En el capítulo 3 se presentaron los enfoques clásicos para el cálculo de la similaridad semántica entre conceptos de una ontología o un tesoro. A continuación, se aborda un análisis de estos enfoques para su adecuación a la recomendación de alimentos en nuestro sistema, referenciando también algunas de las nuevas tendencias aportadas por otros autores. Se detallarán las decisiones tomadas en el cálculo de la similaridad y sus motivaciones.

7.6.1 Análisis de Similaridad Semántica desde la Perspectiva del Sistema

Los enfoques para el cálculo de similaridad semántica clásicos estudiados en el capítulo 3 diferenciaban en la inclusión de las relaciones taxonómicas entre conceptos y, los grados de interés estaban basados principalmente en el análisis de frecuencia de aparición de sus propiedades. Aunque los primeros Sistemas de Recomendación Semánticos utilizaban estos enfoques clásicos, pronto resultaron ser insuficientemente precisos para muchos investigadores por diversos motivos, algunos de los cuales se recogen más adelante. Según (Albertoni and De Martino, 2008) no se puede decir que exista un enfoque general aplicable en todos los dominios. Estos autores afirman que el dominio de la aplicación influye en el enfoque de cálculo empleado en la similaridad y debe refinarse según las necesidades del sistema. Siguiendo este criterio, se han tomado diferentes decisiones en el cálculo de la similaridad para solventar algunas deficiencias de los enfoques iniciales ajustándolas a las necesidades de esta propuesta.

- **Enfoques puros.** La aplicación de un único tipo de enfoque de los citados en el capítulo 3 suele resultar en recomendaciones débiles, puesto que se ignoran diferentes tipos de relaciones que se pueden encontrar en la ontología, por ello, la mayoría de las propuestas más recientes utilizan enfoques híbridos que combinan y refinan las técnicas clásicas.
- **Distinción entre instancias y clases.** Según (Albertoni and De Martino, 2008) es importante definir a priori si las comparaciones se van a llevar a cabo entre conceptos de tipo *clase* o conceptos de tipo *instancia*, ya que la selección de un tipo u otro determinará la técnica a utilizar.
- **Grandes espacios de búsqueda.** En ontologías donde el número de conceptos es muy elevado, los cálculos de similaridad de un concepto con todos los demás puede resultar en tiempos de respuesta ineficientes. Según las necesidades del usuario en el sistema, es posible reducir el espacio de comparación según algún criterio. Existen dos técnicas básicas para reducir el espacio de comparación (Euzenat and Shvaiko, 2007): partición de la ontología (*partition*) que consiste en dividir la ontología en otras más pequeñas y, poda de la ontología (*pruning*) que consiste en ignorar dinámicamente partes de la ontología en el momento de la comparación. Reducir el espacio de búsqueda de forma efectiva resulta

en un menor número de comparaciones posibilitando una mejora en los tiempos de respuesta y una disminución de *falsos positivos* (Duan et al., 2012).

- **Exact matching.** Como se indica en (Blanco-Fernández et al., 2008a), una desventaja general de los métodos clásicos se encuentra en la búsqueda de coincidencias exactas en los procesos de *matching*, lo que conlleva que dos propiedades con valores similares no se consideren como un posible *match*. Este problema puede abordarse desde dos de las diferentes capas del *framework* de similaridad en ontologías propuesto por (Ehrig et al., 2005). La capa de datos abarca los datos de tipo cadena y numéricos. Supongamos que un alimento tiene 120 kcal. y otro alimento 121 kcal. En este caso no se detectaría *matching* aunque los valores sean casi idénticos, lo cual decrementaría su grado de similitud, sin embargo, si ambos estuvieran clasificados como de alto contenido calórico resultaría en un *matching* de tipo *exact match*, lo que aumentaría su similaridad. Supongamos ahora que un alimento *i* está clasificado como contenido calórico muy alto, otro alimento *j* se clasifica como contenido calórico alto y un tercer alimento *k* como contenido calórico muy bajo. Desde la perspectiva de la capa de datos, los tres alimentos se consideran diferentes en igual grado. Sin embargo, abstrayéndonos un nivel hacia la capa ontológica, la diferencia semántica entre los tres alimentos no es la misma puesto que los alimentos *i* y *j* son más similares entre sí que al compararlos con el alimento *k* clasificado como muy bajo.
- **Diferentes medidas y rangos de las propiedades de datos.** Puesto que las diferencias entre medidas y rangos en que se clasifican las diferentes propiedades comparadas entre instancias derivan en similaridades descompensadas, es necesario un proceso de normalización que transforme todas las medidas a un mismo rango de valores. La mayoría de los autores proponen una normalización en el rango $\{0,1\}$ (Schickel-Zuber and Faltings, 2007).
- **Herencia de intereses.** Los autores de (Blanco-Fernández et al., 2008a) detectan también deficiencias en el cálculo del grado de interés de un ítem al no contemplar el grado de interés de sus clases ancestrales. Para solventar estas carencias proponen los llamados índices DOI (Degree of Interest) que miden el interés del usuario sobre un ítem en el rango $\{-1,1\}$. Estos DOIs se aplican sobre los ítems, que heredan los DOIs de sus ancestros y, sobre los atributos de los ítems.
- **Definición adecuada de uso de frecuencias de aparición.** La frecuencia de aparición de un atributo de un ítem no tiene necesariamente que determinar unívocamente el grado de interés para el usuario. Esto se acentúa en el caso de los alimentos ya que, por ejemplo, el hecho de que un usuario haya seleccionado repetidas veces alimentos con valor proteínico elevado no tiene por qué implicar que esté interesado en que se le recomienden alimentos con valor proteínico de ese tipo. Esto conlleva la toma de decisiones acerca de qué clases de frecuencia analizar y sobre qué atributos o ítems se llevarán a cabo dependiendo del dominio y finalidad del sistema.

7.6.2 Cálculo de Similaridad Semántica en NutEiCare

El análisis expuesto previamente revela decisiones a tomar en el cálculo de la similaridad semántica durante este proceso de recomendación en nuestro sistema.

Resulta evidente que debe utilizarse un enfoque híbrido que combine técnicas basadas, tanto en las relaciones taxonómicas, como en las demás propiedades semánticas de los ítems. Puesto que en la estrategia de recomendación propuesta la comparación se va a realizar entre los

diferentes alimentos de la ontología, los cuales se han definido como instancias, el cálculo de la similaridad se va a efectuar entre conceptos de tipo instancia.

Reducción del Espacio de Búsqueda

No todas las instancias de la ontología de dominio son alimentos, sin embargo, sí que lo son las instancias descendientes de la clase *Food Products*, lo que supone una reducción inicial clara. Sería posible realizar una partición en la cual simplemente redujéramos la ontología a todas las instancias de la ontología de alimentos. Sin embargo, la definición de la ontología y la estrategia de recomendación definida permiten una disminución mayor del espacio de búsqueda dependiendo del alimento seleccionado a comparar. Dicho alimento y su posición en la ontología determinan el espacio de búsqueda de alimentos candidatos, por lo que se realiza una poda dinámica que englobe sólo los conceptos relevantes a comparar, disminuyendo la probabilidad de aparición de falsos positivos y mejorando los tiempos de respuesta.

En primer lugar, debe decidirse el criterio mediante el que se realizará la poda de la ontología, identificando como “puntos de parada” los nodos a partir de los cuales se llevará a cabo dicha poda. Guiándonos por la estrategia de recomendación propuesta se seleccionó una propiedad nutricional fundamental que restringía los criterios de recomendación. El diseño de nuestra ontología permite que todas las instancias a recomendar tengan algún ancestro en la jerarquía *is-a* que se corresponda con uno de esos puntos parada. Así, en el proceso de búsqueda de instancias similares a un alimento, se necesitaría ascender como máximo a ese punto de parada. Esta propiedad es *Recommended rations a week* que agrupa a los alimentos según la cantidad de raciones a ingerir semanalmente por dicho tipo de alimento. Por lo tanto, se definió como *nodo de parada* en la búsqueda de alimentos, la clase ancestral a la instancia por la cual venga restringida esta propiedad. Este nodo de parada será la clase donde se realice la poda, por lo que pasa a ser la clase raíz, cuyos descendientes forman el espacio de búsqueda en el proceso de *matching*.

Sea F el conjunto de todas las instancias alimento y f_s el alimento seleccionado para el proceso de *matching*. El nodo de parada, C_s , se determina como la superclase de f_s que restringe su aparición en la restricción *Recommended rations a week*:

$$C_s = \{ \text{Recommended rations a week } (C_s) \wedge C_s \xrightarrow{\text{hasMember}} f_s, f_s \in F \} \quad (18)$$

donde $F = [f_1, \dots, f_s, \dots, f_N]$ con $1 \leq s \leq N$ y $\xrightarrow{\text{hasMember}}$ denota la descendencia recursiva de instancias de una clase.

Finalmente, se define el espacio de búsqueda de alimentos similares como el subconjunto A de F al que denominaremos *Adyacentes* y que contiene todas las instancias descendientes de la clase de parada C_s :

$$A = \{ [f_1, \dots, f_i, \dots, f_n] \subseteq F \wedge f_i \xrightarrow{\text{typeOf}} C_s, i \neq s, 1 \leq i \leq n \leq N \} \quad (19)$$

donde $\xrightarrow{\text{typeOf}}$ denota la pertenencia recursiva a una clase.

Propiedades de Datos y de Objeto

Como se mencionó en la anterior sección, la comparación de las propiedades de datos compartidas únicamente mediante *exact matching* conlleva el no reconocimiento de diversas similitudes implícitas, por lo que debe definirse un método para el cálculo de dicha similaridad.

Una clasificación por rangos de valores numéricos permite identificar similitudes por la pertenencia a dichos rangos. Aprovechando las clasificaciones nutricionales de los expertos, se pueden definir dichas categorizaciones mediante valores umbrales que determinan la pertenencia a una u otra categoría. De este modo, surgen nuevas propiedades de objeto que denominaremos propiedades de nivel (Level Properties, LP) y que servirán para categorizar ciertas propiedades de datos numéricas permitiendo identificar nuevas relaciones de similitud implícitas. Por cada propiedad de datos numérica descendiente de *Nutritional Properties* que representa el contenido de algún nutriente por 100 gramos de alimento, se ha creado una propiedad de objeto de tipo LP descendiente a su vez de la propiedad *Nutritional Properties*⁶⁶. El rango de estas propiedades LP es la clase *LowtoHighValues* cuyas instancias son {VERY_LOW, LOW, MEDIUM, HIGH, VERY_HIGH}.

El uso de reglas SWRL permite incorporar estas clasificaciones en la base de conocimiento. En el capítulo 6 (apartado 6.7) se mostró un ejemplo del uso de reglas para clasificar la propiedad de datos *Calories per 100 gr* mediante la propiedad de nivel *Caloric Level*. Con esta tipificación, dos alimentos categorizados como de alto contenido calórico incrementan su grado de similitud. Sin embargo, aún no es posible inferir que dos alimentos con contenido *muy alto* y *alto* son más similares entre sí que comparados con uno de contenido *muy bajo*. Para ello, introducimos la anotación numérica *levelValue*⁶⁷ sobre cada una de las instancias que toma valores de 1 (VERY_LOW) a 5 (VERY_HIGH). De este modo, los valores numéricos iniciales, que venían restringidos por el tipo de medida, se normalizan a un valor discreto de 1 a 5 independientemente de la unidad de medida.

El resto de propiedades nutricionales de los alimentos las denominamos propiedades típicas (*Typical Properties*, TP).

Así, siendo C_{LH} la clase *LowtoHighValues* y PO el conjunto de todas las propiedades de objeto de la ontología de dominio, se define el conjunto de las propiedades de objeto nutricionales PON como las propiedades de objeto cuyo dominio es la clase *Food products* y se forma mediante la unión de los conjuntos de propiedades de objeto LP y TP:

$$PON = \{ LP \cup TP \mid LP \cap T = \emptyset, PON \xrightarrow{\text{subPropertyOf}} PO, PON \xrightarrow{\text{hasDomain}} F \} \quad (20)$$

donde:

$$LP = \{ [P_1, \dots, P_i, \dots, P_n] \mid P_i \xrightarrow{\text{hasRange}} C_{LH}, 1 \leq i \leq n, LP \xrightarrow{\text{subPropertyOf}} PON \} \quad (21)$$

$$TP = \{ [P_1, \dots, P_j, \dots, P_n] \mid 1 \leq j \leq n, j \neq i, TP \xrightarrow{\text{subPropertyOf}} PON \} \quad (22)$$

Las relaciones de herencia entre las propiedades añaden más valor semántico a la ontología permitiendo, además, cálculos de similitud más ajustados a las restricciones nutricionales.

⁶⁶ Aunque ambas propiedades *Nutritional Properties* tienen la misma etiqueta no son la misma propiedad ya que una es de datos y la otra de objeto y por tanto se representan mediante diferente IRI.

⁶⁷ Previamente se reutilizó la anotación predeterminada *isDefinedBy* pero posteriormente se creó una anotación específica *levelValue* sólo para este uso.

Distancia Taxonómica

El enfoque utilizado en el cálculo de similitud es un híbrido que combina la relación de subsunción entre clases e instancias de la ontología y las demás relaciones nutricionales que aparecen de forma explícita o implícita. Rada y colaboradores en (Rada et al., 1989) introdujeron el concepto de distancia como una métrica para medir el camino mínimo entre dos conceptos de una red semántica calculando el número mínimo de arcos que los separa. Basándonos en esta teoría definimos la distancia taxonómica entre dos instancias i y j mediante la ecuación:

$$distance(i, j) = \min|path(i, j)| = \min|path(i, ca_{ij})| + \min|path(j, ca_{ij})| \quad (23)$$

donde ca_{ij} es el ancestro común que subsume a i y j . Si dos instancias pertenecen a la misma clase, su distancia taxonómica se considera igual a 1, en otro caso, la distancia es la suma de los arcos recorridos. La figura 7.10 ilustra un ejemplo de cálculo de distancia taxonómica entre dos instancias i y j , donde las flechas azules indican el camino mínimo de subsunción de i al ancestro común y las flechas rojas desde j a dicho ancestro. En el ejemplo, la distancia entre ambas instancias es 7.

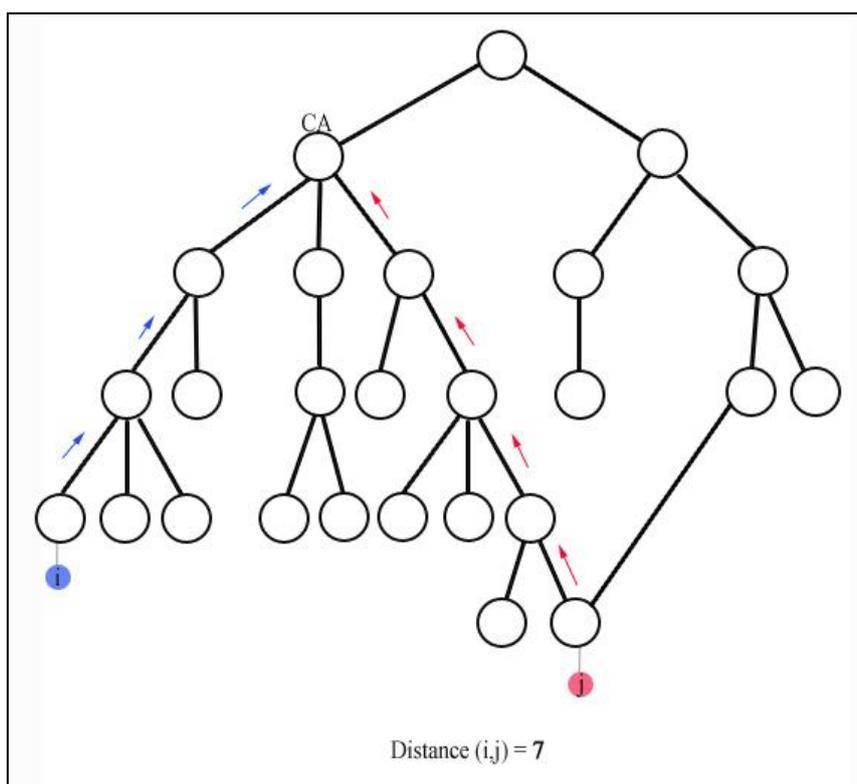


Figura 7.10. Ejemplo de distancia taxonómica entre dos instancias i y j .

Cálculo de la Similitud Semántica

Los cálculos de similitud se van a realizar contemplando las propiedades de objeto nutricionales de los alimentos, ya que las propiedades de datos han sido transformadas a propiedades de objeto (propiedades de nivel, LP).

Puesto que las propiedades LP son propiedades funcionales (sólo pueden tomar un valor), se asignan pesos normalizados en el intervalo $[0,1]$ en función de los valores de sus propiedades

de datos, suponiendo, además, que todas tienen la misma probabilidad de aparición. El peso en función de cada LP comparada entre f y j se obtiene mediante:

$$w_{LP}(f, j) = \frac{N_{LP} - abs(v_{LP_f} - v_{LP_j})}{N_{LP}} \in [0,1] \quad (24)$$

donde N_{LP} es el número de posibles valores de la propiedad de nivel LP y v_{LP} es el valor de la anotación *levelValue* para cada una de las dos instancias. Finalmente, para cada instancia j que se está comparando con el alimento f respecto a las propiedades de nivel, se construye un vector de pesos $V_{LP_j} = \{w_{LP}(1, j), w_{LP}(2, j), \dots, w_{LP}(n, j)\}$.

El resto de las propiedades de objeto a comparar son las propiedades típicas, TP , que son no-funcionales. Por ejemplo, la propiedad *Contains*, que puede tomar varios valores, como *Gluten* o *Lactose*. Para cada propiedad TP , se obtienen los *exact match* (TP_{fj}) de dicha propiedad de entre los N_{TP} valores que contiene dentro de la instancia y se calcula el peso $w_{TP}(f, i)$ de esa propiedad en j mediante una versión ligeramente adaptada de la técnica TF-IDF vista en el capítulo 3:

$$\begin{aligned} TF_{TP_j}(f, j) &= \frac{TP_{fj}}{N_{TP_j}} ; IDF_{TP_j}(f, j) = \log \frac{N}{n_{TP_f}} \\ w_{TP}(f, j) &= TF_{TP_j}(f, j) \cdot IDF_{TP_j}(f, j) \end{aligned} \quad (25)$$

donde TP_{fj} es el número de coincidencias de la propiedad, N_{TP_j} es el número máximo de valores que toma la propiedad TP en j , N es el número de instancias adyacentes y n_{TP_f} es el número de adyacentes donde coincide la propiedad TP .

Para cada instancia j que se está comparando con el alimento f respecto a las propiedades típicas, se construye un vector de pesos $V_{TP_j} = \{w_{TP}(1, j), w_{TP}(2, j), \dots, w_{TP}(m, j)\}$.

Finalmente, se obtiene la similaridad semántica total entre f y j mediante la media aritmética de los vectores de pesos en sus propiedades LP y TP y dividiendo entre su distancia taxonómica.

$$SIM(f, j) = \frac{\frac{1}{T} \sum_{i=1}^N w_{LP_i}(f, j) + \frac{1}{T} \sum_{l=1}^M w_{TP_l}(f, j)}{distance(f, j)} \in [0,1] \quad (26)$$

donde T es el número total de propiedades comparadas. Partiendo de estos valores de similaridad se obtiene una lista de recomendación de alimentos R'_f ordenada de mayor a menor similaridad.

A continuación se explica el ajuste de cada uno de estos valores de similaridad utilizando los gustos del usuario aprendidos durante el uso del sistema.

Aprendizaje del Grado de Interés del Usuario sobre un Ítem

La primera vez que el usuario va a recibir recomendaciones sobre alimentos, su grado de interés sobre un alimento es 0 si está clasificado en *Dislikes* y 0.5 en otro caso. Conforme el usuario utiliza el sistema y va realizando selecciones, está proporcionando *feedback* relevante para el aprendizaje de sus gustos, por lo que su grado de interés se va modificando. Este comportamiento se analiza para extraer patrones de comportamiento del usuario de forma implícita en función de las recomendaciones seguidas. Así, cuando un alimento no es

seleccionado recibe una penalización que disminuye su probabilidad de aparición en próximas recomendaciones.

En particular, en cada recomendación sobre un determinado alimento f a un usuario se gestiona la lista R_f obtenida en el paso anterior, como un vector con N elementos:

$$R_f = \{R_1, R_2, \dots, R_i, \dots, R_N\} = \{(A_1, \text{Sim}_1, \mu_1), (A_2, \text{Sim}_2, \mu_2), \dots, (A_i, \text{Sim}_i, \mu_i), \dots, (A_N, \text{Sim}_N, \mu_N)\}$$

donde N es el número de alternativas posibles (número de adyacentes) comparadas para el reemplazo de f , A_i cada una de las instancias adyacentes, Sim_i es la similaridad semántica de A_i con f y μ_i es el factor de interés del usuario sobre A_i obtenido mediante la siguiente ecuación:

$$\mu_i = \frac{fs}{fr \cdot i} \in [0,1] \quad (27)$$

Siendo:

i → posición de A en el vector de recomendaciones R_f .

fs → número de veces seleccionado como alternativa de f (valor de la propiedad *timesSelected*).

fr → número de veces recomendado como alternativa de f (valor de *timesRecommended*).

El factor de interés del usuario, μ_i , reajusta los valores de similaridad con los gustos inferidos obteniéndose un nuevo vector R'_f con N elementos ordenados de mayor a menor Sim'_i :

$$R'_f = \{R'_1, R'_2, \dots, R'_i, \dots, R'_N\} = \{(A_1, \text{Sim}'_1), (A_2, \text{Sim}'_2), \dots, (A_k, \text{Sim}'_i), \dots, (A_N, \text{Sim}'_N)\}$$

Donde:

$$\text{Sim}'_i = \text{Sim}_i + \mu_i \in [0,2] \quad (28)$$

Obtención de la Lista de Recomendación

Una vez obtenidos los valores para el grado de similaridad de cada instancia adyacente de f , el componente de Recomendación devuelve una lista $R_f = \{f_1, \dots, f_s, \dots, f_K\}$ conteniendo los Top- K alimentos más similares ordenados de mayor a menor similaridad, es decir, los K primeros alimentos de R'_f . El valor de K será 1 si se envía al componente de dietas y K mayor que 1 si se envía a la interfaz de usuario. En el caso de nuestra propuesta, el valor inicial de K es 5, puesto que se considera que 5 alternativas son adecuadas para mantenerse dentro del rango de alimentos *suficientemente* similares, pero se permite al usuario solicitar más alternativas si lo desea.

7.7. Aplicación de la Estrategia de Recomendación.

Para demostrar la aplicabilidad de la estrategia de recomendación nutricional propuesta, en este apartado se presenta un ejemplo completo del razonamiento que tiene lugar al aplicar dicha estrategia sobre la base de conocimiento de NutElCare. Por simplificación, en algunos casos se muestran sólo parte de las clases, propiedades o restricciones que actúan en los diferentes procesos y para una mejor comprensión la mayoría de los ejemplos se muestran en Protégé.

Supongamos que el usuario identificado en el sistema, *Mary*, se instancia en la ontología con las propiedades de la Figura 7.11. El primer paso a realizar por el sistema es recomendarle

una dieta ajustada a su perfil de usuario. Para ello, el Componente de Dietas lleva a cabo la recomendación basada en conocimiento explicada en el apartado anterior.



Figura 7.11. Propiedades instanciadas del usuario del sistema para la recomendación de dietas.

Supongamos que existen 50 dietas en el repositorio (clasificables en 12 clases de restricción C_i diferentes, como se muestra en la figura 7.12) y que 20 de ellas han disparado alguna regla. Por lo tanto, tenemos 20 dietas candidatas, las cuales se instancian como valores de la propiedad de objeto *Recommended Diets*. Para cada una de ellas se calcula su *dietRating* y se obtiene la de mayor valor.

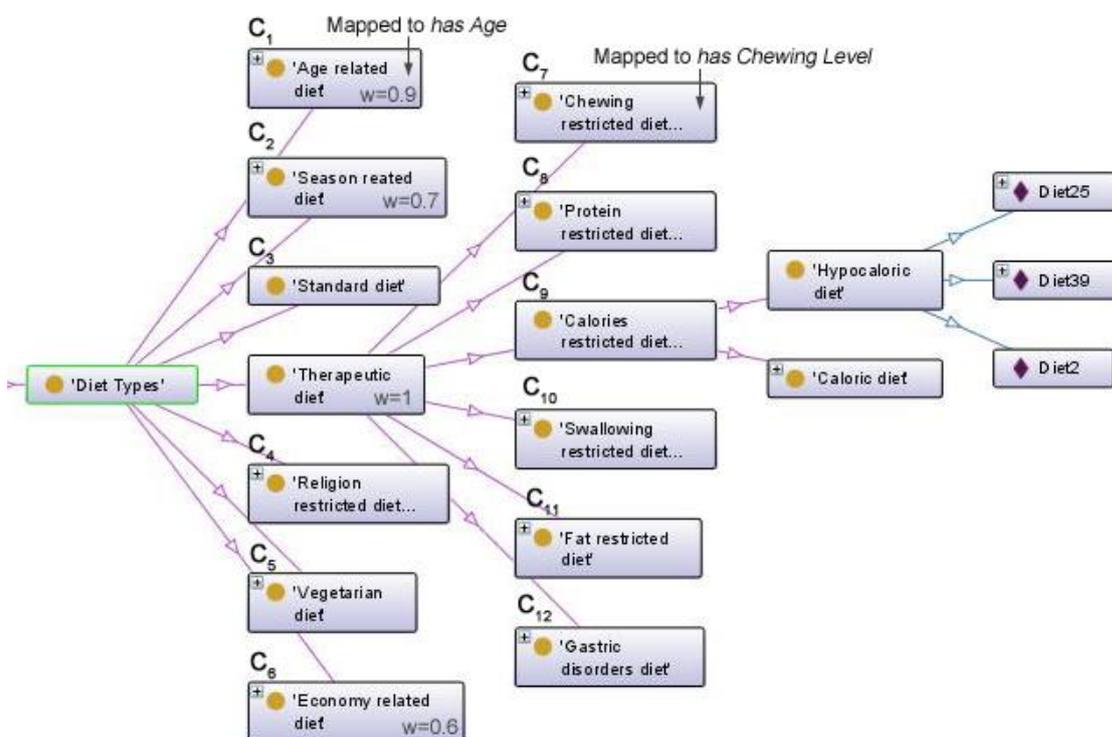


Figura 7.12. Ejemplo de clasificación de dietas según restricciones de usuario con valores de pesos sobre las restricciones.

A continuación, se muestra un ejemplo de cálculo de *dietRating* para dos dietas candidatas identificadas como #Diet11 y #Diet39. En la figura 7.13 se muestra la clasificación de ambas dietas en la ontología mediante la relación *hasType* que indica la pertenencia de una instancia a una clase D_i .

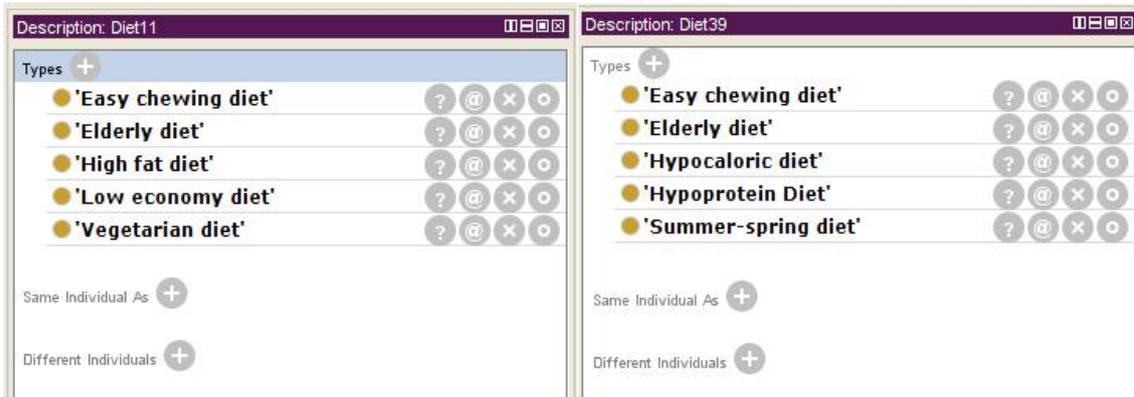


Figura 7.13. Clasificación de las dietas #Diet11 y #Diet39.

Cada dieta se representa mediante el vector de satisfacción de restricciones:

$D = [S_{C_1}, \dots, S_{C_i}, \dots, S_{C_{12}}]$ donde S_{C_i} denota el grado de cumplimiento de la restricción i del usuario que hace *matching* con la clase C_i .

La representación de dichos vectores para #Diet11 y #Diet39:

$Diet11 = [0.9, 0, 0, 0, -0.8, 0, 1, 0, 0, 0, -1, 0]$. Se han disparado 2 reglas que coinciden con los valores positivos del vector:

R_1 : Person(?p), 'Elderly diet'(?d), 'has Group Age'(?p, Elder) -> 'Recommended Diets' (?p, ?d) ,

R_2 : Person(?p), 'Easy chewing diet'(?d), 'has Chewing Level'(?p, LOW) -> 'Recommended Diets' (?p, ?d)

$Diet39 = [0.9, -0.7, 0, 0, 0, 1, 1, 1, 0, 0, 0, 0]$. Se han disparado 4 reglas:

R_1 , R_2 ,

R_5 : Person(?p), 'Hypocaloric diet'(?d), 'has Nutritional State'(?p, overweight) -> 'has Recommended Diets' (?p, ?d) ,

R_6 : 'Hypoproteic diet'(?d), Person(?p), 'has Activity'(?p, SedentaryActivity) -> 'has Recommended Diets' (?p, ?d)

A partir de los vectores se obtiene la utilidad de cada dieta para el usuario mediante el cálculo de su *dietRating*:

$$dietRating (Diet11) = \frac{\sum_{i=1}^{12} S_{C_i}}{12} = 0.008$$

$$dietRating (Diet39) = \frac{\sum_{i=1}^{12} S_{C_i}}{12} = 0.266$$

Suponiendo que el modelo de dieta #Diet39 ha obtenido el mayor *rating* de entre las 20 candidatas, éste será el modelo seleccionado por el *administrador de dietas*. Si el sistema considera que el contenido de la dieta se ajusta a los intereses del usuario, se instancia la dieta en la ontología y se manda a la interfaz de usuario.

Si Mary decide cambiar un alimento de la dieta, por ejemplo, la pechuga de pollo, cuyo nombre de instancia es *Chicken thigh*, lo solicita al sistema, el cual debe calcular los Top-K alimentos más similares de entre las posibles alternativas. En este momento, empieza el proceso

de recomendación basado en contenido entre alimentos, teniendo en cuenta los intereses (gustos y contraindicaciones) del usuario.

En primer lugar, se obtienen los adyacentes a *Chicken thigh* para podar la ontología reduciendo el espacio de búsqueda. Recordemos que las instancias adyacentes son los descendientes del nodo de parada. Este nodo se obtiene observando la restricción *Recommended rations a week* sobre una de sus clases ancestrales, que en este caso es la clase *Fresh Meat*, como se aprecia en la figura 7.14.

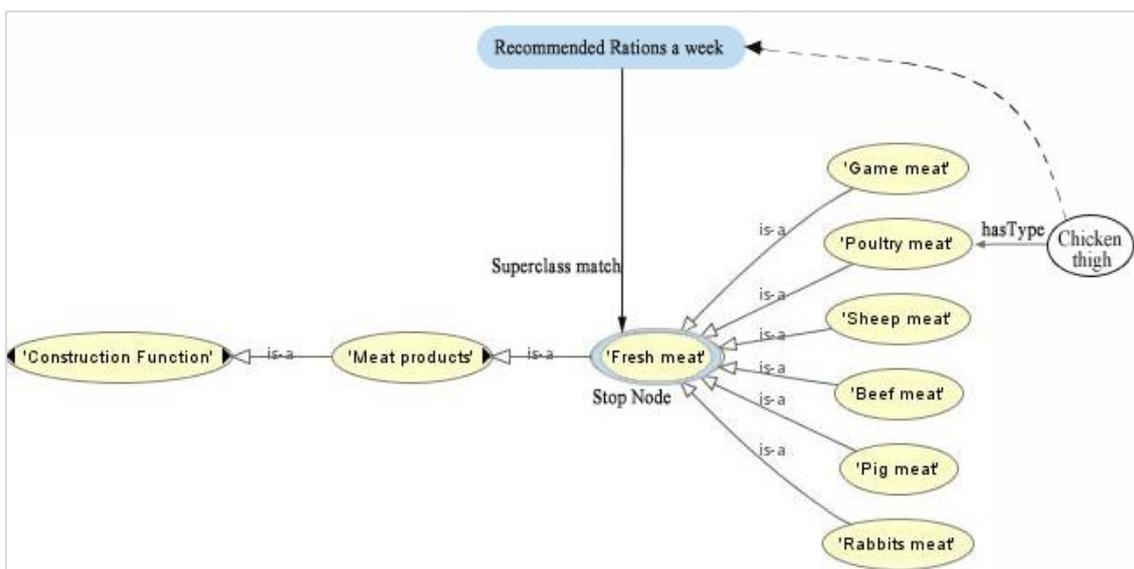


Figura 7.14. Selección de la clase “Fresh Meat” como nodo de parada y sus clases descendientes.

Los cálculos de similaridad semántica se llevarán a cabo entre los descendientes de *Fresh Meat*, excluyendo los alimentos contraindicados. Es decir, se eliminan de la lista inicial de adyacentes:

- Todas las instancias que se infieran como *not Can Eat*, es decir, que llevan a inconsistencia de la ontología al añadirlos como miembros de *has Recommended Food* (como en el ejemplo del apartado 7.2.2.). En este ejemplo se excluye la clase carne de caza, *Game meat*.
- Todas las instancias de alimentos que hayan sido consumidos dos veces en esa semana.
- Todas las instancias de alimento pertenecientes a la clase *Intolerances*.

Supongamos que se ha indicado en la ontología que a las personas mayores no se les puede recomendar *casquería*. Mediante la siguiente regla SWRL, se añaden como miembros de la clase *Intolerances* a todas aquellas carnes cuyo corte (propiedad *Cut as*) pertenece a la clase *casquería (Offal)*:

Person(?P), 'has Group Age'(?P, Elder), Offal(?O), 'Meat products'(?M), 'Cut as'(?M, ?O) -> Intolerances(?M).

Una vez eliminados los alimentos no recomendables, para cada uno de los adyacentes de la nueva lista se obtiene su distancia taxonómica como se muestra en la figura 7.15.

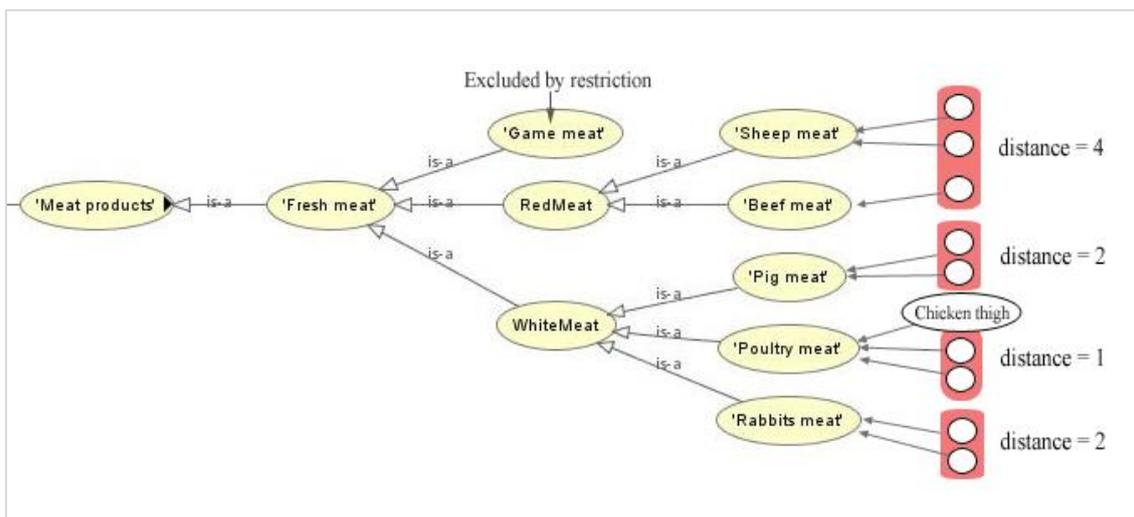


Figura 7.15. Instancias adyacentes a “Chicken thigh” y su distancia taxonómica.

Para cada alimento de entre los candidatos se calcula su similaridad para las propiedades LP y TP. Como ejemplo, calculemos la similaridad de *Chicken thigh* con su instancia adyacente *Pork shoulder*, que es miembro de la clase *Pig meat*. En primer lugar, se obtienen los pesos resultantes de la comparación entre las propiedades LP, cuyos resultados se muestran en la figura 7.16, donde se marca en rojo el ejemplo de cálculo de peso para la propiedad *Caloric Level* entre ambas instancias:

$$w_{\text{Caloric Level}}(\text{Chicken Thigh}, \text{Pork Shoulder}) = \frac{5 - \text{abs}(3 - 4)}{5} = 0.8$$

El vector de pesos de propiedades LP para la instancia adyacente:

$$V_{LP_{\text{Pork Shoulder}}} = \{0.8, 1.0, 0.8, 0.8, 1.0, 1.0, 1.0, 1.0, 1.0\}$$

```

INSTANCE Chicken thigh
L.PROPERTY: Caloric level VALUE: MEDIUM LEVEL: 3
L.PROPERTY: Iron level VALUE: LOW LEVEL: 2
L.PROPERTY: Protein level VALUE: VERY_HIGH LEVEL: 5
L.PROPERTY: Fat level VALUE: VERY_LOW LEVEL: 1
L.PROPERTY: Calcium level VALUE: VERY_LOW LEVEL: 1
L.PROPERTY: Cholesterol level VALUE: MEDIUM LEVEL: 3
L.PROPERTY: Sodium level VALUE: VERY_LOW LEVEL: 1
L.PROPERTY: Fibre level VALUE: VERY_LOW LEVEL: 1
L.PROPERTY: Carbohydrates level VALUE: VERY_LOW LEVEL: 1

INSTANCE Pork shoulder
L.PROPERTY: Caloric level VALUE: HIGH LEVEL: 4 WEIGHTLP: 0.8
L.PROPERTY: Iron level VALUE: LOW LEVEL: 2 WEIGHTLP: 1.0
L.PROPERTY: Protein level VALUE: HIGH LEVEL: 4 WEIGHTLP: 0.8
L.PROPERTY: Fat level VALUE: LOW LEVEL: 2 WEIGHTLP: 0.8
L.PROPERTY: Calcium level VALUE: VERY_LOW LEVEL: 1 WEIGHTLP: 1.0
L.PROPERTY: Cholesterol level VALUE: MEDIUM LEVEL: 3 WEIGHTLP: 1.0
L.PROPERTY: Sodium level VALUE: VERY_LOW LEVEL: 1 WEIGHTLP: 1.0
L.PROPERTY: Fibre level VALUE: VERY_LOW LEVEL: 1 WEIGHTLP: 1.0
L.PROPERTY: Carbohydrates level VALUE: VERY_LOW LEVEL: 1 WEIGHTLP: 1.0
  
```

Figura 7.16. Cálculo de pesos en propiedades LP entre “Chicken thigh” y su adyacente “Pork shoulder”.

A continuación, han de obtenerse los pesos para las propiedades TP. En la figura 7.17 se muestran los valores TF_IDF para cada una de las propiedades TP que se comparan entre *Chicken thigh* y *Pork shoulder*.

```

INSTANCE Pork shoulder
Property: Contains Total: 0.0 Matches: 0.0 TF: 0.0 Adyacent Matches:0.0 IDF: 0.0 TF IDF: 0.0
Property: Cut as Total: 2.0 Matches: 1.0 TF: 0.5 Adyacent Matches:2.0 IDF: 1.146128 TF IDF: 0.573064
Property: Cooked as Total: 1.0 Matches: 1.0 TF: 1.0 Adyacent Matches:12.0 IDF: 0.36797678 TF IDF: 0.36797678
Property: Is rich in Total: 1.0 Matches: 0.0 TF: 0.0 Adyacent Matches:12.0 IDF: 0.36797678 TF IDF: 0.0
Property: has Intake Dish Total: 0.0 Matches: 0.0 TF: 0.0 Adyacent Matches:0.0 IDF: 0.0 TF IDF: 0.0

```

Figura 7.17. Cálculo de pesos en propiedades TP entre “Chicken thigh” y su adyacente “Pork shoulder”.

Como ejemplo se ilustra el cálculo de la propiedad *Cut as*, marcada en rojo en la figura. El valor de la dicha propiedad para la instancia adyacente es 2: *Leg* y *Shoulder*. El número de coincidencias con el alimento original es 1: *Leg* y por tanto el valor de TF es:

$$TF_{Cut\ as}(Chicken\ thigh, Pork\ shoulder) = \frac{1}{2} = 0.5$$

Para el cálculo de IDF, se toma N como el número de instancias adyacentes candidatas, que en este caso es 28, y n_{TP_f} como el número de coincidencias de la propiedad *Cut as* en los demás adyacentes, que en este caso es 2. Por lo tanto el valor de IDF es:

$$IDF_{Cut\ as}(Chicken\ thigh, Pork\ shoulder) = \log \frac{28}{2} = 1.146128$$

Finalmente:

$$w_{Cut\ as}(Chicken\ thigh, Pork\ shoulder) = 0.5 \cdot 1.146128 = 0.573064$$

El vector de pesos de propiedades LP para la instancia adyacente:

$$V_{TP_{Pork\ Shoulder}} = \{0.0, 0.573064, 0.36797678, 0.0, 0.0, \}$$

En en la figura 7.15 vemos que la distancia taxonómica entre ambas instancias es 2. Así, la similaridad semántica entre el alimento original y el candidato se obtiene con la fórmula (12):

$$SIM = \frac{\frac{1}{T} \sum_{i=1}^N w_{LP_i}(f, j) + \frac{1}{T} \sum_{l=1}^M w_{TP_l}(f, j)}{distance(f, j)} = \frac{\frac{1}{14} \cdot 8.4 + \frac{1}{14} \cdot 0.94104078}{2} = 0.3336086$$

Una vez obtenida la similaridad para todas las instancias adyacentes se construye el vector R_f ordenado de mayor a menor similaridad. Para cada uno de esos valores, y usando la posición del alimento candidato en la lista, se calcula el factor de interés observando sus propiedades *times Selected* y *times Recommended*. En la figura 7.18 se muestran sólo las 10 primeras instancias del vector $R_{Chicken\ thigh}$ (recordar que son 28 elementos en total) y encuadrado en rojo el factor de interés del adyacente ejemplificado obtenido como:

$$\mu_{Pork\ shoulder} = \frac{ts}{fr \cdot k} = \frac{1}{3 \cdot 6} = 0.055555556$$

```

FIRST LIST
Pos: 1: Turkey thigh Initial Similarity: 0.8502336 Interest factor: 0.8333333
Pos: 2: Chicken breast Initial Similarity: 0.72986877 Interest factor: 0.5
Pos: 3: Turkey breast Initial Similarity: 0.66914123 Interest factor: 0.22222222
Pos: 4: Duck meat Initial Similarity: 0.6119984 Interest factor: 0.041666668
Pos: 5: Chicken wings Initial Similarity: 0.59999996 Interest factor: 0.05
Pos: 6: Pork shoulder Initial Similarity: 0.3336086 Interest factor: 0.055555556
Pos: 7: Pork leg Initial Similarity: 0.332573 Interest factor: 0.5
Pos: 8: Pork fillet Initial Similarity: 0.32142857 Interest factor: 0.05
Pos: 9: Pork lean Initial Similarity: 0.31428573 Interest factor: 0.5
Pos: 10: Pork ribs Initial Similarity: 0.30714288 Interest factor: 0.5

```

Figura 7.18. Diez primeras instancias más similares a “Chicken thigh” previamente al uso del factor de interés.

Sumando a cada similaridad de instancia su factor de interés, se reordena la lista con sus valores finales, de nuevo, de mayor a menor, como se muestra en la figura 7.19, resultando en el vector $R'_{Chicken\ thigh}$. Finalmente, se devuelve al usuario la lista $R_{Chicken\ thigh}$ con los K alimentos más similares ($K = 5$).

RECOMMENDATION LIST (using interest factor)	
Turkey thigh	1.6835669
Chicken breast	1.2298688
Turkey breast	0.89136344
Pork leg	0.832573
Pork lean	0.81428576
Pork ribs	0.80714285
Rabbit meat	0.80714285
Pork chop	0.79999995
Pork sirloin	0.79999995
Pork pancetta	0.7357143

} Top K (K=5)

Figura 7.19. Diez primeras instancias más similares a “Chicken thigh” ajustadas al gusto del usuario, al cual se le devuelven las Top K más similares, siendo $K=5$.

En este momento, el usuario puede seleccionar una de las recomendaciones o solicitar más, en cuyo caso el sistema le ofrece el siguiente valor de la lista de recomendación obtenida. La acción que realice el usuario sobre esta recomendación será registrada en el sistema y servirá para futuras recomendaciones, seguimiento de ingestas y análisis de consumo.

Mediante el ejemplo propuesto se ha demostrado la aplicabilidad de la estrategia de recomendación propuesta para explotar la base de conocimiento semántica de NutElCare.

Referencias del capítulo

- ALBERTONI, R. & DE MARTINO, M. 2008. Asymmetric and context-dependent semantic similarity among ontology instances. *In: HEIDELBERG, S. B. (ed.) Journal on data semantics X.*
- BLANCO-FERNÁNDEZ, Y., PAZOS-ARIAS, J. J., GIL-SOLLA, A., RAMOS-CABRER, M., LÓPEZ-NORES, M., GARCÍA-DUQUE, J. & BERMEJO-MUÑOZ, J. 2008. A flexible semantic inference methodology to reason about user preferences in knowledge-based recommender systems. *Knowledge-Based Systems*, 21, 305-320.
- BURKE, R. 2000. Knowledge-based Recommender Systems. *In: KENT, A. (ed.) Encyclopedia of Library and Information Systems.* New York: Marcel Dekker.
- DUAN, S., FOKOUE, A., HASSANZADEH, O., KEMENTSIETSIDIS, A., SRINIVAS, K. & WARD, M. J. 2012. Instance-based matching of large ontologies using locality-sensitive hashing. *The Semantic Web-ISWC 2012.* Springer Berlin Heidelberg.
- EHRIG, M., HAASE, P., HEFKE, M. & STOJANOVIC, N. Similarity for ontologies-a comprehensive framework. ECIS 2005, 2005 Regensburg, Germany
- ESPÍN, V., HURTADO, M. V. & NOGUERA, M. 2015. Nutrition for Elder Care: a nutritional semantic recommender system for the elderly. *Expert Systems*, n/a-n/a.
- ESPÍN, V., HURTADO, M. V., NOGUERA, M. & BENGHAZI, K. Semantic-Based Recommendation of Nutrition Diets for the Elderly from Agroalimentary Thesauri. *Flexible Query Answering Systems*, 2013. Springer Berlin Heidelberg, 471-482.
- EUZENAT, J. & SHVAIKO, P. 2007. *Ontology matching*, Heidelberg Springer.
- GRAU, B. C., HORROCKS, I., MOTIK, B., PARSIA, B., PATEL-SCHNEIDER, P. & SATTLER, U. 2008. OWL 2: The next step for OWL. *Web Semantics: Science, Services and Agents on the World Wide Web*, 6, 309-322.
- HORRIDGE, M. & BECHHOFFER, S. 2011. The OWL API: A Java API for OWL ontologies. *Semantic Web*, 2, 11-21.

- HORROCKS, I., PATEL-SCHNEIDER, P. F., BOLEY, H., TABEL, S., GROSOFF, B. & DEAN, M. 2004. *SWRL: A semantic web rule language combining OWL and RuleML* [Online]. W3C Member submission. Available: <http://www.w3.org/Submission/SWRL/> [Accessed 30 November 2015].
- LOPS, P., GEMMIS, M. D. & SEMERARO, G. 2011. Content-based recommender systems: State of the art and trends. *Recommender systems handbook*. Springer US.
- MOONEY, R. J. & ROY, L. Content-based book recommending using learning for text categorization. Fifth ACM conference on Digital libraries, DL '00, 2000 New York, USA. ACM, 195–204.
- RADA, R., MILI, H., BICKNELL, E. & BLETNER, M. 1989. Development and application of a metric on semantic nets. *IEEE Transactions on Systems, Man and Cybernetics*, 19, 17-30.
- SCHICKEL-ZUBER, V. & FALTINGS, B. 2007. OSS: A Semantic Similarity Function based on Hierarchical Ontologies. *IJCAI* 7, 551-556.
- SHEARER, R., MOTIK, B. & HORROCKS, I. 2008. HermiT: A Highly-Efficient OWL Reasoner. *OWLED*, 432, 91.
- SIRIN, E., PARSIA, B., GRAU, B. C., KALYANPUR, A. & KATZ, Y. 2007. Pellet: A practical owl-dl reasoner. *Web Semantics: science, services and agents on the World Wide Web*, 5, 51-53.
- TSARKOV, D. & HORROCKS, I. 2006. FaCT++ description logic reasoner: System description. *Automated reasoning*. Springer Berlin Heidelberg, 292-297.

CHAPTER 8

CONCLUSIONS & ONGOING WORK

8.1. Conclusions

Nutrition is an important concern of health for all people in general, and still more critical for the elderly. Nutritional assessments and recommendations can prevent and even treat the adverse effects of ageing, contributing to the well-being of this portion of the population.

The large volume of data available online about *nutritional stuff* sustains the enormous potential of the Web in this domain, but also unveils certain weaknesses. In many cases, deciding whether a nutritional information source is reliable or not, is not a trivial task. In addition, this wealth of information can be difficult to assimilate by non-expert users due to, among other factors, the complexity of managing this large volume of data to suit each individual's specific needs, the disparity in the reliability of the sources or the completeness of the information described.

Recent awareness and familiarity of the elderly with new technologies encourages the development of tools they can harness for their own benefit. However, many of the current proposals providing ageing support are intended for telemonitoring purposes without intervention of the users. The lack of systems that elderly can use by themselves to deal with age effects, particularly, in the nutritional domain, has been one of the main motivations of this thesis.

Recommender Systems (RS) are counselling tools which generate recommendations on a particular object of study using different techniques, trying to satisfy the needs and preferences of users (Herrera-Viedma et al., 2004). Research on RS constitutes a problem-rich research area because of the abundance of practical applications that help users to deal with information overload and provide personalized recommendations (Adomavicius and Tuzhilin, 2005), so that, this kind of systems can help to overcome the aforementioned drawbacks when expert knowledge is transferred into them. An introduction to RS has been presented in chapter 1.

However, traditional RS suffer from several shortcomings. One of the most important limitations is the heterogeneity in data representation, which hinders the reuse of knowledge already provided by expert sources and the interoperability or cooperation between different entities in the same domain. In order to overcome heterogeneity (among other benefits), several authors propose the inclusion of Semantic Web technologies in the RS development (Ziegler, 2005), (Loizou and Dasmahapatra, 2006) and (Lops et al., 2011).

In chapter 2, we have presented an study of Semantic Web Technologies (SWT) and how some technologies, such as thesauri and ontologies, allow formal and shareable representations of information to be provided, in such a way that fosters the reuse and cooperation and benefits from the recent Linked Open Data (Bizer et al., 2009) view of the Web. In addition, ontologies represented in the OWL language support reasoning through semantic reasoners, such as Pellet

(Sirin et al., 2007) allowing the inference of new knowledge which is not expressed explicitly in the ontology. This reasoning capacity implies an important advantage over traditional data representation as relational databases.

Next, in chapter 3, we have carried out a study of the application of the Semantic Web technologies in the development of RS, resulting in the so called Semantic Recommender Systems (SRS). We have analysed some of the different techniques used in the literature, starting with the general semantic similarity calculation techniques and finishing with some SRS general proposals and examples.

Chapter 4 introduces the main challenges to deal with when tackling the development of a RS in the nutritional domain considering, in particular, the health restrictions of target users:

- an effective user representation in the system,
- retrospective methods to recognize the impact that the system's utilisation has on the users, such as possible changes in their nutritional state,
- keep track of previous user's ingestions and diet habits identifying also nutritional deficits and trying to correct them,
- recommendations must always consider healthy restrictions and user preferences,
- user's non-compliance, or an incorrect interaction with the system.

Throughout this chapter, we have also evaluated different approaches proposed by other researchers in order to cope with these challenges, together with some nutritional recommendation strategies, identifying several lacks in the proposals, particularly, when providing holistic solutions.

After the study of the background related to Semantic Recommender Systems, its application to the nutritional domain, and the recognition of the challenges and goals to achieve when developing this kind of systems, we have proceed to the description of the development of our proposal in the second part of this work.

Thus, subsequent chapters focus on the implementation of NutElCare, a SRS that provides nutritional recommendations to the elderly. It is important to note that every relevant decision related to nutritional information representation and subsequent recommendations has been guided by expert nutritionists and geriatricians who have cooperated in this work.

The main contributions contained herein can be summarized as follows:

- We have proposed a system architecture which includes nutritional knowledge that have been imported from expert sources of information. The intensive application of SWT allows the reuse of expert external knowledge in the nutritional domain, fostering also cooperation with semantic entities in the Linked Open Data environment.
- We have formalized the NutElCare conceptual framework by its specification in a *domain ontology* and represented as an OWL 2 ontology, therefore enabling the automatic inference of implicit new relevant knowledge to be used in food recommendation processes. Moreover, this declarative approach boosts the inference of the same facts when the ontology changes. For instance, this system could be adapted to different sectors of the population just customizing the ontology, mainly adapting the restrictions and rules included in the ontology to produce healthy recommendations
- Furthermore, the ontology serves as a reference conceptual structure from which to start describing particular recommendation strategies in the nutritional domain through a more specific *application ontology*. The ontology specified in the knowledge base

comprises some smaller ontologies, namely, nutritional, diets and user profile ontologies. They can also be reused as single ontologies:

- For reusability, internationalization and institutionalization purposes, the nutritional ontology is built from the Agrovoc FAO Thesaurus (Caracciolo et al., 2013) of the United Nations. After a transformation process of the thesaurus and the enrichment with a nutritional oriented taxonomy and nutritional properties of food, we have provided an OWL 2 food & nutrition ontology to be reusable in the nutritional domain. The links to Agrovoc concepts are maintained in the ontology.
- The user profile ontology has been made as reusable as possible to serve in other applications or systems in the nutritional recommendations domain with minor or no changes. It has been developed extending GUMO -Generic User Model Ontology- (Heckmann et al., 2005) with new properties and concepts related to the nutritional domain.
- The diets ontology is designed from the XML representation of the diets contained in the diet repository. We have presented a semi-automatic transformation from the syntactic model to a semantic model in order to enable its inclusion and further population in the NutElCare knowledge base for reasoning purposes. This transformation is reusable for similar XML representations of diets in other systems.
- The attachment of SWRL rules (Horrocks et al., 2004) to the knowledge base allows the inclusion of axioms which could not otherwise be expressed in OWL, namely axioms that include Horn-like rules. The rules used in the system are used for the classification of nutritional properties, calculations of values of data properties, recommendation processes and restrictions in the ingestions. The use of SWRL rules in our proposal has been validated including examples with the semantic reasoner used in the system: Pellet (Sirin et al., 2007).
- We have proposed a hybrid recommendation strategy following the guidelines provided by the experts who collaborate in this work. The formalisation of the conceptual framework strategy through UML has revealed the main processes involved which have been further analysed and implemented: the instantiation of the user profile, the diets recommendation process and the food alternatives recommendation process.
 - Diets recommendation is carried out through a knowledge-based strategy which obtains the diet that best fits user needs. It uses SWRL rules and TF-IDF techniques.
 - Food alternatives recommendation is also a hybrid strategy which combines knowledge-based and content-based paradigms. It is grounded on semantic similarity among food items, nutritional restrictions and user-learned preferences.
- We have analysed semantic similarity from the point of view of different authors in order to overcome several drawbacks found and propose a proper adaptation to the nutritional domain, in particular, to our specific application. Furthermore, we have described the evolution of the initial semantic similarity suggested in previous versions of the system until the current proposal in order to refine the results.
- We have provided meaningful examples of reasoning processes involved in the different proposed recommendation methods and validated the global recommendation

strategy proposal with a large example starting from user identification onto the system towards a final recommendation.

- In order to address the challenges to face when target users belong to the elderly sector of population, it has been included a monitoring component which allows keeping track of user previous ingestions and identifying nutritional deficits in the dietary habits of the users. Several results of monitoring are included in the recommendation processes. Changes in the nutritional state of users are also monitored through the MNA (Mini Nutritional Assessment) Screening method (Guigoz et al., 1997) in its short form MNA-SF (Rubenstein et al., 2001). The usage of alerts and notifications allow carers and the user himself to be aware of any abnormality or deviation detected. Finally, to tackle the *user non compliance* problem, the inferred food preferences of the users have been included in the recommendation processes and the user interface has been designed following usability standards for older people along with the use of Arasaac⁶⁸ pictograms.
- Finally, NutEiCare has been integrated into VIRTRA-EL (Rute-Pérez et al., 2014), a platform to provide active ageing support. This integration aims to contribute to the holistic active ageing care supplying sustenance to three main dimensions associated with health and remarkably related to each other: cognitive stimulation, physical activity and nutrition. This will allow the detection of possible deficiencies or improvements to perform and the validation by the users themselves.

In conclusion, this thesis has studied the application of the Semantic Web Technologies in the area of Recommender Systems, in particular, in the domain of nutrition. The reutilisation of semantic sources of nutritional knowledge, as the Agrovoc FAO thesaurus, has allowed the development of an OWL ontology representing the concepts and the relations among them in the nutritional domain. Domain knowledge and user profile representations by means of OWL ontologies have proven to provide the following benefits:

- Overcome heterogeneity problems of RS.
- Serve as solution of other inherent shortcomings of RS. For instance, the addition of a new item to the right position into the ontology provides semantic information about it, overcoming part of the *cold start* problem explained in chapter 1.
- Supply a formal, precise and reusable representation of a domain of discourse and the related user profiles.
- The inclusion of nutritional restrictions using OWL constructions and SWRL rules.
- The inference of new knowledge not expressed formally in the ontology through semantic reasoners and the application of automated inference techniques to learn user preferences.
- The definition of recommendation strategies which take advantage of the users' and domain semantic representation to perform semantic similarity calculations among food items.

Our proposal has been validated by implementing a RS which exploits the information contained in an ontological knowledge base on the nutritional domain to supply personalized,

⁶⁸ ARASAAC. Gobierno de Aragón, Portal Aragonés de la Comunicación Aumentativa y Alternativa. <http://www.catedu.es/arasaac/>.

healthy and appealing diet plans to elderly users. The system fills the gaps found in other literature proposals, in order to provide a solution as holistic as possible, tackling the main challenges which arise when developing a Nutritional Recommender System for the elderly.

8.2. Ongoing Work

Future work on this thesis is grounded on two main ideas: firstly, the improvement of the system by taking advantage of the current pilot programme of VIRTRA-EL, in which users are starting to use NutEiCare as a part of it, and secondly, the formal exposure of the different ontologies developed and knowledge inferred to other researchers and developers in the Semantic Web and Linked Open Data environments.

8.2.1 Intra-System Enhancements

It is important to note that one of the original goals of this research work was the recommendation of diet plans to elderly users which promote healthy eating and simultaneously tailor to their food preferences. Then, an immediate future work is the formal evaluation of the results of the users' system interaction and satisfaction with the obtained recommendations both from the health and the preferences points of view. This evaluation is already being prepared and planned to take place at some moment of the current pilot programme of the VIRTRA-EL project.

As a constant evolution of the system, more diets and food elements together with new restrictions and ontology refinements can be introduced in the system on demand.

We also plan a deeper research on automated inference, such as the use of machine learning approaches, for ontology learning in the nutritional recommendations domain. Next, it is intended an experimental phase in order to compare the results of the application of relevant machine learning techniques and determine whether they are able to improve the results provided by our current proposal.

As future work, and taking advantage from the collaboration capability of VIRTRA-EL, we plan the addition of collaborative filtering and recipe sharing among users promoting the social aspects of active aging.

Furthermore, a mobile version of the system is projected, so that ubiquity may be promoted, context information can be transparently perceived and notifications awareness could occur more quickly.

8.2.2 Knowledge Sharing

Keeping in main another important goal of this thesis, which is the reuse and sharing of expert knowledge, the ontologies developed in the system are going to be formally released as part of the Linked Open Data community project. This will allow other researchers and developers to benefit from the already formalised and validated knowledge contained in the proposed ontologies. Some research efforts found in the literature on this area applied to Recommender Systems are (Di Noia et al., 2012) and (Ostuni et al., 2013).

Semantic Web Services, as a research paradigm, is generally defined as the augmentation of Web Service descriptions through Semantic Web annotations, to facilitate the higher automation of service discovery, composition, invocation, and monitoring in the Semantic Web (Payne and Lassila, 2004). Although NutEiCare software architecture is designed on the basis of

Web Services, there is still a labour to do towards the conversion into a Semantic Service Oriented Architecture (SSOA) which further encourages the reuse of Semantic Web Services.

Moreover, additional Semantic Web Services will be developed and published, so that they can be discovered and queried by external semantic agents, processes or applications for knowledge reutilisation purposes. One example is a nutritional calculator which responds with the overall nutritional properties of one course or one menu.

Chapter References

- ADOMAVICIUS, G. & TUZHILIN, A. 2005. Toward the next generation of recommender systems: A survey of the state-of-the-art and possible extensions. *IEEE Transactions on Knowledge and Data Engineering*, 17, 734-749.
- BIZER, C., HEATH, T. & BERNERS-LEE, T. 2009. Linked data-the story so far. *Semantic Services, Interoperability and Web Applications: Emerging Concepts*, 205-227.
- CARACCILO, C., STELLATO, A., MORSHED, A., JOHANNSEN, G., RAJBHANDARI, S., JAQUES, Y. & KEIZE, J. 2013. The Agrovoc linked dataset. *Semantic Web*, 4, 341-348.
- DI NOIA, T., MIRIZZI, R., OSTUNI, V. C., ROMITO, D. & ZANKER, M. Linked open data to support content-based recommender systems. Proceedings of the 8th International Conference on Semantic Systems, 2012. ACM, 1-8.
- GUIGOZ, Y., VELLAS, B. & GARRY, P. J. 1997. Mini Nutritional Assessment: a practical assessment tool for grading the nutritional state of elderly patients. *The mini nutritional assessment: MNA. Nutrition in the elderly*.
- HECKMANN, D., SCHWARTZ, T., BRANDHERM, B., SCHMITZ, M. & VON WILAMOWITZ-MOELLENDORFF, M. 2005. Gumo—the general user model ontology. *User Modelling*. Springer Berlin Heidelberg.
- HERRERA-VIDEVA, E., PORCEL, C. & HIDALGO, L. 2004. Sistemas de recomendaciones: herramientas para el filtrado de información en Internet. *Hipertext.net*, 2.
- HORROCKS, I., PATEL-SCHNEIDER, P. F., BOLEY, H., TABET, S., GROSOFF, B. & DEAN, M. 2004. *SWRL: A semantic web rule language combining OWL and RuleML* [Online]. W3C Member submission. Available: <http://www.w3.org/Submission/SWRL/> [Accessed 30 November 2015].
- LOIZOU, A. & DASMAHAPATRA, S. Recommender systems for the semantic web. Proceedings of the ECAI 2006, 2006. 76-81.
- LOPS, P., GEMMIS, M. D. & SEMERARO, G. 2011. Content-based recommender systems: State of the art and trends. *Recommender systems handbook*. Springer US.
- OSTUNI, V. C., DI NOIA, T., DI SCIASCIO, E. & MIRIZZI, R. Top-N recommendations from implicit feedback leveraging linked open dat. Proceedings of the 7th ACM conference on Recommender systems 2013. ACM, 85-92.
- PAYNE, T. R. & LASSILA, O. 2004. Semantic web services. *IEEE Intelligent Systems*, 19, 14-15.
- RUBENSTEIN, L. Z., HARKER, J. O., SALVÀ, A., GUIGOZ, Y. & VELLAS, B. 2001. Screening for undernutrition in geriatric practice developing the short-form mini-nutritional assessment (MNA-SF). *The Journals of Gerontology Series A: Biological Sciences and Medical Sciences*, 56, M366-M372.
- RUTE-PÉREZ, S., SANTIAGO-RAMAJO, S., HURTADO, M. V., RODRÍGUEZ-FÓRTIZ, M. J. & CARACUEL, A. 2014. Challenges in software applications for the cognitive evaluation and stimulation of the elderly. *Journal of NeuroEngineering and Rehabilitation*, 11, 88.

- SIRIN, E., PARSIA, B., GRAU, B. C., KALYANPUR, A. & KATZ, Y. 2007. Pellet: A practical owl-dl reasoner. *Web Semantics: science, services and agents on the World Wide Web*, 5, 51-53.
- ZIEGLER, C. N. 2005. Semantic web recommender systems. *Current Trends in Database Technology-EDBT 2004 Workshops*. Springer Berlin Heidelberg.

ANEXO A

EJEMPLOS DE MODELOS DE DIETAS PARA MAYORES

<i>DIETA ESTÁNDAR</i>				
	<i>Desayuno</i>	<i>Comida</i>	<i>Merienda</i>	<i>Cena</i>
<i>Lunes</i>	Café descafeinado o cacao en polvo + con leche semidescremada. Pan con mantequilla y mermelada/ aceite de oliva	Lentejas estofadas con verduritas. Pescado a la marinera. Naranja.	Zumo de frutas. Magdalenas.	Sopa de pasta. Redondo de pavo con salsa ligera. Kiwi.
<i>Martes</i>	Café descafeinado o cacao en polvo + con leche semidescremada. Pan con tomate con jamón cocido /aceite de oliva.	Macarrones napolitana. Muslitos de pollo al horno con verduras. Pera.	Leche semidescremada. Pan con membrillo.	Acelgas con patatas. Croquetas de queso con ensalada variada. Melocotón en almíbar.
<i>Miércoles</i>	Café descafeinado o con cacao en polvo + con leche semidescremada. Pan con mantequilla y mermelada/aceite de oliva.	Crema de verduras. Butifarra a la plancha con patatas fritas. Macedonia de fruta natural.	Yogur. Bizcocho.	Arroz hervido con caldo. Pescado al horno con tomate. Manzana.
<i>Jueves</i>	Café descafeinado o cacao en polvo + con leche semidescremada. Pan con tomate con pavo /aceite de oliva.	Sopa de pasta. Estofado de ternera. Manzana.	Leche semidescremada. Pan con queso en porciones.	Menestra de verduras. Tortilla francesa. Mandarinas.
<i>Viernes</i>	Café descafeinado o con cacao en polvo + con leche semidescremada. Pan con tomate con queso.	Ensalada variada de hortalizas. Paella mixta. Mandarinas.	Leche semidescremada. Galletas tipo "María".	Sopa de pescado. Hamburguesa de ternera con escalibada. Plátano.
<i>Sábado</i>	Café descafeinado o con cacao en polvo + con leche semidescremada. Pan con tomate con jamón cocido/aceite de oliva.	Garbanzos con espinacas. Calamares a la romana con tomate aliñado. Kiwi.	Infusión. Requesón con azúcar o miel.	Puré de patatas. Pescado al horno. Macedonia de fruta natural.
<i>Domingo</i>	Café descafeinado o con cacao en polvo + con leche semidescremada. Bizcocho o bollo con mantequilla y mermelada/ aceite de oliva.	Ensaladilla rusa con mayonesa con un centro de ensalada variada. Lomo relleno con pasas, ciruelas y piñones. Producto lácteo (flan, natillas, yogur, requesón...).	Leche semidescremada. Bollería.	Pan con tomate y embutido o tortilla rellena de verduras. Ensalada variada (lechuga, zanahoria rallada, etc.). Pera.

Incluir en los menús el pan y las bebidas.

Figura A.1: Ejemplo de dieta estándar para personas mayores. Fuente: (Arbonés et al., 2003).

<i>DIETA BLANDA</i>				
	<i>Desayuno</i>	<i>Comida</i>	<i>Merienda</i>	<i>Cena</i>
<i>Lunes</i>	Leche semidescremada. Pan con mantequilla y mermelada/aceite de oliva.	Puré de patata con poca verdura. Pescado a la marinera (jugo suave). Fruta cocida.	Zumo de frutas (no ácidos). Galletas tipo María.	Sopa de pasta. Redondo de pavo con salsa ligera. Manzana al horno.
<i>Martes</i>	Leche semidescremada. Pan con jamón cocido/ aceite de oliva.	Macarrones con salsa blanca. Muslitos de pollo al horno con zanahoria. Pera hervida.	Leche semidescremada. Pan con membrillo.	Calabacín sin piel con patata. Rebanadas de pan con queso. Melocotón en almíbar.
<i>Miércoles</i>	Leche semidescremada. Pan con mantequilla y mermelada/aceite de oliva.	Crema de zanahoria. Butifarra a la plancha con patata al horno. Macedonia de fruta en almíbar.	Yogur. Bizcocho.	Aroz hervido con caldo. Pescado al horno. Manzana con canela.
<i>Jueves</i>	Leche semidescremada. Pan con pavo/aceite de oliva.	Sopa de pasta. Estofado de ternera magra con zanahoria. Compota.	Leche semidescremada. Pan con queso en porciones.	Patata y zanahoria hervida. Tortilla francesa. Membrillo.
<i>Viernes</i>	Leche semidescremada. Pan con queso.	Puré de calabacín sin piel y zanahoria. Arroz blanco con tofilla. Melocotón en almíbar.	Leche semidescremada. Galletas tipo "María"	Sopa de pescado suave. Hamburguesa de ternera con escalibada. Piña en almíbar.
<i>Sábado</i>	Leche semidescremada. Pan con jamón cocido/ aceite de oliva.	Puré de legumbre pasado por el colador chino con patata y zanahoria. Pescado hervido con laurel. Pera al horno.	Infusión. Requesón con azúcar o miel.	Puré de patatas. Pescado al horno. Macedonia de fruta en almíbar.
<i>Domingo</i>	Leche semidescremada. Bizcocho o bollo con mantequilla y mermelada.	Patatas hervidas con calabacín sin piel. Hamburguesa. Producto lácteo (flan, natillas, yogur, requesón...)	Leche semidescremada. Brioche.	Pan con pavo o jamón cocido o queso o tortilla francesa. Pera hervida con canela.

Incluir en los menús el pan y las bebidas.

Figura A.2: Ejemplo de dieta semilíquida para personas mayores. Fuente: (Arbonés et al., 2003).

<i>DIETA SEMILÍQUIDA</i>				
	<i>Desayuno</i>	<i>Comida</i>	<i>Merienda</i>	<i>Cena</i>
<i>Lunes</i>	Infusión. Leche semidescremada.	Puré de patata líquido. Tx Pescado. Yogur.	Zumo de frutas (no ácidos).	Sopa de pasta. Tx de pavo. Compota de manzana.
<i>Martes</i>	Infusión. Leche semidescremada.	Sopa de pasta. Tx de pollo al horno con zanahoria. Pera licuada.	Leche semidescremada.	Puré de calabacín sin piel con patata. Tx de ternera. Zumo de frutas.
<i>Miércoles</i>	Infusión. Leche semidescremada.	Crema de zanahoria. Tx de carne. Zumo de frutas.	Batido de yogur.	Sopa de arroz. Tx de pescado. Compota.
<i>Jueves</i>	Infusión. Leche semidescremada.	Sopa de pasta. Tx de carne. Compota.	Leche semidescremada.	Crema de patata y zanahoria. Tx de huevo. Infusión.
<i>Viernes</i>	Infusión. Leche semidescremada.	Sopa de arroz. Tx de huevo con puré de calabacín sin piel y zanahoria. Zumo de fruta.	Batido de yogur.	Sopa de pescado muy suave. Tx de ternera. Zumo de frutas.
<i>Sábado</i>	Infusión. Leche semidescremada.	Sopa vegetal. Tx de pescado. Compota.	Infusión o leche semidescremada.	Puré de patatas. Tx pescado al horno. Batido de yogur y Fruta cocida.
<i>Domingo</i>	Infusión. Leche semidescremada.	Puré de patatas con calabacín sin piel. Hamburguesa. Batido de yogur.	Leche semidescremada.	Sopa de pasta. Tx de jamón cocido. Compota de pera.

Los Tx de pasarán por un colador chino.

La leche semidescremada también se puede ofrecer con azúcar, excepto en aquellas personas que presenten alguna patología que no recomiende su utilización. Puede ser substituida por un edulcorante artificial.

Los zumos de fruta a ofrecer deben ser de frutas no ácidas. Las cocciones serán sencillas: planchas, hervidos, horno, etc.

La textura de los alimentos será líquida o semisólido.

Figura A.3: Ejemplo de dieta blanda para personas mayores. Fuente: (Arbonés et al., 2003).

ANEXO B

RESTRICCIONES Y RECOMENDACIONES NUTRICIONALES
PARA MAYORES

Grupos de alimentos	Frecuencia recomendada	Peso de cada ración (en crudo y neto)	Medidas caseras
Patatas, arroz, pan, pan integral y pasta	4-6 raciones al día ↑ formas integrales	60-80 g de pasta, arroz 40-60 g de pan 150-200 g de patatas	1 plato normal 3-4 rebanadas o un panecillo 1 patata grande o 2 pequeñas
Verduras y hortalizas	≥ 2 raciones al día	150-200 g	1 plato de ensalada variada 1 plato de verdura cocida 1 tomate grande, 2 zanahorias
Frutas	≥ 3 raciones al día	120-200 g	1 pieza mediana, 1 taza de cerezas, fresas..., 2 rodajas de melón...
Aceite de oliva	3-6 raciones al día	10 ml	1 cucharada sopera
Leche y derivados	2-4 raciones al día	200-250 ml de leche 200-250 g de yogur 40-60 g de queso curado 80-125 g de queso fresco	1 taza de leche 2 unidades de yogur 2-3 lonchas de queso 1 porción individual
Pescados	3-4 raciones a la semana	125-150 g	1 filete individual
Carnes magras, aves y huevos	3-4 raciones de cada a la semana. Alternar su consumo	100-125 g	1 filete pequeño, 1 cuarto de pollo o conejo, 1-2 huevos
Legumbres	2-4 raciones a la semana	60-80 g	1 plato normal individual
Frutos secos	3-7 raciones a la semana	20-30 g	1 puñado o ración individual
Embutidos y carnes grasas	Ocasional y moderado		
Dulces, snacks, refrescos	Ocasional y moderado		
Mantequilla, margarina y bollería	Ocasional y moderado		
Agua de bebida	4-8 raciones al día	200 ml aprox.	1 vaso o 1 botellín
Cerveza o vino	Consumo opcional y moderado en adultos	Vino: 100 ml Cerveza: 200 ml	1 vaso o 1 copa
Práctica de actividad física	Diario	> 30 minutos	

Figura B.1. Consumo de grupos de alimentos recomendado. Pesos, raciones y medidas caseras. Fuente: (Dapcich et al., 2004).

Primeros platos		Segundos platos
Cereales Arroz: en paellas, al horno, en sopas, en ensaladas. Pasta: con tomate, en ensaladas, con atún, con verduras al vapor.		Pescado (blanco y azul) Al horno, al papillote, a la plancha, en guisos, en fritos o rebozados, en ensaladas.
Legumbres Lentejas, judías, garbanzos, guisantes y habas: en potajes, en sopas, en guisos, en ensaladas.		Huevos Cocidos en ensaladas, en tortillas, revueltos, fritos.
Tubérculos y verduras Patatas: guisos, en ensaladillas, en tortillas, asadas. Verduras cocidas. Verduras crudas.		Carne Aves, cerdo, ternera, buey, cordero, conejo: a la plancha, al horno, asadas, hervidas, estofadas, fritas, en ensaladas.

Guarniciones	Postres	Bebidas
Ensaladas Zanahoria, lechuga, tomate, aceitunas, maíz, judías, aguacate, cebolla, espinaca, remolacha, etc.	Fruta Manzana, pera, plátano, naranja, mandarina, albaricoque, cereza, melocotón, melón, frutos secos, macedonias, etc.	Agua Recordar que debemos consumir como mínimo 1 litro y medio (de 6 a 8 vasos) de agua al día.
Patatas, cereales, legumbres		Aguas con gas, bebidas refrescantes
Otros Verduras y hortalizas cocidas: hervidas, asadas, al vapor o rebozadas.	Otros Lácteos: yogur fresco, leche fermentada, flan, queso, natilla, cuajada.	Zumos
		Bebida alcohólica de baja graduación Vino, cerveza, cava. (Consumo opcional y moderado en adultos).

Figura B.2. Platos de ingesta de alimentos considerados en la confección de menús. Fuente: (Dapcich et al., 2004).



Figura B.3. Pirámide de raciones diarias recomendadas para personas mayores. Fuente: (Dapcich et al., 2004).



Figura B.4. Rueda de consumo recomendado de alimentos para personas mayores. Fuente: (Pinilla and de Torres Aured, 2007).

<i>Minerales de especial atención en ancianos. Ingestas Recomendadas (IR)</i>				
<i>IR/día</i>	<i>Ca* 1.000 mg</i>	<i>Zn* 10 (H) y 7 (M) mg</i>	<i>Fe* 10 mg</i>	<i>Se** 55 µg</i>
<i>Fuentes</i>	Leche y derivados, pescados pequeños comidos enteros, hortalizas, leguminosas, aguas con Ca.	Carnes rojas, pescados, leche, leguminosas.	Fe hemo: vísceras, carnes, aves, y pescados. Fe no hemo: leguminosas, frutos secos y verduras.	Alimentos de origen animal y verduras y hortalizas.
<i>Deficiencia</i>	Menor absorción. Menor 1,25 dihidroxi-vitamina D. Desmineralización.	Implicado en: mantenimiento del sentido del gusto. Función inmune. Cicatrización.	Pérdidas de sangre. Menor absorción de hierro no hemo (hipo o aclorhidria en gastritis atrófica).	Enfermedad coronaria. Cáncer. Sistema inmune.

Observaciones:

* DACH⁶, ** DRI⁶**. Se ha elegido un valor u otro por estar en el mismo nivel. (H) hombres; (M) mujeres

Calcio: Existen grandes discrepancias. Se ha tomado el valor más bajo de IR por considerar que las cifras más altas entran en el rango de no absorción.

Cinc: Las necesidades rara vez son cubiertas con la dieta, sobre todo si la ingesta de energía es baja. La fibra y los fitatos de los cereales pueden limitar su absorción.

Hierro: El pH ácido es necesario para transformar la forma férrica en ferrosa, para evitar la precipitación del hierro y favorecer la captación por el enterocito.

Figura B.5. Ingestas de minerales recomendadas diariamente. Fuente: (Arbonés et al., 2003).

<i>Vitaminas hidrosolubles de especial atención en ancianos. Ingestas Recomendadas (IR)</i>				
<i>IR/día</i>	<i>Folato*</i> 400 µg	<i>Vit. B₁₂**</i> 3 µg	<i>Vit. B₆* 1,4 (H), 1,2 (M) mg</i> <i>B₉ (mg)/proteína (g) > 0,02)</i>	<i>Vit. C*</i> 100 mg
<i>Fuentes</i>	Verduras, leguminosas, hígado. Lábil a la acción del calor y aire.	Origen animal.	Carnes, pescados, huevos y cereales.	Frutas y hortalizas, (cítricos, fresas, tomates, pimientos y patatas). Lábil a la acción del oxígeno, luz y calor.
<i>Deficiencia</i>	Enfermedad cardiovascular (junto a menor ingesta de vitaminas B ₆ y B ₁₂ = aumento de homocisteína). Deterioro de la pared arterial. Demencia.	Neuropatía periférica, ataxia o alteraciones cognitivas.	Alteración del metabolismo proteico, función cognitiva e inmune.	Implicada en la síntesis de colágeno, cicatrización, funcionamiento adrenal, absorción del hierro no hemo, cataratas, algunos tipos de cáncer y otras enfermedades crónicas.

*Observaciones:** DACH⁶, ** DRI^{6,46}. (H) hombres; (M) mujeres.*Folato.* El cambio en las IR de folato, aumentándolas, es un claro ejemplo del mejor conocimiento de sus funciones.*Vitamina B₁₂.* Existen mayores necesidades de vitamina B₁₂ por la atrofia gástrica, menor secreción ácida y de factor intrínseco. Entre un 10% y un 30% de las personas mayores no absorben adecuadamente la vitamina, aunque gracias a su gran reserva hepática y vida media larga, son necesarios años para desarrollar una deficiencia.*Vitamina B₆.* Los niveles plasmáticos y séricos de vitamina B₆ en ancianos disminuyen con la edad.

Figura B.6. Ingesta de vitaminas hidrosolubles recomendada diariamente. Fuente: (Arbonés et al., 2003).

<i>Ingestas Recomendadas (IR) de vitaminas liposolubles en personas mayores</i>				
	<i>Vit. A: Eq. Retinol**</i>	<i>Vitamina D***</i>	<i>Vitamina E***</i>	<i>Vitamina K**</i>
<i>IR/día</i>	1.000 mcg (H) 800 mcg (M)	10-15 mcg (H) 10-15 mcg (M)	12-15 mg (H) 11-15 mg (M)	65 mcg (H) 80 mcg (M)
<i>Fuente + factor externo</i>	Hígado Leche Mantequilla Vegetales: carotenos	Pescados Alimentos enriquecidos + Sol	Aceites Verduras Frutos secos Cereales	Carnes, leche Cereales Verduras, frutas
<i>Deficiencia</i>	Menor resistencia a infecciones. Trastornos de visión y alteración de piel y mucosas.	Osteomalacia. Osteoporosis. Deformaciones óseas. Menor resistencia a infecciones.	Está disminuida en: Arteriosclerosis. Anemias hemolíticas. Algunos tipos de cáncer.	Hemorragias. Se ha relacionado con menor densidad ósea.

*Observaciones:** DACH⁶, ** DRI^{6,46}. (H) hombres; (M) mujeres.

Figura B.7. Ingesta de vitaminas liposolubles recomendada diariamente. Fuente: (Arbonés et al., 2003).

<i>Consumo de alimentos (g por persona y día).</i> <i>Estudio SÉNECA</i>		
	<i>Hombres</i>	<i>Mujeres</i>
Cereales y derivados.....	294 ± 189	251 ± 160
Pan blanco	243 ± 186	212 ± 160
Leche y derivados.....	448 ± 270	435 ± 280
Leche entera de vaca	316 ± 294	320 ± 292
Huevos	26 ± 25	27 ± 25
Azúcar.....	20 ± 22	15 ± 18*
Aceites y grasas	43 ± 40	50 ± 35
Aceite de oliva.....	26 ± 22	30 ± 33
Mantequilla.....	0,5 ± 2,5	0,2 ± 1,0
Verduras y hortalizas	305 ± 187	353 ± 240
Patatas.....	162 ± 166	199 ± 202
Leguminosas.....	12,7 ± 16,2	7,2 ± 9,8**
Frutas	466 ± 434	414 ± 320
Naranjas.....	172 ± 278	156 ± 191
Verduras (excepto patatas) + frutas.....	601 ± 481	573 ± 382
Came y derivados	137 ± 83	124 ± 85
Pescados	88 ± 67	100 ± 80
Bebidas no alcohólicas	53 ± 126	49 ± 109
Alcohol	30 ± 46	8,9 ± 17***

* p < 0,05; ** p < 0,01.

Figura B.8. Consumo diario recomendado de alimentos según género. Fuente: (Arbonés et al., 2003).

<i>Ingesta de energía y nutrientes (persona y día).</i> <i>Estudio SÉNECA</i>		
	<i>Hombres</i>	<i>Mujeres</i>
Energía (kcal).....	2.672 ± 799	2.334 ± 822**
Proteína (g).....	97 ± 28	91 ± 33**
Lípidos (g).....	96 ± 48	97 ± 49
AGS (g).....	29 ± 13	28 ± 15
AGM (g).....	41 ± 21	43 ± 24
AGP (g).....	17 ± 20	17 ± 17
Colesterol (mg)	320 ± 168	321 ± 175
Colesterol (mg/1.000 kcal).....	126 ± 64	145 ± 71
Hidratos de carbono (g)	321 ± 131	275 ± 114**
Fibra (g)	23 ± 11	21 ± 9
Calcio (mg)	1.051 ± 449	1.005 ± 485
Hierro (mg)	16,3 ± 5,9	13,8 ± 4,7**
Magnesio (mg).....	252 ± 93	237 ± 89
Cinc (mg)	13,4 ± 4,8	11,9 ± 4,6*
Sodio (g).....	2,1 ± 1,0	1,9 ± 1,0
Potasio (g).....	3,6 ± 1,3	3,5 ± 1,3
Tiamina (mg)	1,4 ± 0,5	1,3 ± 0,5
Riboflavina (mg).....	1,8 ± 0,6	1,8 ± 0,8
Eq. niacina (mg).....	32 ± 9,3	31 ± 11
Vitamina B ₆ (mg).....	1,5 ± 0,6	1,5 ± 0,6
Folato (µg)	232 ± 126	240 ± 122
Vitamina B ₁₂ (µg).....	7,6 ± 6,0	8,1 ± 14
Vitamina C (mg).....	171 ± 132	179 ± 106
Vitamina D (µg).....	2,4 ± 2,5	2,1 ± 2,9
Vitamina E (mg)	12 ± 19	13 ± 16
Eq. retinol (µg).....	1.193 ± 797	1.404 ± 2.633

* p < 0,05; ** p < 0,01; *** p < 0,001.

Figura B.9. Consumo diario recomendado de energía y nutrientes según género. Fuente: (Arbonés et al., 2003).

ANEXO D

CUESTIONARIOS DE SCREENING

Mini Nutritional Assessment

MNA[®]Nestlé
Nutrition Institute

Apellidos:					Nombre:		
Sexo:		Edad:		Peso, kg:		Talla, cm:	
Fecha:							

Responda al cuestionario eligiendo la opción adecuada para cada pregunta. Sume los puntos para el resultado final.

Cribaje	
A Ha comido menos por falta de apetito, problemas digestivos, dificultades de masticación o deglución en los últimos 3 meses? 0 = ha comido mucho menos 1 = ha comido menos 2 = ha comido igual	<input type="checkbox"/>
B Pérdida reciente de peso (<3 meses) 0 = pérdida de peso > 3 kg 1 = no lo sabe 2 = pérdida de peso entre 1 y 3 kg 3 = no ha habido pérdida de peso	<input type="checkbox"/>
C Movilidad 0 = de la cama al sillón 1 = autonomía en el interior 2 = sale del domicilio	<input type="checkbox"/>
D Ha tenido una enfermedad aguda o situación de estrés psicológico en los últimos 3 meses? 0 = sí 2 = no	<input type="checkbox"/>
E Problemas neuropsicológicos 0 = demencia o depresión grave 1 = demencia moderada 2 = sin problemas psicológicos	<input type="checkbox"/>
F1 Índice de masa corporal (IMC) = peso en kg / (talla en m)² <input type="checkbox"/> 0 = IMC <19 1 = 19 ≤ IMC < 21 2 = 21 ≤ IMC < 23 3 = IMC ≥ 23	<input type="checkbox"/>
SI EL ÍNDICE DE MASA CORPORAL NO ESTÁ DISPONIBLE, POR FAVOR SUSTITUYA LA PREGUNTA F1 CON LA F2. NO CONTESTE LA PREGUNTA F2 SI HA PODIDO CONTESTAR A LA F1.	
F2 Circunferencia de la pantorrilla (CP en cm) 0 = CP <31 3 = CP ≥ 31	<input type="checkbox"/>
Evaluación del cribaje (max. 14 puntos)	<input type="checkbox"/> <input type="checkbox"/>
12-14 puntos: <input type="checkbox"/> estado nutricional normal	<input type="button" value="Guardar"/>
8-11 puntos: <input type="checkbox"/> riesgo de malnutrición	<input type="button" value="Imprimir"/>
0-7 puntos: <input type="checkbox"/> malnutrición	<input type="button" value="Reset"/>

Figura D.1. MNA-SF en español. Fuente: (Vellas et al., 2006) y (Rubenstein et al., 2001).

LISTADO DE ALIMENTOS	¿CUÁNTAS VECES COME?	
	A LA SEMANA	AL MES
Leche		
Yogur		
Chocolate: tableta, bombones, "Kit Kat", "Mars"...		
Cereales inflados de desayuno ("Corn-Flakes", "Kellog's")		
Galletas tipo "maría"		
Galletas con chocolate, crema...		
Magdalenas, bizcocho...		
Ensamada, donut, croissant...		
	A LA SEMANA	AL MES
Ensalada: lechuga, tomate, escarola...		
Judías verdes, acelgas o espinacas		
Verduras de guarnición: berenjena, champiñones		
Patatas al horno, fritas o hervidas		
Legumbres: lentejas, garbanzos, judías...		
Arroz blanco, paella		
Pasta: fideos, macarrones, espaguetis...		
Sopas y cremas		
	A LA SEMANA	AL MES
Huevos		
Pollo o pavo		
Tertera, cerdo, cordero (bistec, empanada, ...)		
Carne picada, longaliza, hamburguesa		
Pescado blanco: merluza, mero, ...		
Pescado azul: sardinas, atún, salmón, ...		
Marisco: mejillones, gambas, langostinos, calamares, ...		
Croquetas, empanadillas, pizza		
Pan (en bocadillo, con las comidas, ...)		
	A LA SEMANA	AL MES
Jamón salado, dulce, embutidos		
Queso blanco o fresco (Burgos, ...) o bajo en calorías		
Otros quesos: curados o semicurado, cremosos		
	A LA SEMANA	AL MES
Frutas cítricas: naranja, mandarina, ...		
Otras frutas: manzana, pera, melocotón, plátano...		
Frutas en conserva (en almíbar...)		
Zumos de fruta natural		
Zumos de fruta comercial		
Frutos secos: cacahuets, avellanas, almendras, ...		
Postres lácteos: natillas, flan, requesón		
Pasteles de crema o chocolate		
Bolsas de aperitivos («chips», «chetos», «fritos»...)		
Golosinas: gominolas, caramelos, ...		
Helados		
	A LA SEMANA	AL MES
Bebidas azucaradas ("coca-cola", "Fanta"...)		
Bebidas bajas en calorías (coca-cola light...)		
Vino, sangría		
Cerveza		
Cerveza sin alcohol		
Bebidas destiladas: whisky, ginebra, coñac, ...		

Figura D.2. Ejemplo de Cuestionario de Frecuencia de Consumo Alimenticio CFCA. Fuente: (Trinidad Rodríguez et al., 2008).

Referencias Bibliográficas de los Anexos

- ARBONÉS, G., CARBAJAL, A., GONZALVO, B., GONZÁLEZ-GROSS, M., JOYANES, M., MARQUES-LOPES, I. & PUIGDUETA, I. 2003. Nutrición y recomendaciones dietéticas para personas mayores: Grupo de trabajo "Salud pública" de la Sociedad Española de Nutrición (SEN). *Nutrición hospitalaria*, 18, 109-137.
- DAPCICH, V., CASTELL, G. S., BARBA, L. R., RODRIGO, C. P., BARTRINA, J. A. & MAJEM, L. S. 2004. Necesidades Nutricionales en el Envejecimiento. *Guía de la Alimentación Saludable. Sociedad Española de Nutrición Comunitaria, Madrid, Everest*, 96-98.
- PINILLA, M. F. & DE TORRES AURED, M. L. 2007. La Dieta equilibrada: guía para enfermeras de Atención Primaria. *Sociedad Española de Dietética y Ciencias de la Alimentación*.
- RUBENSTEIN, L. Z., HARKER, J. O., SALVÀ, A., GUIGOZ, Y. & VELLAS, B. 2001. Screening for undernutrition in geriatric practice developing the short-form mini-nutritional assessment (MNA-SF). *The Journals of Gerontology Series A: Biological Sciences and Medical Sciences*, 56, M366-M372.
- TRINIDAD RODRÍGUEZ, I., FERNÁNDEZ BALLART, J., CUCÓ PASTOR, G., BIARNÉS JORDÀ, E. & ARIJA VAL, V. 2008. Validación de un cuestionario de frecuencia de consumo alimentario corto: reproducibilidad y validez. *Nutrición Hospitalaria*, 23, 242-252.
- VELLAS, B., VILLARS, H., ABELLAN, G. & SOTO, M. E. 2006. Overview of the MNA®- Its history and challenges / discussion. *The journal of nutrition, health & aging*, 10, 456.

Bibliografía / References

- ABERG, J. 2006. Dealing with Malnutrition: A Meal Planning System for Elderly. *AAAI Spring Symposium: Argumentation for Consumers of Healthcare*.
- ADOMAVICIUS, G. & TUZHILIN, A. 2005. Toward the next generation of recommender systems: A survey of the state-of-the-art and possible extensions. *IEEE Transactions on Knowledge and Data Engineering*, 17, 734-749.
- AL-NAZER, A., HELMY, T. & AL-MULHEM, M. 2014. User's Profile Ontology-based Semantic Framework for Personalized Food and Nutrition Recommendation. *Procedia Computer Science*, 32, 101-108.
- ALBERTONI, R. & DE MARTINO, M. 2008. Asymmetric and context-dependent semantic similarity among ontology instances. In: HEIDELBERG, S. B. (ed.) *Journal on data semantics X*.
- ANDREJKO, A., BARLA, M. & BIELIKOVÁ, M. 2007. Ontology-based user modeling for web-based information systems. In *Advances in Information Systems Development* Springer US.
- ANTONIOU, G. & HARMELEN, F. V. 2003. Web Ontology Language: OWL. *Handbook on Ontologies in Information Systems*. Springer-Verlag.
- ARBONÉS, G., CARBAJAL, A., GONZALVO, B., GONZÁLEZ-GROSS, M., JOYANES, M., MARQUES-LOPES, I. & PUIGDUETA, I. 2003. Nutrición y recomendaciones dietéticas para personas mayores: Grupo de trabajo "Salud pública" de la Sociedad Española de Nutrición (SEN). *Nutrición hospitalaria*, 18, 109-137.
- ARWAN, A., PRIYAMBADHA, B., SARNO, R., SIDIQ, M. & KRISTIANO, H. 2013. Ontology and semantic matching for diabetic food recommendations. *Information Technology and Electrical Engineering (ICITEE)*. IEEE.
- BAADER, F., HORROCKS, I. & SATTLER, U. 2003. Description Logics as ontology languages for the Semantic Web. In *Festschrift in honor of Jörg Siekmann, Lecture Notes in Artificial Intelligence*. Springer-Verlag.
- BAADER, F., HORROCKS, I. & SATTLER, U. 2005. Description logics as ontology languages for the semantic web. *Mechanizing Mathematical Reasoning*. Springer Berlin Heidelberg.
- BALABANOVIC, M. & SHOHAM, Y. 1997. Fab: content-based, collaborative recommendation. *Communications in the ACM*, 40, 66-72.
- BARNARD, Y., BRADLEY, M. D., HODGSON, F. & LLOYD, A. D. 2013. Learning to use new technologies by older adults: Perceived difficulties, experimentation behaviour and usability. *Computers in Human Behavior*, 29, 1715-1724.
- BASIT, K. A. & MATSKIN, M. GUMO inspired ontology to support user experience based Citywide Mobile Learning. International Conference on User Science and Engineering (i-USEr) 2011 2011. IEEE, 195-200.
- BASU, C., HIRSH, H. & COHEN, W. Recommendation as Classification: Using Social and Content-Based Information in Recommendation. Fifteenth National Conference on Artificial Intelligence, 1998. AAAI Press, 714-720.
- BECKER, M. H. & MAIMAN, L. A. 1980. Strategies for enhancing patient compliance. *Journal of community health*, 6, 113-135.
- BECKER, S. A. 2004. A study of Web usability for older adults seeking online health resources. *ACM Transactions on Computer-Human Interaction*, 11, 387-406.
- BENGHAZI, K., HURTADO, M. V., HORNOS, M. J., RODRÍGUEZ, M. L., RODRÍGUEZ-DOMÍNGUEZ, C., PELEGRINA, A. B. & RODRÍGUEZ-FÓRTIZ, M. J. 2012. Enabling correct design and formal analysis of Ambient Assisted Living systems. *Journal of Systems and Software*, 85, 498-510.
- BERGLUND, A., BOAG, S., CHAMBERLIN, D., FERNÁNDEZ, M. F., KAY, M., ROBIE, J. & SIMÉON, J. 2003. Xml path language (xpath). *World Wide Web Consortium (W3C)*.
- BERNERS-LEE, T., HENDLER, J. & LASSILA, O. 2001. The semantic web. *Scientific american*, 284, 28-37.

- BIZER, C., HEATH, T. & BERNERS-LEE, T. 2009. Linked data-the story so far. *Semantic Services, Interoperability and Web Applications: Emerging Concepts*, 205-227.
- BLANCO-FERNÁNDEZ, Y., PAZOS-ARIAS, J. J., GIL-SOLLA, A., RAMOS-CABRER, M., LÓPEZ-NORES, M., GARCÍA-DUQUE, J. & BERMEJO-MUÑOZ, J. 2008a. A flexible semantic inference methodology to reason about user preferences in knowledge-based recommender systems. *Knowledge-Based Systems*, 21, 305-320.
- BLANCO-FERNÁNDEZ, Y., PAZOS-ARIAS, J. J., GIL-SOLLA, A., RAMOS-CABRER, M., LÓPEZ-NORES, M., GARCÍA-DUQUE, J., FERNÁNDEZ-VILAS, A. & DÍAZ-REDONDO, R. P. 2008b. Exploiting synergies between semantic reasoning and personalization strategies in intelligent recommender systems: A case study. *Journal of Systems and Software*, 81, 2371-2385.
- BLANCHARD, E., HARZALLAH, M., BRIAND, H. & KUNTZ, P. 2005. A Typology Of Ontology-Based Semantic Measures. *Proceedings of the Open Interop. Workshop on Enterprise Modelling and Ontologies for Interoperability*.
- BOHRING, H. & AUER, S. 2005. Mapping XML to OWL Ontologies. *Leipziger Informatik-Tage*, 72, 147-156.
- BOLEY, H., TABET, S. & WAGNER, G. 2001. Design Rationale for RuleML: A Markup Language for Semantic Web Rules. *SWWS (Vol. 1, pp. 381-401)*, 1, 381-401.
- BORST, W. N. 1997. *Construction of engineering ontologies for knowledge sharing and reuse*. Universiteit Twente.
- BOUDINY, K. 2013. 'Active ageing': From empty rhetoric to effective policy tool. *Ageing and society*, 33, 1077-1098.
- BOUILLANNE, O., MORINEAU, G., DUPONT, C., COULOMBEL, I., VINCENT, J. P., NICOLIS, I. & AUSSEL, C. 2005. Geriatric Nutritional Risk Index: a new index for evaluating at-risk elderly medical patients. *The American journal of clinical nutrition*, 82, 777-783.
- BRIDGE, D., GÖKER, M. H., MCGINTY, L. & SMYTH, B. 2005. Case-based recommender systems. *The Knowledge Engineering Review*, 20, 315-320.
- BRUG, J., STEENHUIS, I., ASSEMA, P. V. & VRIES, H. D. 1996. The Impact of a Computer-Tailored Nutrition Intervention. *Preventive Medicine*, 25, 236-242.
- BRUSILOVSKY, P. & MILLÁN, E. 2007. User models for adaptive hypermedia and adaptive educational systems. *The adaptive web*. Springer-Verlag.
- BÜHMANN, L. & LEHMANN, J. 2012. Universal OWL axiom enrichment for large knowledge bases. *Knowledge Engineering and Knowledge Management*. Springer Berlin Heidelberg.
- BURKE, R. 2000. Knowledge-based Recommender Systems. In: KENT, A. (ed.) *Encyclopedia of Library and Information Systems*. New York: Marcel Dekker.
- BURKE, R. 2002. Hybrid Recommender Systems: Survey and Experiments. *User Modeling and User-Adapted Interaction*, 12, 331-370.
- BURKE, R. 2007. Hybrid web recommender system. *The adaptive web*. Springer Berlin Heidelberg.
- CANTADOR, I., CASTELLS, P. & BELLOGÍN, A. 2011. An enhanced semantic layer for hybrid recommender systems: Application to news recommendation. *International Journal on Semantic Web and Information Systems*, 7, 44-78.
- CARACCILO, C., STELLATO, A., MORSHED, A., JOHANNSEN, G., RAJBHANDARI, S., JAQUES, Y. & KEIZE, J. 2013. The Agrovoc linked dataset. *Semantic Web*, 4, 341-348.
- CELMA, Ò. & SERRA, X. 2008. FOAFing the music: Bridging the semantic gap in music recommendation. *Web Semantics: Science, Services and Agents on the World Wide Web*, 6, 250-256.
- COUTO, F. M., SILVA, M. J. & COUTINHO, P. M. Semantic similarity over the gene ontology: family correlation and selecting disjunctive ancestors. *Proceedings of the*

- 14th ACM international conference on Information and knowledge management 2005. ACM, 343-344.
- CHANDRASEKARAN, B., JOSEPHSON, J. R. & BENJAMINS, V. R. 1999. What are ontologies, and why do we need them? *IEEE Intelligent Systems*, 1, 20-26.
- CHESNEVAR, C. I. & MAGUITMAN, A. G. ArgueNet: An Argument-Based Recommender System for Solving Web Search Queries. 2nd IEEE Intelligent Systems Conference, IS-2004, 2004. 282-287.
- DAPCICH, V., CASTELL, G. S., BARBA, L. R., RODRIGO, C. P., BARTRINA, J. A. & MAJEM, L. S. 2004. Necesidades Nutricionales en el Envejecimiento. *Guía de la Alimentación Saludable. Sociedad Española de Nutrición Comunitaria, Madrid, Everest*, 96-98.
- DE PESSEMIER, T., DOOMS, S. & MARTENS, L. A food recommender for patients in a care facility. Proceedings of the 7th ACM conference on Recommender systems, 2013. ACM, 209-212.
- DEGEMMIS, M., LOPS, P. & SEMERARO, G. 2007. A content-collaborative recommender that exploits WordNet-based user profiles for neighborhood formation. *User Modeling and User-Adapted Interaction*, 17, 217-255.
- DI NOIA, T., MIRIZZI, R., OSTUNI, V. C., ROMITO, D. & ZANKER, M. Linked open data to support content-based recommender systems. Proceedings of the 8th International Conference on Semantic Systems, 2012. ACM, 1-8.
- DUAN, S., FOKOUE, A., HASSANZADEH, O., KEMENTSIETSIDIS, A., SRINIVAS, K. & WARD, M. J. 2012. Instance-based matching of large ontologies using locality-sensitive hashing. *The Semantic Web-ISWC 2012*. Springer Berlin Heidelberg.
- EHRIG, M., HAASE, P., HEFKE, M. & STOJANOVIC, N. Similarity for ontologies-a comprehensive framework. ECIS 2005, 2005 Regensburg, Germany
- EL-DOSUKY, M. A., RASHAD, M. Z., HAMZA, T. T. & EL-BASSIOUNY, A. H. 2012. Food Recommendation Using Ontology and Heuristics. *Communications in Computer and Information Science*, 322, 423-429.
- ESPÍN, V., HURTADO, M. V. & NOGUERA, M. 2016. Nutrition for Elder Care: a nutritional semantic recommender system for the elderly. *Expert Systems*, 33, 201-210.
- ESPÍN, V., HURTADO, M. V., NOGUERA, M. & BENGHAZI, K. Semantic-Based Recommendation of Nutrition Diets for the Elderly from Agroalimentary Thesauri. Flexible Query Answering Systems, 2013. Springer Berlin Heidelberg, 471-482.
- ESPÍN, V., NOGUERA, M. & HURTADO, M. V. 2015. A Mixed Approach for the Representation of Nutritional Information Through XML-to-OWL Mappings. *Languages, Applications and Technologies*. Springer International Publishing.
- EUZENAT, J. & SHVAIKO, P. 2007. *Ontology matching*, Heidelberg Springer.
- EVANS, W. J. & CYR-CAMPBELL, D. 1997. Nutrition, exercise, and healthy aging. *Journal of the American Dietetic Association*, 97, 632-638.
- FARSANI, H. K. & NEMATBAKHS, M. 2006. A semantic recommendation procedure for electronic product catalog. *International Journal of Applied Mathematics and Computer Sciences*, 3, 86-91.
- FELFERNIG, A., FRIEDRICH, G., JANNACH, D. & ZANKER, M. 2006. An integrated environment for the development of knowledge-based recommender applications. *International Journal of Electronic Commerce*, 22, 11-34.
- FERDINAND, M., ZIRPINS, C. & TRASTOUR, D. Lifting XML schema to OWL. Web Engineering, 2004. Springer Berlin Heidelberg, 354-358.
- FREYNE, J. & BERKOVSKY, S. Intelligent food planning: personalized recipe recommendation. Proceedings of the 15th international conference on Intelligent user interfaces, 2010. ACM, 321-324.
- FREYNE, J., BERKOVSKY, S. & SMITH, G. 2011. Recipe recommendation: Accuracy and reasoning. *User Modeling, Adaption and Personalization*. Springer Berlin Heidelberg.

- FUDHOLI, D. H., MANEERAT, N. & VARAKULSIRIPUNTH, R. Ontology-based daily menu assistance system. 6th International Conference on Electrical Engineering/Electronics, Computer, Telecommunications and Information Technology, ECTI-CON 2009, 2009. IEEE, 694-697.
- GANESAN, P., GARCIA-MOLINA, H. & WIDOM, J. 2003. Exploiting hierarchical domain structure to compute similarity. *ACM Transactions on Information Systems (TOIS)*, 21, 64-93.
- GARSHOL, L. M. 2004. Metadata? Thesauri? Taxonomies? Topic maps! Making sense of it all. *Journal of information science*, 30, 378-391.
- GAUCH, S., SPERETTA, M., CHANDRAMOULI, A. & MICARELLI, A. 2007. User profiles for personalized information access. *The adaptive web*. Springer Berlin Heidelberg.
- GE, M., DELGADO-BATTENFELD, C. & JANNACH, D. Beyond accuracy: evaluating recommender systems by coverage and serendipity. Proceedings of the fourth ACM conference on Recommender systems 2010. ACM, 257-160.
- GHAWI, R. & CULLOT, N. 2009. Building Ontologies from XML Data Sources. *DEXA Workshops*, 480-484.
- GILCHRIST, A. 2003. Thesauri, taxonomies and ontologies-an etymological note. *Journal of documentation*, 59, 7-18.
- GOLBECK, J. Semantic web interaction through trust network recommender systems. Proceedings of the ISWC 2005 Workshop on End User Semantic Web Interaction, 2005 Galway, Ireland.
- GÓMEZ-PÉREZ, A., FERNÁNDEZ-LÓPEZ, M. & CORCHO, O. 2007. *Ontological Engineering: with examples from the areas of Knowledge Management, e-Commerce and the Semantic Web*, (Advanced Information and Knowledge Processing). Secaucus, NJ, USA, Springer-Verlag New York.
- GRAU, B. C., HORROCKS, I., MOTIK, B., PARSIA, B., PATEL-SCHNEIDER, P. & SATTLER, U. 2008. OWL 2: The next step for OWL. *Web Semantics: Science, Services and Agents on the World Wide Web*, 6, 309-322.
- GROSOFF, B. N., HORROCKS, I., VOLZ, R. & DECKER, S. Description logic programs: Combining logic programs with description logic. Proceedings of the 12th international conference on World Wide Web, 2003. ACM, 48-57.
- GRUBER, T. R. 1995. Toward principles for the design of ontologies used for knowledge sharing? *International journal of human-computer studies*, 43, 907-928.
- GRUBER, T. R. 1993. A translation approach to portable ontology specifications. *Knowledge Acquisition*, 5, 199-220.
- GRÜNINGER, M., ATEFI, K. & FOX, M. S. 2000. Ontologies to support process integration in enterprise engineering. *Computational & Mathematical Organization Theory*, 6, 381-394.
- GRUNINGER, M. & FOX, M. S. Methodology for the design and evaluation of ontologies. Proceedings of the Workshop on Basic Ontological Issues in Knowledge Sharing IJCAI, 1995. AAAI Press, 1-10.
- GUARINO, N. 1998. *Formal Ontology in Information Systems: Proceedings of the 1st International Conference June 6-8, 1998, Trento, Italy*, IOS Press.
- GUIGOZ, Y., VELLAS, B. & GARRY, P. J. 1997. Mini Nutritional Assessment: a practical assessment tool for grading the nutritional state of elderly patients. *The mini nutritional assessment: MNA. Nutrition in the elderly*.
- HACHEROUF, M., BAHLOUL, S. N. & CRUZ, C. 2015. Transforming XML documents to OWL ontologies: A survey. *Journal of Information Science*, 41, 242-259.
- HARVEY, M., LUDWIG, B. & ELSWEILER, D. You are what you eat: Learning user tastes for rating prediction. String Processing and Information Retrieval 2013. Springer International Publishing, 153-164.

- HAWKINS, R. P., KREUTER, M., RESNICOW, K., FISHBEIN, M. & DIJKSTRA, A. 2008. Understanding tailoring in communicating about health. *Health education research*, 23, 454-466.
- HEART, T. & KALDERON, E. 2013. Older adults: are they ready to adopt health-related ICT? *International journal of medical informatics*, 82, e209-e231.
- HECKMANN, D. & KRUEGER, A. 2003. A user modeling markup language (UserML) for ubiquitous computing. *User Modeling*. Springer Berlin Heidelberg.
- HECKMANN, D., SCHWARTZ, T., BRANDHERM, B., SCHMITZ, M. & VON WILAMOWITZ-MOELLENDORFF, M. 2005. Gumo—the general user model ontology. *User Modelling*. Springer Berlin Heidelberg.
- HERRERA-VIEDMA, E., PORCEL, C. & HIDALGO, L. 2004. Sistemas de recomendaciones: herramientas para el filtrado de información en Internet. *Hipertext.net*, 2.
- HOEKSTRA, R. Ontology representation design patterns and ontologies that make sense. Proceedings of the 2009 conference on Ontology Representation, 2009 Amsterdam, The Netherlands. IOS Press, 1–236.
- HORRIDGE, M. & BECHHOFFER, S. 2011. The OWL API: A Java API for OWL ontologies. *Semantic Web*, 2, 11-21.
- HORROCKS, I., KUTZ, O. & SATTLER, U. The Even More Irresistible SROIQ. Proceedings of the 10th Int. Conf. on Principles of Knowledge Representation and Reasoning (KR2006), 2006. AAAI Press, 57-67.
- HORROCKS, I., PATEL-SCHNEIDER, P. & HARMELEN, F. V. 2003. From SHIQ and RDF to OWL: The making of a web ontology language. *Journal of Web Semantics*, 1, 7-26.
- HORROCKS, I., PATEL-SCHNEIDER, P. F., BOLEY, H., TABET, S., GROSOFF, B. & DEAN, M. 2004. *SWRL: A semantic web rule language combining OWL and RuleML* [Online]. W3C Member submission. Available: <http://www.w3.org/Submission/SWRL/> [Accessed 30 November 2015].
- HUNTER, J. & MCLAUGHLIN, B. 2000. *The jdom project* [Online]. Available: <http://www.jdom.org> [Accessed 3th March 2015].
- IGLESIAS, R., IBARGUREN, I., DE SEGURA, N. G., UGALDE, J., COELLO, L. & ITURBURU, M. FoodManager: a cooking, eating and appliance controlling support system for the elderly. Proceedings of the 3rd International Conference on Pervasive Technologies Related to Assistive Environments, 2010. ACM.
- IJSSELSTEIJN, W., NAP, H. H., KORT, Y. & POELS, K. 2007. Digital game design for elderly users. *Proceedings of the 2007 conference on Future Play*. ACM.
- JOHNSON, R. & KENT, S. 2007. Designing universal access: web-applications for the elderly and disabled. *Cognition, Technology & Work*, 9, 209-218.
- KELLER, H. H., MCCULLOUGH, J., DAVIDSON, B., VESNAVER, E., LAPORTE, M., GRAMLICH, L. & JEEJEEBHOY, K. 2015. The Integrated Nutrition Pathway for Acute Care (INPAC): Building consensus with a modified Delphi. *Nutrition journal*, 14, 1.
- KONDRUP, J. E. S. P. E. N., ALLISON, S. P., ELIA, M., VELLAS, B. & PLAUTH, M. 2003. ESPEN guidelines for nutrition screening 2002. *Clinical nutrition*, 22, 415-421.
- KURNIAWAN, S. & ZAPHIRIS, P. 2005. Research-derived web design guidelines for older people. *Proceedings of the 7th international ACM SIGACCESS conference on Computers and accessibility*. ACM.
- LASSILA, O. & MCGUINNESS, D. 2001. The role of frame-based representation on the semantic web. Technical Report KSL-01-02. Knowledge Systems Laboratory, Stanford University, Stanford, CA.
- LAWRENCE, R. D., ALMASI, G. S., KOTLYAR, V., VIVEROS, M. & DURİ, S. S. 2001. Personalization of supermarket product recommendations. *Data Mining and Knowledge Discovery*, 5, 11-32.

- LEE, W. N., SHAH, N., SUNDLASS, K. & MUSEN, M. Comparison of ontology-based semantic-similarity measures. *AMIA annual symposium proceedings*, 2008. American Medical Informatics Association, 384.
- LINDEN, G., SMITH, B. & YORK, J. 2003. Amazon.com recommendations: item-to-item collaborative filtering. *Internet Computing, IEEE*, 7, 76-80.
- LOIZOU, A. & DASMAHAPATRA, S. Recommender systems for the semantic web. *Proceedings of the ECAI 2006*, 2006. 76-81.
- LOPS, P., GEMMIS, M. D. & SEMERARO, G. 2011. Content-based recommender systems: State of the art and trends. *Recommender systems handbook*. Springer US.
- LORD, P. W., STEVENS, R. D., BRASS, A. & GOBLE, C. A. 2003. Investigating semantic similarity measures across the Gene Ontology: the relationship between sequence and annotation. *Bioinformatics*, 19, 1275-1283.
- LORENZI, F. & RICCI, F. 2005. Case-based Recommender Systems: a unifying view. *Intelligent Techniques for Web Personalization*. Springer Berlin Heidelberg., 89-113.
- MAGNINI, B. & STRAPPARAVA, C. 2001. Improving user modelling with content-based techniques. *User Modeling*. Springer Berlin Heidelberg.
- MAK, H., KOPRINSKA, I. & POON, J. INTIMATE: A web-based movie recommender using text categorization. *IEEE/WIC International Conference on Web Intelligence*, 2003. 602-605.
- MASSA, P. & AVESANI, P. Trust-aware Collaborative Filtering for Recommender Systems. *Federated Int. Conference On The Move to Meaningful Internet: CoopIS, DOA, ODBASE*, 2004. 492-508.
- MCBRIDE, B. I. S. Jena: Implementing the RDF Model and Syntax Specification. *Proceedings of 2nd Int. Workshop on the Semantic Web*, 2001.
- MCCARTHY, J. F. Pocket Restaurant Finder: A situated recommender systems for groups. *Workshop on Mobile Ad-Hoc Communication at the 2002 ACM Conference*, 2002. Human Factors in Computer Systems.
- MCGUINNESS, D. L. 2003. Ontologies come of age. *Spinning the Semantic Web: Bringing the World Wide Web to Its Full Potential*. MIT Press.
- MCGUINNESS, D. L. & HARMELEN, F. V. 2004. OWL web ontology language overview. *W3C recommendation, W3C*.
- MCNEE, S. M., LAM, S. K., KONSTAN, J. A. & RIEDL, J. Interfaces for eliciting new user preferences in recommender systems. *User Modeling*, 2003. Springer Berlin Heidelberg, 178-187.
- MCSHERRY, D. 2005. Explanation in Recommender Systems. *Artificial Intelligence Review*, 24, 179-197.
- MELVILLE, P. & SINDHWANI, V. 2010. Recommender systems. *Encyclopedia of machine learning*. Springer US.
- MIDDLETON, S. E., SHADBOLT, N. R. & DE ROURE, D. C. 2004. Ontological user profiling in recommender systems. *ACM Transactions on Information Systems (TOIS)*, 22, 54-88.
- MIKA, S. 2011. Challenges for Nutrition Recommender Systems. *Proceedings of the 2nd Workshop on Context Aware Intel. Assistance*. Berlin, Germany.
- MILES, A., MATTHEWS, B., WILSON, M. & BRICKLEY, D. SKOS Core: Simple knowledge organisation for the web. *International Conference on Dublin Core and Metadata Applications*, 2005.
- MILLER, B. N., ALBERT, I., LAM, S. K., KONSTAN, J. A. & RIEDL, J. Movielen unplugged: experiences with an occasionally connected recommender system. *8th international conference on Intelligent user interfaces, IUI '03*, 2003 New York, USA. ACM, 263-266.
- MLADENIC, D. 1996. Personal WebWatcher: design and implementation. Department of Intelligent Systems, J.Stefan Institute, Slovenia.

- MOBASHER, B., COOLEY, R. & SRIVASTAVA, J. 2000. Automatic personalization based on Web usage mining. *Communications in ACM*, 43, 142-151.
- MOONEY, R. J. & ROY, L. Content-based book recommending using learning for text categorization. Fifth ACM conference on Digital libraries, DL '00, 2000 New York, USA. ACM, 195-204.
- MOUKAS, A., MOUKAS, R. & MAES, P. 1998. Amalthea: An Evolving Multi-Agent Information Filtering and Discovery System for the WWW. *Autonomous Agents and Multi-Agent Systems*, 59-88.
- NIEDRITIS, A., NIEDRITE, L. & KOZMINA, N. Integration Architecture of User Models. Perspectives in Business Informatics Research Local Proceedings of 10th International Conference BIR2011 Associated Workshops and Doctoral Consortium, 2011 Riga Technical University. 323-330.
- NILES, I. & PEASE, A. Towards a standard upper ontology. Proceedings of the international conference on Formal Ontology in Information Systems, 2001. ACM, 2-9.
- NOY, N. F. & MCGUINNESS, D. L. 2001. Ontology development 101 a guide to creating your first ontology. Technical Report. Stanford Knowledge Systems Laboratory: Stanford University, Stanford, AC.
- O'DONOVAN, J. & SMYTH, B. Trust in Recommender Systems. 10th international conference on Intelligent user interfaces, IUI '0, 2005 New York, USA. ACM, 167-174.
- OSTUNI, V. C., DI NOIA, T., DI SCIASCIO, E. & MIRIZZI, R. Top-N recommendations from implicit feedback leveraging linked open dat. Proceedings of the 7th ACM conference on Recommender systems 2013. ACM, 85-92.
- OWL WORKING GROUP. 2012. *OWL (Web Ontology Language)* [Online]. W3C. Available: <http://www.w3.org/2001/sw/wiki/OWL> [Accessed 26th November 2015].
- PAYNE, T. R. & LASSILA, O. 2004. Semantic web services. *IEEE Intelligent Systems*, 19, 14-15.
- PAZZANI, M. & BILLSUS, D. 2007. Content-based recommendation systems. *The Adaptive Web*. Lecture Notes in Computer Science, Springer Berlin, Heidelberg.
- PEIS, E., DEL CASTILLO, J. M., & DELGADO-LÓPEZ, J. A. 2008. Semantic recommender systems. Analysis of the state of the topic. *Hipertext.net*, 6, 1-5.
- PESQUITA, C., FARIA, D., FALCAO, A. O., LORD, P. & COUTO, F. M. 2009. Semantic similarity in biomedical ontologies. *PLoS Computational Biology*, 5.
- PINILLA, M. F. & DE TORRES AURED, M. L. 2007. La Dieta equilibrada: guía para enfermeras de Atención Primaria. *Sociedad Española de Dietética y Ciencias de la Alimentación*.
- PIRLICH, M. & LOCHS, H. 2001. Nutrition in the elderly. *Best Practice & Research Clinical Gastroenterology*, 15, 869-884.
- PROTÉGÉ. 2013. *Protégé ontology editor* [Online]. Available: <http://protege.stanford.edu/> [Accessed 30 Nov 2015].
- RADA, R., MILI, H., BICKNELL, E. & BLETNER, M. 1989. Development and application of a metric on semantic nets. *IEEE Transactions on Systems, Man and Cybernetics*, 19, 17-30.
- RAZMERITA, L., ANGEHRN, A. & MAEDCHE, A. 2003. Ontology-based user modeling for knowledge management systems. *User Modeling*. Springer Berlin Heidelberg.
- RESNICK, P. & VARIAN, H. R. 1997. Recommender systems. *Communications of the ACM*, 40, 56-58.
- RESNIK, P. Using information content to evaluate semantic similarity in a taxonomy. Proceedings of the 14th International Joint Conference on Artificial Intelligence, 1995. 448-453.
- ROCCHIO, J. 1971. Relevance Feedback in Information Retrieval. *The SMART System: Experiments in Automatic Document Processing*. Prentice Hall.

- RUBENSTEIN, L. Z., HARKER, J. O., SALVÀ, A., GUIGOZ, Y. & VELLAS, B. 2001. Screening for undernutrition in geriatric practice developing the short-form mini-nutritional assessment (MNA-SF). *The Journals of Gerontology Series A: Biological Sciences and Medical Sciences*, 56, M366-M372.
- RUOTSALO, T., HAAV, K., STOYANOV, A., ROCHE, S., FANI, E., DELIAI, R. & HYVÖNEN, E. 2013. SMARTMUSEUM: a mobile recommender system for the web of data. *Web semantics: Science, services and agents on the world wide web*, 20, 50-67.
- RUTE-PÉREZ, S., SANTIAGO-RAMAJO, S., HURTADO, M. V., RODRÍGUEZ-FÓRTIZ, M. J. & CARACUEL, A. 2014. Challenges in software applications for the cognitive evaluation and stimulation of the elderly. *Journal of NeuroEngineering and Rehabilitation*, 11, 88.
- SALTON, G. & BUCKLEY, C. 1988. Term weighting approaches in automatic retrieval. *Information Processing and Management*, 24, 513-523.
- SÁNCHEZ, D., BATET, M., ISERN, D. & VALLS, A. 2012. Ontology-based semantic similarity A new feature-based approach. *Expert Systems with Applications*, 39, 7718-7728.
- SARWAR, B., KARYPIS, G., KONSTAN, J. & RIEDL, J. Application of dimensionality reduction in recommender system-a case study. Proceedings of the 6th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining, Workshop on Web Mining for E-Commerce -- Challenges and Opportunities, 2000.
- SCHICKEL-ZUBER, V. & FALTINGS, B. 2007. OSS: A Semantic Similarity Function based on Hierarchical Ontologies. *IJCAI* 7, 551-556.
- SENPE & SEGG 2007. *Valoración nutricional en el anciano. Documento de Consenso.*, Galenitas-Nigra Trea.
- SHEARER, R., MOTIK, B. & HORROCKS, I. 2008. HermiT: A Highly-Efficient OWL Reasoner. *OWLED*, 432, 91.
- SHENOY, K. M., SHET, K. C. & ACHARYA, U. D. 2012. A new similarity measure for taxonomy based on edge counting. *International Journal of Web & Semantic Technology.* , 3, 23-30.
- SIEG, A., MOBASHER, B. & BURKE, R. D. 2007. Learning Ontology-Based User Profiles: A Semantic Approach to Personalized Web Search. *IEEE Intelligent Informatics Bulletin*, 8, 7-18.
- SIRIN, E., PARSIA, B., GRAU, B. C., KALYANPUR, A. & KATZ, Y. 2007. Pellet: A practical owl-dl reasoner. *Web Semantics: science, services and agents on the World Wide Web*, 5, 51-53.
- SIVILAI, S., SNAE, C. & BRUECKNER, M. 2012. Ontology-Driven Personalized Food and Nutrition Planning System for the Elderly. *ICB MIS*.
- SIXSMITH, A. & GUTMAN, G. M. 2013. *Technologies for Active Aging*, New York, Springer.
- SKILLEN, K., CHEN, L., NUGENT, C. D., DONNELLY, M. P. & SOLHEIM, I. A user profile ontology based approach for assisting people with dementia in mobile environments. Annual International Conference on Engineering in Medicine and Biology Society (EMBC) 2012 2012. IEEE, 6390-6393.
- SKIPPER, A., FERGUSON, M., THOMPSON, K., CASTELLANOS, V. H. & PORCARI, J. 2012. Nutrition Screening Tools An Analysis of the Evidence. *Journal of Parenteral and Enteral Nutrition*, 36, 292-298.
- SNAE, C. & BRUCKNER, M. FOODS: a food-oriented ontology-driven system. 2nd IEEE International Conference on Digital Ecosystems and Technologies, 2008. IEEE, 168-176.
- SOBECKI, J., BABIAK, E. & ŚLANINA, M. 2006. Application of hybrid recommendation in web-based cooking assistant. *Knowledge-Based Intelligent Information and Engineering Systems*. Heidelberg: Springer Berlin.
- STRATTON, R. J., HACKSTON, A., LONGMORE, D., DIXON, R., PRICE, S., STROUD, M. & ELIA, M. 2004. Malnutrition in hospital outpatients and inpatients: prevalence,

- concurrent validity and ease of use of the 'malnutrition universal screening tool' ('MUST') for adults. *British Journal of Nutrition*, 92, 799-808.
- STUDER, R., BENJAMINS, V. R. & FENSEL, D. 1998. Knowledge engineering: Principles and methods. *Data & Knowledge Engineering*, 25, 161 – 197.
- SU, C. J., CHEN, Y. A. & CHIH, C. W. 2013. Personalized ubiquitous diet plan service based on ontology and web services. *International Journal of Information and Education Technology*, 5, 522-528.
- SUKSOM, N., BURANARACH, M., THEIN, Y. M., SUPNITHI, T. & NETISOPAKUL, P. A Knowledge-based Framework for Development of Personalized Food Recommender System. Proceedings of the 5th Int. Conf. on Knowledge, Information and Creativity Support Systems, 2010.
- SVENSSON, M., LAAKSOLAHTI, J., HÖÖK, K. & WAERN, A. 2000. A recipe based on-line food store. *Proceedings of the 5th international conference on Intelligent user interfaces*. ACM.
- THAKKER, D., DIMITROVA, V., LAU, L., DENAUX, R., KARANASIOS, S. & YANG-TURNER, F. A priori ontology modularisation in ill-defined domains. Proceedings of the 7th International Conference on Semantic Systems, 2011. ACM, 167-170.
- TINTAREV, N. & MASTHOFF, J. A survey of explanations in recommender systems. 23rd International Conference on Data Engineering Workshop, 2007. IEEE, 801-810.
- TRAJKOVA, J. & GAUCH, S. 2004. Improving Ontology-Based User Profiles. *RIAO*, 2004.
- TRIANA, F. C., GONZÁLEZ, C. R. & MARTÍN, P. M. 2006. Valoración nutricional en el anciano. *Medicine-Programa de Formación Médica Continuada Acreditado*, 9, 4037-4047.
- TRINIDAD RODRÍGUEZ, I., FERNÁNDEZ BALLART, J., CUCÓ PASTOR, G., BIARNÉS JORDÀ, E. & ARIJA VAL, V. 2008. Validación de un cuestionario de frecuencia de consumo alimentario corto: reproducibilidad y validez. *Nutrición Hospitalaria*, 23, 242-252.
- TSARKOV, D. & HORROCKS, I. 2006. FaCT++ description logic reasoner: System description. *Automated reasoning. Springer Berlin Heidelberg*, 292-297.
- VAILAS, L. I., NITZKE, S. A., BECKER, M. & GAST, J. 1998. Risk indicators for malnutrition are associated inversely with quality of life for participants in meal programs for older adults. *Journal of the American Dietetic Association*, 98, 548-553.
- VAN HEIJST, G., SCHREIBER, A. T. & WIELINGA, B. J. 1997. Using explicit ontologies in KBS development. *International Journal of Human-Computer Studies*, 46, 183–292.
- VAN PINXTEREN, Y., GELEIJNSE, G. & KAM, P. 2011. Deriving a recipe similarity measure for recommending healthful meals. *Proceedings of the 16th international conference on Intelligent user interfaces*. ACM.
- VELLAS, B., VILLARS, H., ABELLAN, G. & SOTO, M. E. 2006. Overview of the MNA®- It's history and challenges / discussion. *The journal of nutrition, health & aging*, 10, 456.
- VOZALIS, E. & MARGARITIS, K. G. 2003. Analysis of recommender systems algorithms. *The 6th Hellenic European Conference on Computer Mathematics & its Applications*, 732-745.
- W3C. 2001. *DAML+OIL Reference Description* [Online]. Available: <http://www.w3.org/TR/daml+oil-reference> [Accessed 3th March 2015].
- W3C. 2004. *OWL Web Ontology Language Reference* [Online]. Available: <http://www.w3.org/TR/owl-ref/> [Accessed 3th March 2015].
- W3C. 2009. *SKOS Simple Knowledge Organization System Reference* [Online]. Available: <http://www.w3.org/TR/skos-reference/> [Accessed 3th March 2015].
- W3C. 2012a. *OWL 2 Direct Semantics (Second Edition)* [Online]. Available: <http://www.w3.org/TR/owl2-direct-semantics/> [Accessed 30 November 2015].
- W3C. 2012b. *SPARQL 1.1* [Online]. Available: <http://www.w3.org/TR/sparql11-query/> [Accessed 10 November 2015].

- W3C. 2013. *RIF Overview (Second Edition)* [Online]. Available: <http://www.w3.org/TR/rif-overview/> [Accessed 10 October 2015].
- W3C. 2014a. *RDF Current Status* [Online]. Available: http://www.w3.org/standards/techs/rdf#w3c_all [Accessed 26th November 2015].
- W3C. 2014b. *RDF Schema 1.1 W3C Recommendation* [Online]. Available: <http://www.w3.org/TR/rdf-schema/> [Accessed 6th November 2015].
- W3C. 2014c. *Web Cryptography API - W3C Candidate Recommendation* [Online]. Available: <http://www.w3.org/TR/WebCryptoAPI/> [Accessed 10 October 2015].
- WAHLSTER, W. & KOBASA, A. 1989. *User models in dialog systems* Springer Berlin Heidelberg.
- WANG, J. Z., DU, Z., PAYATTAKOOL, R., PHILIP, S. Y. & CHEN, C. F. 2007. A new method to measure the semantic similarity of GO terms. *Bioinformatics*, 23, 1274-1281.
- WANG, P. 1998. Why recommendation is special? *Papers from the 1998 Workshop on Recommender Systems, part of the 15th National Conference on Artificial Intelligence*.
- WANG, R. Q. & KONG, F. S. 2007. Semantic-enhanced personalized recommender system. *International Conference on Machine Learning and Cybernetics*. IEEE.
- WIESNER, M. & PFEIFER, D. Adapting recommender systems to the requirements of personal health record systems. *Proceedings of the 1st ACM International Health Informatics Symposium, 2010*. ACM, 410-414.
- WORLD HEALTH ORGANIZATION 2002a. *Keep fit for life: meeting the nutritional needs of older persons*. Geneva, World Health Organization.
- WORLD HEALTH ORGANIZATION, W. 2002b. *Active ageing: A policy framework*.
- ZHANG, Q., HU, R., MAC NAMEE, B. & DELANY, S. J. 2008. *Back to the future: Knowledge light case base cookery*. Technical report, Dublin Institute of Technology.
- ZIEGLER, C. N. 2005. Semantic web recommender systems. *Current Trends in Database Technology-EDBT 2004 Workshops*. Springer Berlin Heidelberg.
- ZIEGLER, C. N., LAUSEN, G. & SCHMIDT-THIEME, L. Taxonomy-driven computation of product recommendations. *Proceedings of the thirteenth ACM international conference on Information and knowledge management, 2004*. ACM, 406-415.