

Ontologia aeroportuària orientada a l'assistència als passatgers

Estudiant: Juan Antonio Doblado Bretón

Titulació: Enginyeria en informàtica

Consultora: Maria Antonia Huertas Sánchez

Data: Gener 2013

Resum

En la actualitat, els usuaris dels aeroports demanden una gran varietat de serveis i productes que van molt més enllà dels associats a l'activitat bàsica de l'aviació comercial.

En la mateixa línia, les gestores de les infraestructures han adaptat el model de negoci per aprofitar aquestes expectatives dels passatgers i així obtenir ingressos per aquesta via.

Per altra banda, de la immersió en l'ús de les tecnologies per part de la població en general i els usuaris d'aeroports en particular, sorgeix la necessitat de desar la informació a disposició del públic per facilitar-los el seu pas per l'aeroport.

En el present treball, que s'emmarca dins de l'àrea de Representació del Coneixement i del Raonament, es proposa un model ontològic per a la representació i publicació de la informació implicada en el trànsit de passatgers per les terminals dels aeroports.

Paraules clau: *aeroport, ontologia, Web semàntica*

Índex de continguts

1. Introducció	5
1.1 Justificació	6
1.2 Objectius.....	8
1.3 Enfocament i mètode seguit	9
1.4 Planificació	9
1.5 Productes obtinguts	11
1.6 Organització de la memòria	12
2. Fonaments teòrics	13
2.1 La Web semàntica	13
2.2 Ontologies	15
2.3 Tecnologies i estàndards	18
2.4 Eines per a la creació i tractament d'ontologies	22
3. Disseny de l'ontologia aeroportuària	24
3.1 Metodologia	24
3.2 Domini i abast	25
3.3 Anàlisi d'ontologies existents	25
3.4 Identificació dels conceptes	26
3.5 Definició de la jerarquia de classes	28
3.6 Definició de les propietats	30
3.7 Definició de les facetes	32
4. Implementació de l'ontologia	35
4.1 Eines utilitzades.....	35
4.2 Nomenclatura adoptada	36
4.3 Implementació de la taxonomia.....	37
4.4 Implementació de les relacions	42
4.4.1 Object Properties	42
4.4.2 Data Properties	49
4.4.3 Restriccions	51
4.5 Integració de la informació	52
4.5.1 Conjunt de dades	54
4.6 Consultes	54
5. Conclusions	57
Bibliografia	58
Annex I. Codi font de PopOnt	60

Índex de figures

<i>Fig. 1.1. Principals causes d'estrès pels passatgers</i>	6
<i>Fig. 1.2. Els punts més estressants pels passatgers</i>	7
<i>Fig. 1.3. Conveniències dels passatgers</i>	7
<i>Fig. 1.4. Disponibilitat i ús de telèfons intel·ligents</i>	8
<i>Fig. 1.5. Diagrama de Gantt amb la planificació de tasques</i>	11
<i>Fig. 2.1. Aparició del terme ontologia en una obra de filosofia</i>	15
<i>Fig. 2.2. Arquitectura de la Web semàntica</i>	19
<i>Fig. 3.1. Identificació de les propietats</i>	30
<i>Fig. 3.2. Identificació de facetes</i>	32
<i>Fig. 4.1. Creació de classe en Protégé</i>	37
<i>Fig. 4.2. Vista de les classes de Protégé</i>	38
<i>Fig. 4.3. Esquema de la jerarquia de classes obtingut amb GraphViz</i>	40
<i>Fig. 4.4. Taula de correspondència entre classes i conceptes</i>	41
<i>Fig. 4.5. Detall d'una Object Property amb Protégé</i>	44
<i>Fig. 4.6. Quadre resum de les Object Properties implementades</i>	45
<i>Fig. 4.7. Propietats entre instàncies de classes</i>	48
<i>Fig. 4.8. Creació d'una Data Property amb rang definit per conjunt de literals</i> ...	49
<i>Fig. 4.9. Creació d'una Data Property amb rang de tipus cadena de caràcters</i> ...	50
<i>Fig. 4.10. Quadre resum de les Data Properties implementades</i>	50
<i>Fig. 4.11. Implementació de restriccions de cardinalitat amb Protégé</i>	52
<i>Fig. 4.12. Pestanya SPARQL Query en Protégé 4.2</i>	55
<i>Fig. 4.13. Resultat de consulta SPARQL</i>	55
<i>Fig. 4.14. Resultat de consulta SPARQL</i>	56
<i>Fig. 4.15. Resultat de consulta SPARQL</i>	56

1. Introducció

Avui dia els aeroports ja no són simplement un espai habilitat per que aterrin i s'enlairin avions i embarquin i desembarquin passatgers. A més d'aquests serveis bàsics i intrínsecs de l'activitat aeronàutica comercial, els aeroports ofereixen als passatgers multitud de serveis orientats, d'una banda, a satisfer les possibles necessitats de les persones i, d'una altra, a incrementar els ingressos d'explotació del negoci.

El projecte que es presenta consisteix en la creació d'una plataforma per a la representació del coneixement en l'àmbit de les activitats que realitzen els passatgers d'un aeroport durant el seu període d'estada així com els serveis que s'hi ofereixen, per tal que aquesta pugui servir de base per a la creació d'aplicacions que puguin explotar aquest coneixement i facilitar també la interconnexió amb altres fonts remotes de coneixement complementari.

Dins l'aeroport conflueixen diferents perfils d'usuaris de les instal·lacions: hi ha passatgers que esperen la sortida del seu vol, d'altres que acaben de desembarcar, també uns altres que fan connexió entre vols, sense oblidar però, als acompanyants que no puguen ni baixen de cap avió. Cada un d'aquests perfils fa recorreguts diferents per l'aeroport i, té a més, necessitats intrínseques: el passatger que ha arribat en un vol ha d'anar a recollir l'equipatge a la sala corresponent i, potser després, agafar un taxi. A més, l'interès en els diferents serveis oferts pot variar en funció de les preferències i circumstàncies particulars de cada persona.

Per dur a terme aquesta tasca, s'ha de dissenyar un model que sigui capaç de representar informació i relacions referents a infraestructures aeroportuàries, vols, companyies aèries, punts d'interès, serveis i productes que s'hi ofereixen, etc. En les següents pàgines es presenta el disseny i implementació d'aquest model mitjançant una ontologia.

1.1 Justificació

Estudis de mercat actuals [15] realitzats per una prestigiosa companyia del sector aeroportuari a nivell mundial presenten els següents resultats sobre l'opinió dels passatgers i les tendències de l'ús de les TIC als aeroports.

- La principal preocupació dels passatgers és la pèrdua de temps d'espera i les cues. (fig. 1.1)

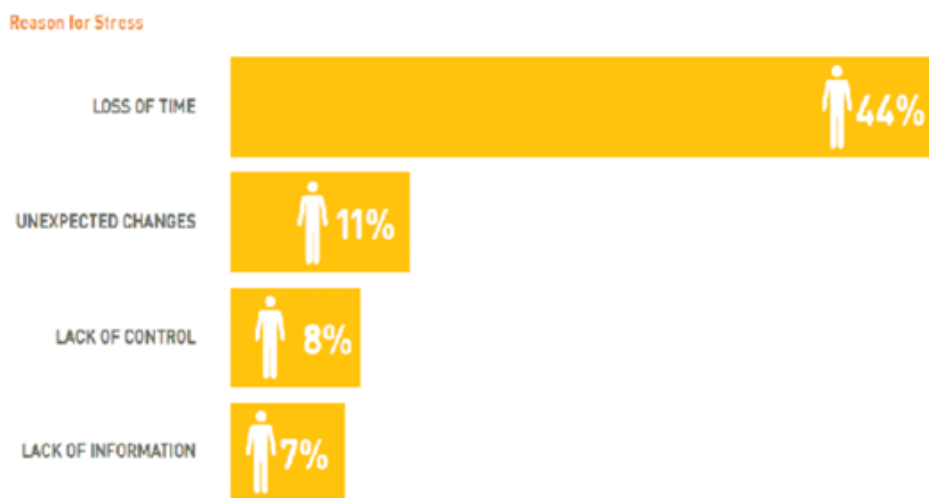


Fig. 1.1. Principals causes d'estrès pels passatgers

- Els punts considerats més estressants són els controls de seguretat, la connexió de vols i la facturació per la incertesa del temps que els hi comportarà el procés. (fig. 1.2)



Fig. 1.2. Els punt més estressants pels passatgers

- Els passatgers creuen convenient disposar d'eines per gestionar-s'hi ells mateixos els processos facturació i embarcament. (fig. 1.3)

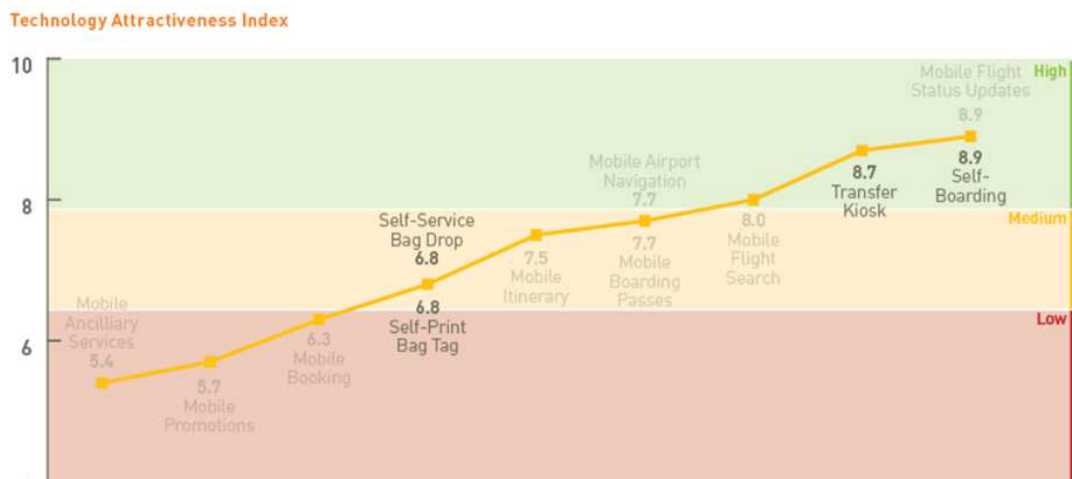


Fig. 1.3. Conveniències dels passatgers

- L'augment del número de passatgers que disposen de telèfons intel·ligents i els utilitzen en els processos de facturació i embarcament és considerable. (fig. 1.4)

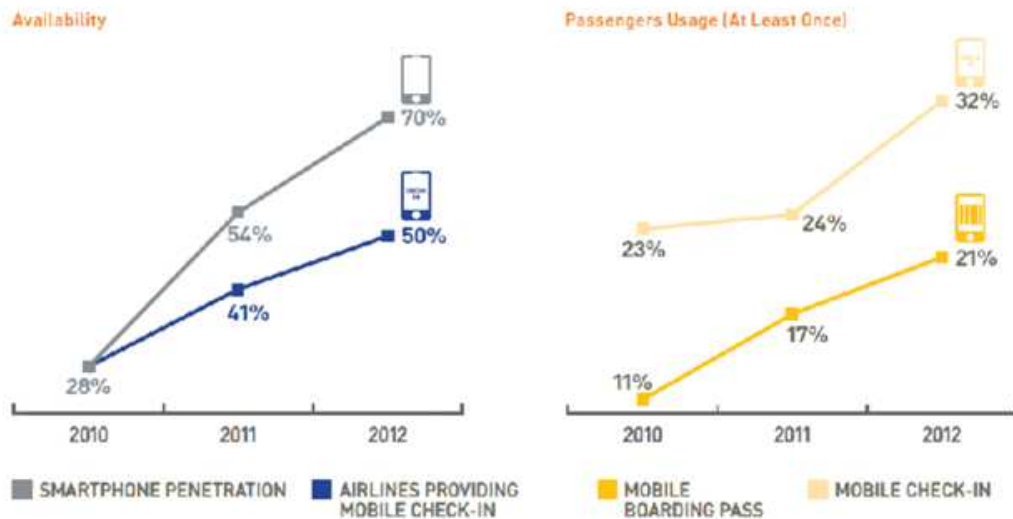


Fig. 1.4. Disponibilitat i ús de telèfons intel·ligents

Vist aquest escenari , seria enormement beneficiós disposar d'una plataforma per concentrar-hi tota la informació que pugui ser d'utilitat als passatgers en la seva estància a l'aeroport. Aquesta informació podria ser utilitzada, per exemple dins d'una aplicació mòbil orientada al passatger, però també per la seva explotació a través de diferents aplicacions que es volguessin desenvolupar amb finalitats diverses, com ara eines per estimular el consum als establiments comercials, o la vinculació del coneixement amb d'altres fonts externes en el context de *Linked Data*.

1.2 Objectius

Els objectius que es pretenen assolir amb la realització d'aquest projecte són:

1. Modelar el coneixement associat a les activitats i serveis que ofereix un aeroport als seus passatgers i representar-lo mitjançant una ontologia.
2. Dissenyar els mecanismes necessaris per a la integració de les dades dinàmiques en l'ontologia.
3. Oferir l'accés a l'ontologia, que farà les funcions de pedra angular per a futurs desenvolupaments els quals podran fer-ne

ús i explotar el coneixement per satisfer les seves finalitats concretes.

1.3 Enfocament i mètode seguit

Per assolir els objectius proposats, el projecte ha estat dividit en tres fases consecutives: una primera fase d'investigació anàlisi per recopilar informació sobre els sistemes de representació del coneixement i les eines a l'abast; una segona fase en la qual s'ha elaborat el disseny conceptual del model; i una tercera per a la implementació.

1.4 Planificació

La planificació del projecte s'ha organitzat en quatre fases consecutives en el següent ordre: Planificació, Anàlisi, Disseny i Implementació. A més, s'han contemplat, com tasques transversals, la documentació i el seguiment que s'hauran de completar durant tota la durada del projecte.

A banda de les fites de seguiment que ja venen marcades en el calendari de l'assignatura, s'han establert les següents:

Fita	Data límit
Fi de la fase de planificació	29/10/2012
Fi de la fase d'anàlisi	23/10/2012
Fi de la fase de disseny	20/11/2012
Fi de la fase d'implementació	09/01/2013
Fi de la documentació	18/01/2013

La descomposició de tasques, dependències i planificació temporal inicial queden reflectides en el diagrama de Gantt que es mostra a la figura (fig. 1.5). Cal dir que finalment es va produir una petita desviació en el temps emprat a las fases d'anàlisi i disseny que provocà un endarreriment en la fita de fi d'implementació i reajustaments en les tasques de documentació.

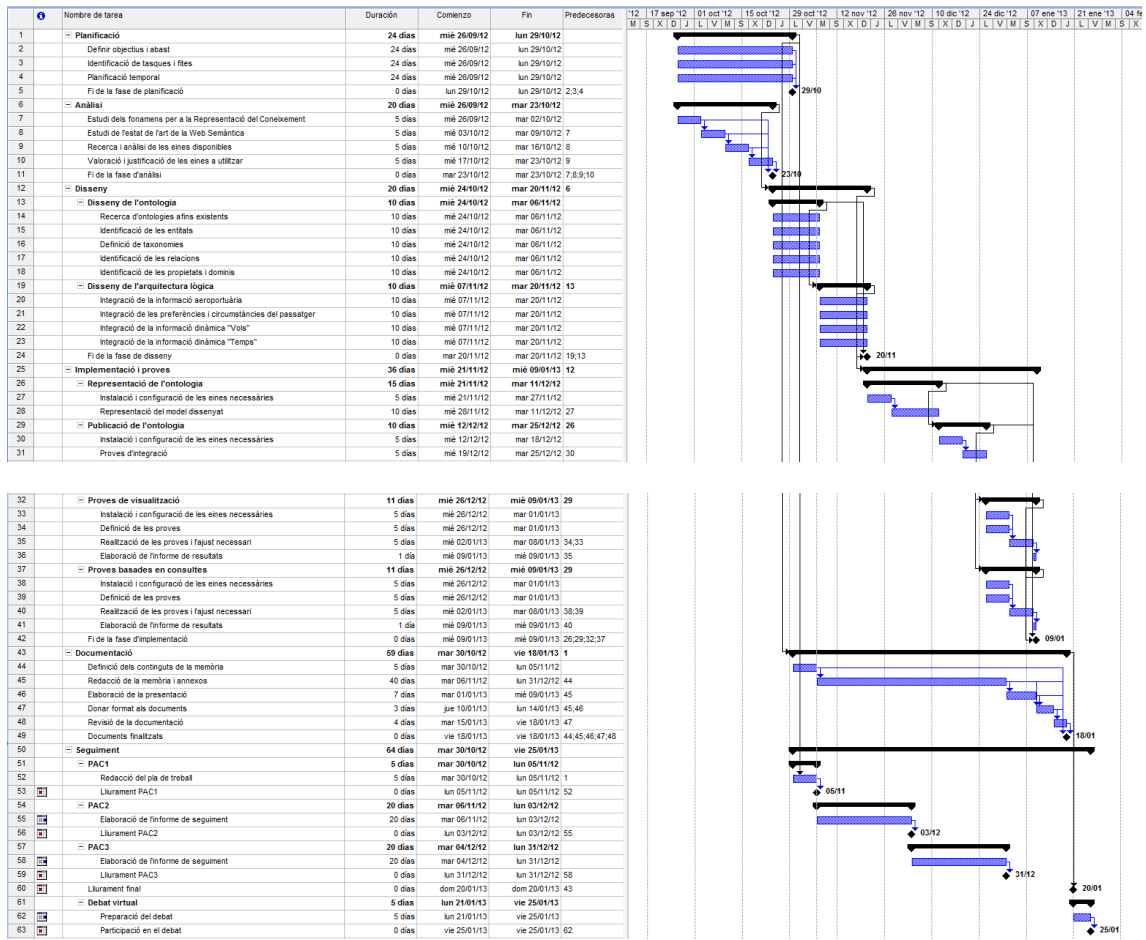


Fig. 1.5. Diagrama de Gantt amb la planificació de tasques

1.5 Productes obtinguts

Els productes obtinguts en finalitzar el projecte i que es lliuraran mitjançant suport electrònic seran:

- Ontologia aeroportuària implementada
- Codi font del programari desenvolupat per a la integració d'informació
- La present memòria

1.6 Organització de la memòria

A banda d'aquesta introducció, la memòria està organitzada en els capítols següents:

- Fonaments teòrics, on es fa un repàs als conceptes de Web semàntica, les ontologies, i l'estat de les tecnologies, estàndards i eines associades.
- Disseny de l'ontologia, en què en primer lloc es presenta un mètode i, a continuació, s'expliquen els detalls de cada pas del procés per al problema que ens ocupa.
- A continuació es detalla el procés d'implementació del model conceptual obtingut mitjançant l'eina *Protégé*, així com la solució adoptada per integrar-hi la informació.
- Finalment es mostren les conclusions extretes del treball realitzat.

2. Fonaments teòrics

En les pàgines següents, es pretén obtenir una visió dels conceptes en què es fonamenta el present projecte: la Web semàntica, les ontologies, i les tecnologies associades.

2.1 La Web semàntica

La Web actual està formada per milions de documents HTML en els que la informació es descriu mitjançant text i continguts multimèdia. Aquesta forma de representar la informació és molt intuïtiva per a les persones, però no és pas adequada per a les màquines, que hi treballen millor amb bases de dades on la informació és molt estructurada. La web semàntica proposa descriure les dades i les seves relacions mitjançant etiquetes semàntiques, tot utilitzant estàndards recolzats en llenguatges formals en els que les propietats d'expressivitat i raonament són ben conegudes.

La Web semàntica (Berners-Lee et al, 2001) proposa superar les limitacions de la Web actual mitjançant la introducció de descripcions explícites del significat, la estructura interna i la estructura global dels continguts i serveis disponibles en la WWW. Davant la semàntica implícita, el creixement caòtic de recursos, i l'absència d'una organització clara de la Web actual, la Web semàntica aposta per classificar, dotar d'estructura i anotar els recursos amb semàntica explícita i processable per màquines.

La Web semàntica (de l'anglès *semantic Web*) es basa en la idea d'afegir metadades semàntiques a la WWW. Aquesta informació afegida, que descriu el contingut, el significat i la relació de les dades, s'han de proporcionar de manera formal, per a que d'aquesta manera, sigui possible avaluar-les automàticament mitjançant màquines que la processin. L'objectiu és millorar Internet fent créixer la interoperabilitat entre els sistemes informàtics i reduir la mediació d'operadors humans.

Es tracta d'un corrent, promogut pel propi inventor de la Web i president del consorci W3C, l'últim objectiu del qual és aconseguir

que les màquines puguin entendre, i per tant utilitzar allò que conté la Web. Aquesta nova Web seria poblada per agents de programari capaços de navegar i realitzar operacions per nosaltres per tal d'estalviar-nos feina i optimitzar els resultats. Per aconseguir aquesta fita, la Web semàntica proposa descriure els recursos de la Web amb representacions processables, és a dir, que puguin ser enteses no sols per persones, sinó també per programes que pugin assistir, representar o reemplaçar a les persones en tasques rutinàries o inabastables per a un humà. Les tecnologies de la Web semàntica cerquen desenvolupar una Web més cohesionada, on sigui encara més fàcil localitzar, compartir i integrar informació i serveis, per a treure un major partit dels recursos a l'abast en la Web. Tim Berners-Lee, el creador de la idea, la va expressar de la següent manera (Berners-Lee et al, 2001):

"El meu somni és una Web en la que les màquines siguin capaces d'analitzar totes les dades –contingut, enllaços i transaccions entre la gent i els ordinadors-. La 'Web Semàntica' que ho faria possible és per arribar encara, però quan arribi, la rutina de les nostres compres, burocràcia i vida diària serà gestionada per màquines comunicant-se amb màquines. Els 'agents intel·ligents' que han estat anunciats durant dècades, esdevindran per fi una realitat."

La idea és que les dades puguin ser utilitzades i interpretades pels ordinadors sense la necessitat de supervisió humana, de manera que els agents Web puguin ser dissenyats per tractar la informació allotjada en les pàgines Web de manera semiautomàtica. Es tracta de convertir la informació en coneixement amb la referència de les dades dins de les pàgines Web a metadades amb un esquema comú consensuat sobre algun domini. Les metadades no especificaran tan sols l'esquema de dades que ha d'aparèixer a cada instància, sinó que a més, podran tenir informació addicional de com fer-hi deduccions. Es a dir, axiomes que es podran aplicar als diferents dominis de que tracti el coneixement emmagatzemat. Els agents Web no sols trobaran la informació de forma precisa, sinó que també podran realitzar inferències mitjançant la cerca d'informació relacionada i amb els requeriments indicats per l'usuari.

Descriure el coneixement a nivell semàntic és una tasca costosa i complicada, així que resulta indispensable disposar de tecnologies i eines que potencien la possibilitat de compartir i reutilitzar aquest coneixement. Un dels principals mecanismes proposats per tal d’alleugerir aquest problema és la utilització d’ontologies. Les ontologies permeten representar el coneixement de manera intuïtiva i extensible mitjançant l’ús de jerarquies conceptuais. La comunitat de la Web semàntica ha invertit molt esforç per facilitar-ne la seva creació i manteniment, desenvolupant llenguatges de representació estàndard, editors d’ontologies, etc.

2.2 Ontologies

La paraula ontologia prové del grec i el seu significat és “estudi de l’ésser”. Encara que la preocupació de l’estudi d’aquesta part de la metafísica té l’origen als primers filòsofs grecs, no va ser fins a començaments del segle XVII que comencés a utilitzar-s’hi per Rudolf Goclenius a “*Lexicon philosophicum quo tanquam clave philosophiae fores aperiuntur*” escrit en 1613 i on apareixia una anotació que deia: “*ontologia, philosophia de ente*”. Posteriorment, al segle XVIII Christian Wolff va popularitzar el terme en cercles filosòfics. Aquest apareix al títol de la seva obra “*Philosophia prima sive ontologia methodo scientifica pertractata, qua omnes cognitionis hmanae principia continentur*” que va ser publicada per primer cop a l’any 1730 [14].



Fig. 2.1 Aparició del terme ontologia en una obra de filosofia

Mes tard, Kant al segle XVIII (Kant, 2001) proposà la següent definició:

“La filosofia transcendental és el sistema de totes nostres cognicions pures a priori, que podem anomenar ontologia. Així, ontologia tracta amb coses en general, des d’abstractes fins particulars. Abasta tots els conceptes purs de la comprensió i tots els principis de la raó. Les ciències principals que pertanyen a la metafísica són: ontologia, cosmologia i teologia. Ontologia és una pura doctrina d’element de tota la nostra cognició al complet; conté la suma de tots els nostres conceptes purs que podem tenir a priori sobre les coses.”

Feta aquesta introducció, ens centrarem en l’ontologia des del punt de vista de la intel·ligència artificial i els sistemes de representació del coneixement, que és l’àrea on s’emmarca el present projecte. Dins d’aquest paradigma, una de les definicions més esteses és la donada per Tom Gruber (Gruber, 1993):

“Una ontologia és una especificació explícita d’una conceptualització. El terme prové de la filosofia, on una ontologia és un recompte sistemàtic de l’existència. En sistemes d’Intel·ligència Artificial, allò que existeix és allò que pot ser representat. Quan el coneixement d’un domini es representa mitjançant un formalisme declaratiu, el conjunt d’objectes que pot ser representat s’anomena univers del discurs. Aquests conjunts d’objectes, i les relacions que s’estableixen entre ells, són reflectits en un vocabulari amb el qual representem el coneixement en un sistema basat en coneixement. Així, en el context d’IA, podem descriure l’ontologia d’un programa com un conjunt de termes. En tal ontologia, les definicions associen noms d’entitats de l’univers del discurs amb textos comprensibles pels humans que descriuen el significat dels noms, i axiomes formals que limiten la interpretació i bon ús dels termes mencionats. Formalment, una ontologia és una teoria lògica.”

En considerar la definició anterior com massa generalista, Nicola Guarino (Guarino, 1995) va donar una definició alternativa:

“En el sentit filosòfic, podem referir-nos a una ontologia com un sistema particular de categories que representa una certa visió del món. Com tal, aquest sistema no depèn d’un llenguatge particular: l’ontologia d’Aristòtil és sempre la mateixa, independentment del llenguatge utilitzat per descriure-la. Per altra banda, en el seu ús més típic en IA, una ontologia és un artefacte d’enginyeria constituït per

un vocabulari específic per a descriure una certa realitat, més un conjunt de supòsits explícits concernents al significat pretès de les paraules del vocabulari. Aquest conjunt de supòsits té generalment la forma de teories lògiques de primer ordre, on les paraules del vocabulari apareixen com predicats unaris o binaris, respectivament anomenats conceptes i relacions. En el cas més senzill, una ontologia descriu una jerarquia de conceptes relacionats per relacions de subsumpció; en els casos més sofisticats, s'afegeixen axiomes per a expressar altres relacions entre conceptes i restringir la possible interpretació.”

La interpretació d'aquestes definicions és que les ontologies proporcionen un vocabulari comú d'una àrea i defineixen, en diferents nivells de formalisme, el significat dels termes i relacions entre ells. El coneixement en les ontologies es formalitza principalment utilitzant cinc tipus de components: classes, relacions, funcions, axiomes i instàncies (Gruber, 1993).

- Classes o conceptes: Són les idees bàsiques que es volen formalitzar. Els conceptes poden ser classes d'objectes, mètodes, plans, estratègies, processos de raonament, etc. Les classes, habitualment, s'organitzen en taxonomies.
- Relacions: Representen un tipus d'interacció entre els conceptes del domini. Es defineixen formalment com qualsevol subconjunt d'un producte de n conjunts. Un tipus concret de relació binària és "subclasse de" que defineix la taxonomia de classes.
- Funcions: Són un tipus concret de relació, on s'hi identifica un element mitjançant el càlcul d'una funció que considera diferents elements de l'ontologia.
- Axiomes: Són expressions que sempre són certes. Poden complir diferents funcions, com ara definir el significat dels components o definir restriccions sobre el valors dels atributs, etc.
- Instàncies: S'utilitzen per a representar objectes determinats d'un concepte.

Existeixen diferents tipus d'ontologies, i s'utilitzen principalment dos criteris per a classificar-les: a) el tipus de coneixement contingut (Van Heijst, 1997); i b) la motivació de l'ontologia (Mizoguchi, 1995). Dins del primer grup trobem:

- Ontologies terminològiques lingüístiques
- Ontologies d'informació
- Ontologies per a modelar coneixement

Segons la motivació, es proposen tres categories:

- Ontologies del domini
- Ontologies de tasca
- Ontologies generals

Des de començaments dels noranta, les ontologies han esdevingut en un tema d'investigació important en diferents comunitats d'Intel·ligència Artificial, on podem incloure Enginyeria del Coneixement, Processament del Llenguatge Natural o Representació del Coneixement. Més recentment, la noció d'ontologia s'ha popularitzat en camps com integració intel·ligent d'informació, sistemes cooperatius d'informació, recuperació de la informació, comerç electrònic i gestió del coneixement. La raó per la qual les ontologies han esdevingut tan populars és, en gran mesura, degut a allò que prometen: una comprensió compartida i comuna d'algun domini que pot ser comunicada entre individus, possiblement en diferents situacions geogràfiques.

2.3 Tecnologies i estàndards

La W3C (*World Wide Web Consortium*) ha proposat un conjunt de tecnologies estàndard per a la representació de la informació en la web semàntica. Com es pot veure a la figura (fig. 2.2), la arquitectura bàsica que es proposa està formada per diferents capes conceptuals: la capa inferior descriu la informació a nivell sintàctic, la següent capa afegeix descripcions utilitzant, per exemple, ontologies i regles; a continuació venen els sistemes d'inferència o raonadors que gestionen la informació semàntica a nivell lògic, oferent diferents serveis de raonament; i finalment, a la capa superior es troben les aplicacions semàntiques.

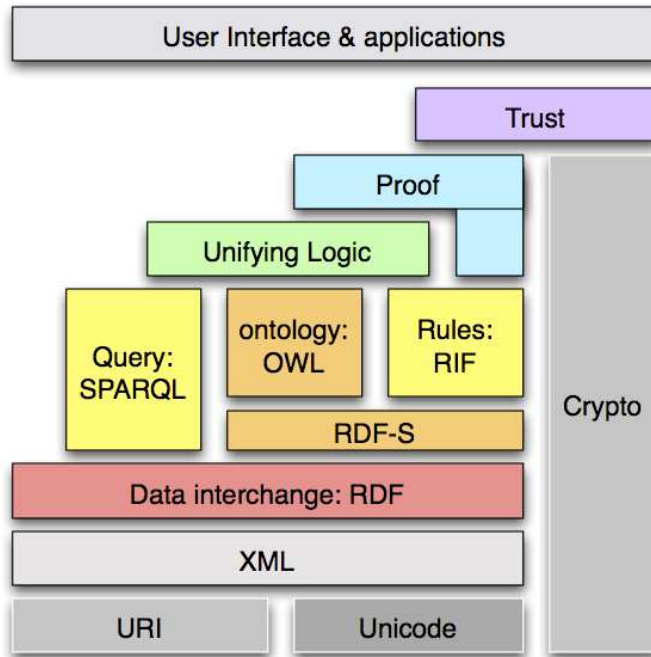


Fig. 2.2. Arquitectura de la Web semàntica

A continuació es descriuen breument cadascuna de les tecnologies implicades a l'esquema:

- **URI** (*Uniform Resource Identifier*) és l'estàndard per a la localització de recursos d'informació en el Web de forma inequívoca i única mitjançant cadenes de text. **Unicode** és l'estàndard per a la codificació de caràcters a nivell internacional. Addicionalment els IRI (*Internationalized Resource Identifier*) són similars als URI però permeten utilitzar caràcters d'altres llengües diferents a l'anglesa.
- **XML** (*eXtensible Markup Language*) és un llenguatge de marcat de propòsit general que permet a cada usuari definir les seves pròpies etiquetes. Permet representar i compartir informació estructurada de manera estàndard, facilitant la comunicació entre diferents aplicacions. Existeixen centenars de llenguatges basats en XML per a representar informació en diferents dominis (XHTML, RSS, MathML, SVL, ...), i cadascú defineix les seves pròpies etiquetes dotant-les-hi d'una semàntica particular. En particular, els llenguatges de la Web semàntica que es descriuen a continuació estan basats en XML.

XML Schema permet definir diferents tipus de documents XML establint restriccions sobre la seva estructura i contingut. Els documents XML que compleixen les restriccions imposades s'anomenen vàlids.

- **RDF** (*Resource Description Framework*) és un llenguatge de propòsit general per a representar informació a la Web. Permet expressar informació utilitzant triplets de la forma (subjecte, predicat, objecte). RDF permet definir models de dades en forma de graf dirigit on cada triplet defineix un node origen (subjecte), una aresta (predicat) i un node destí (objecte). Els nodes del graf poden contenir una URI que identifiqui una entitat concreta, un literal (cadena, números, ...), o bé un node en blanc. Les arestes es representen mitjançant URIs que hi estableixen relacions entre nodes o bé defineixen parells atribut-valor.

RDF-S (*RDF Schema*) és una extensió semàntica de RDF que proporciona els elements bàsics per a descriure ontologies: classes, propietats i instàncies. Les classes representen conjunts d'elements amb característiques comunes y solen estructurar-s'hi en jerarquies mitjançant la relació *rdfs:subClassOf*. Les instàncies són elements concrets de les classes i es relacionen amb ells mitjançant la propietat *rdf:type*. Finalment, les propietats són predicats RDF que defineixen relacions entre instàncies o entre instàncies i classes, i es solen caracteritzar pel seu domini i rang. Les propietats poden estructurar-s'hi de manera jeràrquica mitjançant la relació *rdfs:subPropertyOf*.

- **SPARQL** (acrònim recursiu de *SPARQL Protocol and RDF Query Language*) és un llenguatge de consultes sobre grafs RDF que permet recuperar triplets o subgrafs RDF que compleixin els patrons especificats.
- **OWL** (*Web Ontology Language*) permet representar de manera explícita el significat dels termes que formen un vocabulari, així com les relacions entre els esmentats termes. Aquesta representació de termes i les seves relacions s'anomena freqüentment ontologia.
OWL es basa en RDF i RDF-S però proporciona molts més recursos per a descriure ontologies: descripcions de classes

utilitzant combinacions lògiques d'altres classes (intersecció, unió, complement), restriccions de cardinalitat sobre les propietats, propietats algebraiques (reflexibilitat, transitivitat, ...), etc.

L'especificació original d'OWL defineix tres variants del llenguatge: OWL Lite, OWL DL i OWL Full. Cadascuna d'elles és una extensió de l'anterior en la que s'hi augmenta l'expressivitat del llenguatge. Per tant, les ontologies OWL Lite són ontologies vàlides OWL DL i aquestes, a la vegada són ontologies vàlides OWL Full. Desafortunadament, l'augment d'expressivitat del llenguatge comporta un augment del cost computacional necessari en els processos de raonament, per tant cada aplicació ha de triar la variant OWL que millor s'adapti a les seves necessitats.

- **OWL Lite** permet definir jerarquies utilitzant restriccions senzilles. L'expressivitat del llenguatge es manté molt limitada per afavorir la gestió eficient de les bases de coneixement. OWL Lite pot ser una bona opció per a representar tesaurus i taxonomies.
- **OWL DL** està pensat per als usuaris que volen la major expressivitat possible i a la vegada necessiten que els processos de raonament condueixin a una solució. OWL DL inclou totes les construccions d'OWL però estableix certes limitacions en el seu ús (per exemple, una classe no pot ser instància d'una altra classe).
- **OWL Full** és la variant més expressiva d'OWL però no proporciona cap tipus de garantia computacional. En OWL Full és possible definir entitats que siguin a la vegada classes, propietats e instàncies. La conseqüència és que els raonadors que utilitzen OWL Full tendeixen a ser correctes però incomplets.

Amb l'objectiu de facilitar la comprensió d'allò que aporta cada tecnologia al conjunt de la Web semàntica, es presenta el següent resum de la pila de llenguatges:

- XML permet representar i compartir informació estructurada a nivell sintàctic, sense imposar restriccions sobre el significat dels documents.
- XML Schema descriu l'estructura d'altres documents XML i estén XML amb tipus de dades.

- RDF representa models de dades en forma de graf dirigit on s'estableixen relacions entre entitats.
- RDFS permet definir classes d'entitats i propietats entre elles. A més hi proporciona una semàntica senzilla que permet definir jerarquies de classes i propietats.
- OWL incorpora noves construccions del llenguatge per a definir ontologies com relacions lògiques entre classes, restriccions de cardinalitat, etc.

2.4 Eines per a la creació i tractament d'ontologies

Així com les metodologies de disseny d'ontologies són variades, també ho són les eines relacionades amb el desenvolupament d'ontologies. Des de les inicials *d'Ontiligua Server*, *Ontosaurus*, fins les més recents com *Protégé*, *WebODE* i *OntoEdit*. Gómez-Pérez (Gómez-Pérez, 1998) ofereix la següent classificació:

- Eines de desenvolupament d'ontologies: Aquest grup inclou les eines que serveixen per a la construcció de noves ontologies o bé per a la reutilització de les existents. Les funcionalitats més destacables són l'edició i consulta, així com la visualització en formats gràfics i la importació i exportació d'ontologies.
- Eines de fusió i integració d'ontologies: Pretenen solucionar el problema de la combinació i la integració de diferents ontologies del mateix domini, allò que succeeix quan s'uneixen dues organitzacions diferenciades, o quan es pretén obtenir una ontologia de qualitat a partir d'altres existents.
- Eines d'avaluació d'ontologies: Apareixen com a instruments d'ajuda que han d'assegurar que les ontologies i les tecnologies relacionades tinguin un mínim nivell de qualitat.
- Eines basades en la anotació: Estan dissenyades per permetre als usuaris que introdueixin dades. La major part d'aquestes eines han aparegut amb la idea de la Web semàntica.

- Eines d'emmagatzemament i consulta: Són instruments que han estat ideats per a permetre utilitzar les ontologies de manera senzilla. La clau està en l'intent de que la Web esdevingui en una autèntica plataforma per a transmetre coneixement.
- Eines d'aprenentatge: Són utilitzades de forma semiautomàtica per a construir ontologies a partir del llenguatge natural.

3. Disseny de l'ontologia

En els apartats que segueixen s'exposen les bases d'una metodologia la qual serà utilitzada pel disseny del nostre model, i a continuació es descriuen cadascuna de les etapes del procés.

3.1 Metodologia

Per dur a terme el disseny de la present ontologia, el procediment adoptat es basa en les regles fonamentals suggerides en (Noy et al, 2005):

"1. No hi ha una forma correcta de modelar un domini –sempre hi ha alternatives viables-. La millor solució habitualment depèn de l'aplicació que tens al cap i les extensions que n'hi anticipes.

2. El desenvolupament d'ontologies és un procés necessàriament iteratiu.

3. Els conceptes en l'ontologia han de ser propers als objectes (físics i lògics) i relacions en el teu domini d'interès. Aquests són molt probablement substantius (objectes) o verbs (relacions) on oracions que descriuen el teu domini."

El mètode és un procediment iteratiu on a cada iteració l'ontologia evoluciona objecte del refinament de l'anterior, i consta dels passos:

1. Determinar el domini i abast de l'ontologia
2. Considerar la reutilització d'ontologies existents
3. Enumerar termes importants de l'ontologia
4. Definir les classes i la jerarquia de classes
5. Definir les propietats de les classes: *slots*
6. Definir les facetes dels *slots*
7. Crear les instàncies

Els apartats següents descriuen cadascú dels passos anteriors a excepció de la creació d'instàncies, tasca que es reserva per a la fase d'implementació.

3.2 Domini i abast de l'ontologia

Determinar el domini i l'abast és la primera activitat en el desenvolupament d'una ontologia. En aquest cas, l'objectiu a assolir és una ontologia que defineixi formalment els actors i elements que conflueixen en un aeroport des del punt de vista d'un usuari i més concretament d'un passatger per ser-ne l'usuari natural.

Per a determinar l'abast es tractarà de respondre a una sèrie de qüestions bàsiques:

Per a què s'utilitzarà l'ontologia?

L'ontologia s'utilitzarà com a base de coneixement perquè futurs desenvolupaments puguin fer-ne ús, amb els objectius principals de l'orientació dels passatgers en la seva estada a l'aeroport.

Per a quin tipus de preguntes la informació en l'ontologia haurà de proveir respostes?

En un principi, s'espera poder respondre a preguntes del tipus:

On puc comprar un diari?

El meu vol, surt amb retard?

Hi ha una cafeteria a prop de la meva porta d'embarcament?

Qui la utilitzarà i s'encarregarà del seu manteniment?

L'ús i el manteniment serien gestionats per professionals de les TIC de l'aeroport.

3.3 Anàlisi d'ontologies existents

Les indagacions fetes per a la reutilització d'ontologies, o parts, afins a la problemàtica suscitada en el present projecte no han donat resultats satisfactoris. S'ha valorat d'una banda, la utilització de la classe *Person* de l'ontologia *FOAF* [19] per a la representació dels passatgers, i d'altra l'ús de *geofile-ont* [18] per representar localitzacions geogràfiques i infraestructures, però finalment s'ha desestimat l'ús de cap de les dues.

3.4 Identificació dels conceptes

A continuació es presenta la llista inicial de termes acompanyats de la definició del concepte que representen dins el domini, i que conformaran la base de l'ontologia:

- *Relatiu a la infraestructura aeroportuària*
- Aeroport: Infraestructura per l'aterratge i enlairament d'aeronaus i l'embarcament i desembarcament de passatgers.
- Terminal de passatgers: Edifici o conjunt d'edificis que pertanyen a un aeroport i que té la funció de facilitar el trànsit de passatgers i hi ofereix els serveis necessaris a més d'altres complementaris.
- Edifici: Construcció que forma part d'una terminal de passatgers
- Zona: Cadascuna de les parts en les que es divideix un edifici
- Àrea d'embarcament: Àrea dedicada a l'embarcament de passatgers.
- Porta d'embarcament: Punt per on els passatgers de sortida abandonen l'aeroport per embarcar dins l'aeronau.
- Àrea de recollida d'equipatges: Àrea dedicada a la recollida d'equipatges per part dels passatgers procedents de vols d'arribada.
- Cinta de recollida d'equipatges: Cinta mecànica utilitzada per entregar els equipatges als passatgers procedents de vols d'arribada.
- Àrea de facturació: Àrea dedicada a la facturació d'equipatges dels passatgers de sortida.
- Mostrador de facturació: Mostrador on els passatges de sortida facturen els equipatges.

- Punt d'interès: Qualsevol element amb ubicació física que pugui ser d'interès per un usuari de la terminal de passatgers.
- Servei: Cadascú dels serveis que la terminal de passatgers ofereix als usuaris.
- Producte: Cadascú dels productes tangibles que la terminal de passatgers ofereix als usuaris.
- *Relatiu a vols*
 - Companyia aèria: Línia aèria que opera vols.
 - Vol d'arribada: Operació d'arribada d'una aeronau a l'aeroport i en la que hi desembarquen passatgers.
 - Vol de sortida: Operació de sortida d'una aeronau de l'aeroport i en la que hi embarquen passatgers.
 - Codi de vol: Codi alfanumèric utilitzat per a la identificació d'un vol, tant d'arribada com de sortida.
 - Aeronau: Avió.
 - Origen: Procedència d'un vol d'arribada.
 - Destí: Lloc de destinació d'un vol de sortida.
- *Relatiu a persones*
 - Passatger de sortida: Usuari que arriba a la terminal per agafar un vol de sortida.
 - Passatger d'arribada: Usuari que arriba a la terminal procedent d'un vol d'arribada.
 - Passatger en trànsit: Usuari que arriba a la terminal procedent d'un vol d'arribada i posteriorment agafarà un vol de sortida.
 -

- Acompanyant: Usuari que acompanya a un passatger a l'aeroport però no viatja.
 - Visitant: Usuari que no viatja i tampoc acompanya a cap passatger.
 - Característica personal: Propietat relativa a les persones, que sigui útil a l'ús que es vulgui fer de la informació.
- *Relatiu a altres aspectes*
- Ciutat: Ciutat internacional a la que pertany un aeroport.
 - País: País del món.
 - Temps: Data i hora actual.
 - Localització: Coordenades GPS per identificar un punt a l'espai.

3.5 Definició de la jerarquia de classes

Per a desenvolupar la jerarquia de classes, el procés seguit ha estat una combinació dels enfocaments *top-down* i *bottom-up* que comencen amb la definició dels conceptes més generals i els més específics respectivament. Les classes i taxonomia obtingudes es mostren a continuació:

- Avió
- Companyia aèria
- Vol
 - Vol d'arribada
 - Vol de sortida
- Codi de vol
- Persona
 - Passatger
 - Passatger d'arribada
 - Passatger de sortida
 - Passatger en trànsit
 - Acompanyant
 - Visitant

- Propietat personal
 - o Circumstància personal
 - o Preferència personal
- Servei i producte
 - o Servei
 - Servei del procés aeroportuari
 - Servei complementaris
 - Atenció al client
 - Informació
 - Suport al passatger
 - Transport
 - o Producte
 - Roba
 - Cultura
 - Decoració
 - Electrònica
 - Menjar i beure
 - Salut
 - Perfums i cosmètics
 - Viatges
 - Tabacs
 - Altres productes
- Concepte amb ubicació física
 - o Aeroport
 - o Ciutat
 - o País
 - o Zona mundial
 - o Edifici
 - o Zona d'edifici
 - o Terminal de passatgers
 - o Recurs del terminal
 - Recurs d'arribades
 - Area de recollida d'equipatges
 - Cinta de recollida d'equipatges
 - Recurs de sortides
 - Area de facturació d'equipatges
 - Mostrador de facturació
 - Area d'embarcament de passatgers
 - Porta d'embarcament
 - o Punt d'interès
 - De companyia aèria

- Comerç
 - Dispositiu
 - Instal·lació
 - Recurs del terminal
 - Espai virtual
- Temps

3.6 Definició de les propietats

A continuació es presenta una llista (fig. 3.1) amb les propietats identificades per a cada una de les classes.

Classe	Propietat
Avió	Matricula
	Descripció
Companyia aèria	Nom
	Codi IATA
	Codi OACI
Vol	Identificador
	Hora programada
	Hora estimada
	Es realitza amb un avió
	El realitza una companyia aèria
	Te assignat un terminal de passatgers
Vol d'arribada	Prové d'un aeroport origen
Vol de sortida	Hora d'inici facturació
	Hora de fi de facturació
	Hora d'inici d'embarcament
	Hora de fi d'embarcament
	Va cap a un aeroport destí
	Està associat amb un vol d'arribada
Codi de vol	Identificador
	Es operat per una companyia aèria
	Correspon a un vol
	Pot estar associat amb un codi de vol principal
	Té assignats recursos del terminal
Persona	Nom
	Edat
	Sexe
	Pot estar interessada en serveis i productes
	Te propietats personals
	Es originària d'un país

Passatger	Té assignat un codi de vol
	Pot viatjar amb un altre passatger
Passatger en trànsit	Té assignat un segon codi de vol
Acompanyant	Acompanya a un passatger
Propietat personal	Nom
	Descripció
Preferència personal	Nivell
Servei i producte	Nom
	Descripció
Concepte amb U.F.	Nom
	Descripció
	Longitud
	Latitud
Aeroport	Codi IATA
	Codi OACI
	Té terminal de passatgers
	Està en una Ciutat
Ciutat	És a un país
País	És a una zona mundial
Edifici	Nombre de plantes
	Pertany a una terminal de passatgers
Zona d'edifici	Planta
	Pertany a un edifici
	Es a prop d'una altra zona
Terminal de passatgers	Pertany a un aeroport
Recurs de terminal	Pertany a un terminal de passatgers
Cinta de recollida	Està dins una àrea de recollida
Mostrador de facturació	Està dins una àrea de facturació
Porta d'embarcament	Està dins una àrea d'embarcament
Punt d'interès	Està situat a una zona d'edifici
	Ofereix serveis i productes
PDI de companyia aèria	Pertany a companyia aèria
Temps	Valor

Fig. 3.1. Identificació de les propietats

3.7 Definició de les facetes

El següent pas és identificar cadascuna de les facetes que descriuen el tipus de valor i cardinalitat que poden prendre les propietats. S'han identificat també les propietats inverses existents. A la taula (fig. 3.2) es pot veure en detall.

Domini	Propietat	Rang	Cardinalitat (mín., màx.)	Propietat inversa
Avió	Matricula	Cadena caràcters	(1,1)	
	Descripció	Cadena caràcters	(0,1)	
Companyia aèria	Nom	Cadena caràcters	(1,1)	
	Codi IATA	Cadena caràcters	(0,1)	
	Codi OACI	Cadena caràcters	(0,1)	
Vol	Identificador	Cadena caràcters	(1,1)	
	Hora programada	Data i hora	(1,1)	
	Hora estimada	Data i hora	(1,1)	
	Es realitza amb un avió	Avió	(0,1)	Està assignat al vol
	El realitza una companyia aèria	Companyia aèria	(1,1)	Opera el vol
	Té assignat un terminal de passatgers	Terminal de passatgers	(1,1)	Està assignat al vol
Vol d'arribada	Prové d'un aeroport origen	Aeroport	(1,1)	
Vol de sortida	Hora d'inici facturació	Data i hora	(0,1)	
	Hora de fi de facturació	Data i hora	(0,1)	
	Hora d'inici d'embarcament	Data i hora	(0,1)	
	Hora de fi d'embarcament	Data i hora	(0,1)	
	Va cap a un aeroport destí	Aeroport	(1,1)	
	Està associat amb un vol d'arribada	Vol d'arribada	(1,1)	Està associat amb un vol de sortida
Codi de vol	Identificador	Cadena caràcters	(1,1)	
	Es operat per una companyia aèria	Companyia aèria	(1,1)	Opera codi de vol
	Correspon a un vol	Vol	(1,1)	Té codi de vol
	Pot estar associat amb un codi de vol principal	Codi de vol	(0,1)	
	Té assignats	Recurs de	(0,n)	Està assignat a

	recursos del terminal	terminal		un codi de vol
Persona	Nom	Cadena caràcters	(1,1)	
	Edat	Número	(0,1)	
	Sexe	{Home,Dona}	(0,1)	
	Pot estar interessada en serveis i productes	Servei i producte	(0,n)	Pot ser interessant per persona
	Te propietats personals	Propietat personal	(0,n)	
	Es originària d'un país	País	(0,1)	
Passatger	Té assignat un codi de vol	Codi de vol	(1,1)	
	Pot viatjar amb un altre passatger	Passatger	(0,n)	
Passatger en trànsit	Té assignat un segon codi de vol	Codi de vol	(1,1)	
Acompanyant	Acompanya a un passatger	Passatger	(1,n)	És acompanyat per
Propietat personal	Nom	Cadena caràcters	(1,1)	
	Descripció	Cadena caràcters	(0,1)	
Preferència personal	Nivell	{Baix,Mig,Alt}	(0,1)	
Servei i producte	Nom	Cadena caràcters	(1,1)	
	Descripció	Cadena caràcters	(0,1)	
Concepte amb U.F.	Nom	Cadena caràcters	(1,1)	
	Descripció	Cadena caràcters	(0,1)	
	Longitud	Número	(0,1)	
	Latitud	Número	(0,1)	
Aeroport	Codi IATA	Cadena caràcters	(1,1)	
	Codi OACI	Cadena caràcters	(1,1)	
	Té terminal de passatgers	Terminal de passatgers	(1,n)	Pertany a aeroport
	Està en una Ciutat	Ciutat	(1,1)	Té aeroport
Ciutat	És a un país	País	(1,1)	Té ciutat
País	És a una zona mundial	Zona mundial	(1,1)	Conté país
Edifici	Nombre de plantes	Número	(0,1)	
	Pertany a una terminal de passatgers	Terminal de passatgers	(1,1)	Està en edifici
Zona d'edifici	Planta	Número	(0,1)	
	Pertany a un edifici	Edifici	(1,1)	Conté zones
	Es a prop d'una	Zona d'edifici	(0,n)	

	altra zona			
Terminal de passatgers	Pertany a un aeroport	Aeroport	(1,1)	Té terminal de passatgers
Recurs de terminal	Pertany a un terminal de passatgers	Terminal de passatgers	(1,1)	Conté recursos de terminal
Cinta de recollida	Està dins una àrea de recollida	Area de recollida	(1,1)	Conté cintes
Mostrador de facturació	Està dins una àrea de facturació	Area de facturació	(1,1)	Conté mostradors
Porta d'embarcament	Està dins una àrea d'embarcament	Area d'embarcament	(1,1)	Conté portes
Punt d'interès	Està situat a una zona d'edifici	Zona d'edifici	(1,1)	Conté punts d'interès
	Ofereix serveis i productes	Servei i producte	(0,n)	És ofert a punt d'interès
PDI de companyia aèria	Pertany a companyia aèria	Companyia aèria	(1,1)	Té punts d'interès
Temps	Valor	Data i hora	(1,1)	

Fig. 3.2. Identificació de facetes

4. Implementació de l'ontologia

Als apartats que hi venen a continuació, s'expliquen els detalls de la creació de classes i propietats de l'ontologia, així com les eines utilitzades i els mecanismes definits per incorporar-hi les instàncies. A la finalització d'aquesta fase haurem obtingut un arxiu amb l'ontologia representada en llenguatge OWL.

4.1 Eines utilitzades

Els criteris de selecció per a les eines de desenvolupament han estat: 1) que fossin de llicència d'utilització lliure; 2) que estiguessin basats en tecnologies actuals; i 3) que hi existís documentació de referència accessible a la web. En aquest sentit, s'enumeren a continuació les eines utilitzades:

- **Protégé.** Per a la implementació de l'ontologia, s'ha triat *Protégé* [16] en la versió 4.1 (distribució més recent). *Protégé* és un programari lliure i de codi obert desenvolupat per la Universitat d'*Stanford* amb finalitats d'investigació biomèdica. Podríem catalogar aquesta eina dins del grup d'eines de desenvolupament d'ontologies esmentat en capítols anteriors. Com a tal, destaca per la seva portabilitat entre diferents plataformes i ofereix una potent interfície gràfica que facilita el desenvolupament de l'ontologia sense haver de preocupar-se pels aspectes sintàctics del llenguatge. Dóna suport als llenguatges RDF, RDF-S i OWL (inclosa la versió 2.0). Degut al seu ús extens, existeixen multitud de *plug-ins* que es poden instal·lar per afegir-hi funcionalitat. Altre aspecte valorat positivament ha estat l'existència d'abundant documentació, prenent la publicació de Matthew Horridge (Horridge, 2011) com a guia de referència. S'ha fet servir també la versió 4.2 que es troba actualment en fase beta i hi ofereix funcionalitat per a realitzar consultes amb llenguatge SPARQL.
- **Apache Jena Ontology API.** És un conjunt d'eines de programació per a manipular ontologies especificades en RDF i OWL, que utilitzen el llenguatge *Java*. Ha estat utilitzat per desenvolupar un mecanisme d'integració de dades en l'ontologia.

- **OPPL API** mitjançant *plug-in* instal·lat a *Protégé*. OPPL (*Ontology Pre-Processor Language*) És un llenguatge per a definir *scripts* de manipulació d'ontologies basades en OWL. És especialment útil quan s'han de manipular ontologies amb una gran quantitat d'instàncies o fer tasques repetitives. Es valora positivament la simplicitat de la seva sintaxi i gramàtica.
- **Graphviz**. Es tracta d'un programari de codi obert per a la representació i visualització d'informació estructurada en forma de diagrames o grafs. Des de *Protégé* s'estableix un mecanisme per a la seva integració.

4.2 Nomenclatura adoptada

Els noms utilitzats per designar les classes i propietats de l'ontologia corresponen als equivalents en llengua anglesa dels que s'han definit en el disseny conceptual del model. El motiu és que en el món aeroportuari s'utilitzen habitualment mots anglesos per fer referència a molts dels conceptes que hi tractem.

Els noms de les classes comencen amb lletra majúscula i no s'utilitzen espais, sinó que les paraules s'escriuen juntes i s'indica el començament d'una nova paraula escrivint la primera lletra d'aquesta també amb majúscula. Es a dir, designem "*FlightOperation*" i no "*Flight operation*" o "*flight_operation*".

Quan es creen classes subordinades, s'afegeix al nom, habitualment encara que no sempre, com a sufix, el nom de la superclasse a què pertanyen. Per exemple, *Departure_FlightOperation*.

Els noms de les propietats comencen amb lletra minúscula, no s'utilitzen espais i es segueix la mateixa norma que per les classes quan està format per més d'una paraula.

4.3 Implementació de la taxonomia

La creació de la jerarquia de classes és un procés que no revist gaire dificultat. Una darrere l'altra es van creant les classes i establint els nivells de l'arbre de jerarquies (fig. 4.1 a 4.3).

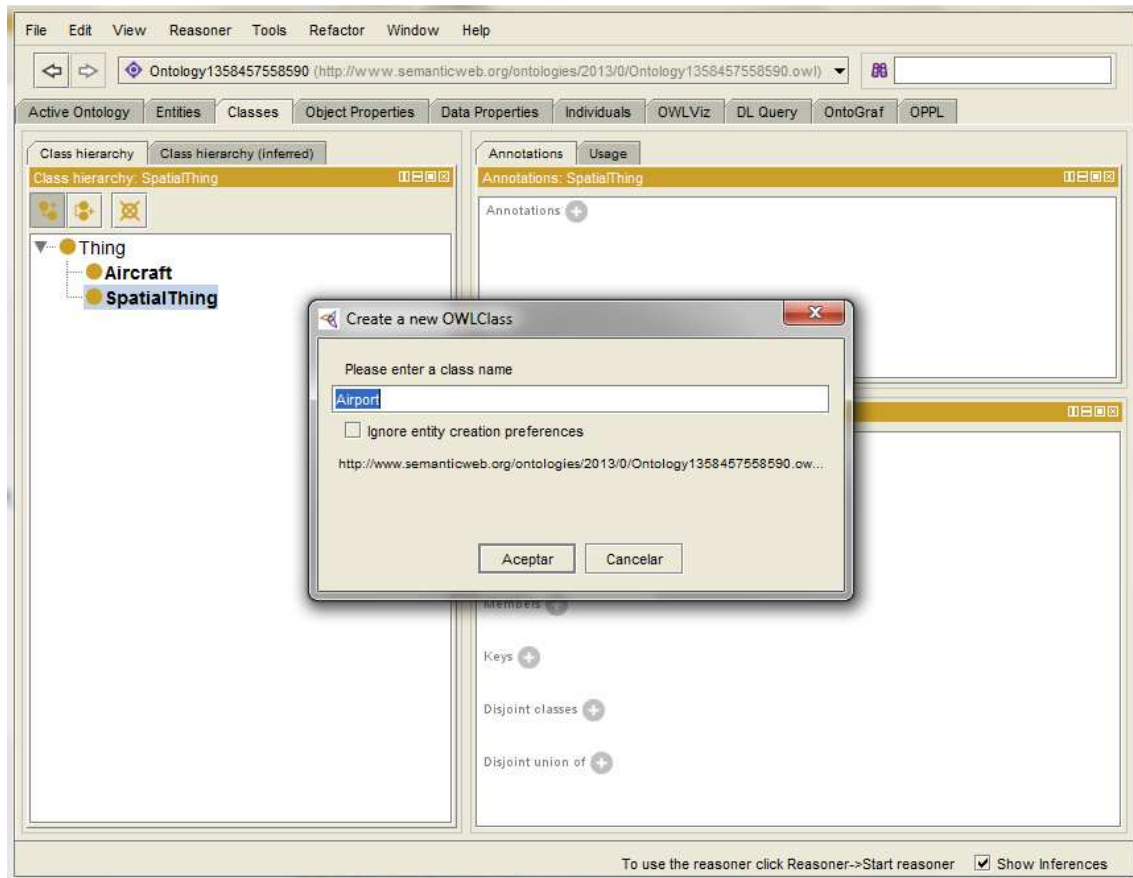
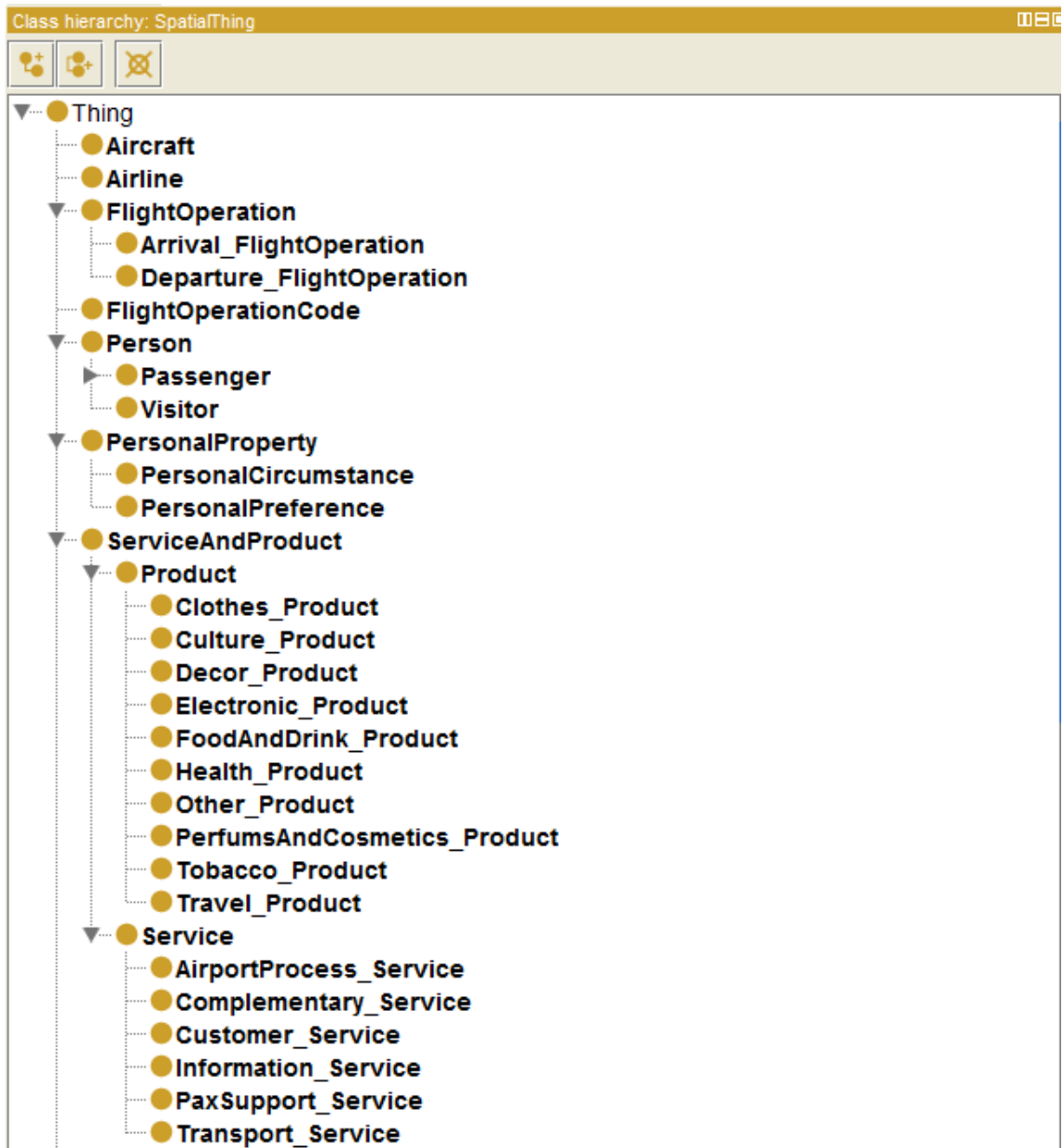


Fig. 4.1. Creació de classe en *Protégé*



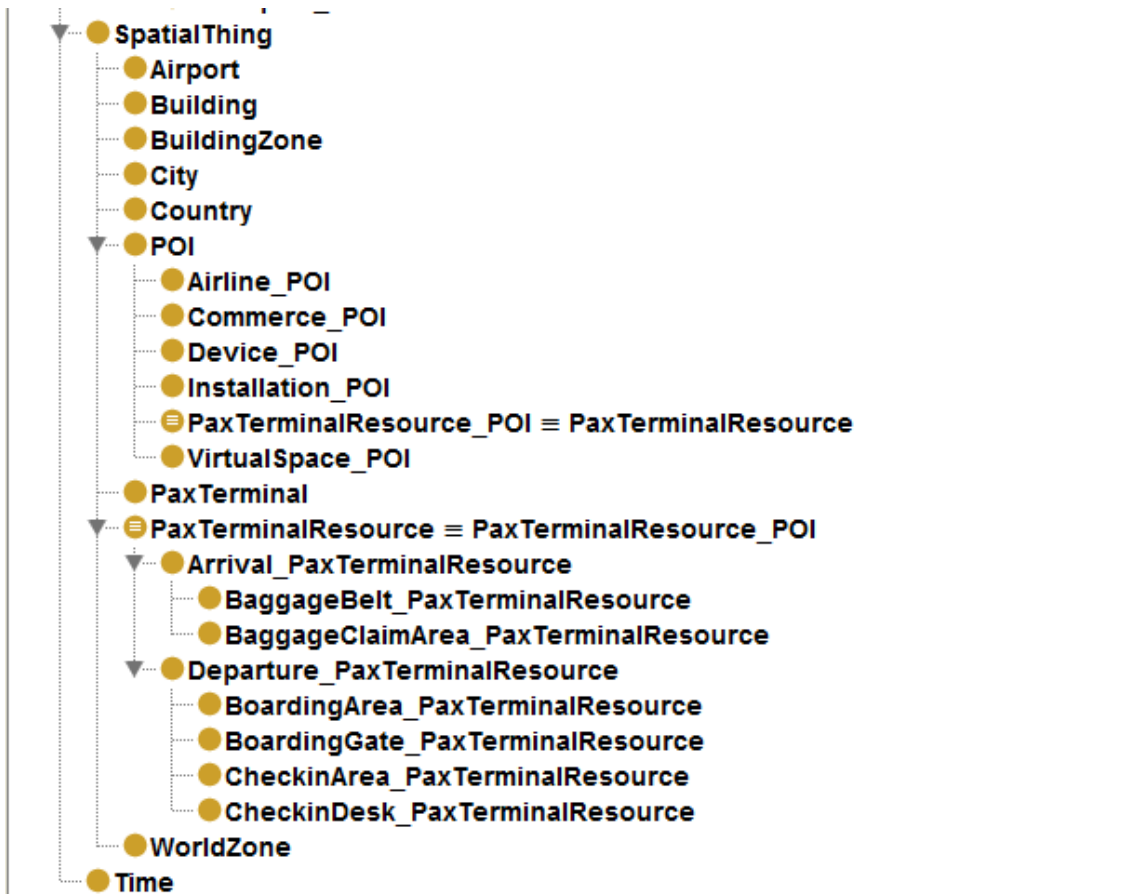


Fig. 4.2. Vista de les classes de *Protégé*



Fig. 4.3. Esquema de la jerarquia de classes obtingut amb GraphViz

A la taula (fig. 4.4) hi ha la correlació entre els noms de les classes implementades i els conceptes definits a la fase de disseny que apareixen al capítol anterior.

Classe implementada	Concepte representat
<i>Aircraft</i>	Avió
<i>Airline</i>	Companyia aèria
<i>FlightOperation</i>	Vol
<i>Arrival_FlightOperation</i>	Vol d'arribada
<i>Departure_FlightOperation</i>	Vol de sortida
<i>FlightOperationCode</i>	Codi de vol
<i>Person</i>	Persona
<i>Passenger</i>	Passatger
<i>Arrival_Passenger</i>	Passatger d'arribada
<i>Departure_Passenger</i>	Passatger de sortida
<i>Transit_Passenger</i>	Passatger en trànsit
<i>Companion</i>	Acompanyant
<i>Visitor</i>	Visitant
<i>PersonalProperty</i>	Propietat personal
<i>PersonalCircumstance</i>	Circumstància personal
<i>PersonalPreference</i>	Preferència personal
<i>ServiceAndProduct</i>	Servei i producte
<i>Product</i>	Producte
<i>Clothes_Product</i>	Roba
<i>Culture_Product</i>	Cultura
<i>Decor_Product</i>	Decoració
<i>Electronic_Product</i>	Electrònica
<i>FoodAndDrink_Product</i>	Menjar i beure
<i>Health_Product</i>	Salut
<i>Other_Product</i>	Altres productes
<i>PerfumsAndCosmetics_Product</i>	Perfums i cosmètics
<i>Tobacco_Product</i>	Tabacs
<i>Travel_Product</i>	Viatges
<i>Service</i>	Servei
<i>AiportProcess_Service</i>	Servei del procés aeroportuari
<i>Complementary_Service</i>	Servei complementari
<i>Customer_Service</i>	Atenció al client
<i>Information_Service</i>	Informació
<i>PaxSupport_Service</i>	Suport al passatger
<i>Transport_Service</i>	Transport
<i>SpatialThing</i>	Concepte amb ubicació física
<i>Airport</i>	Aeroport
<i>Building</i>	Edifici
<i>BuildingZone</i>	Zona d'edifici
<i>City</i>	Ciutat
<i>Country</i>	País
<i>POI</i>	Punt d'interès
<i>Airline_POI</i>	De companyia aèria

<i>Commerce_POI</i>	Comerç
<i>Device_POI</i>	Dispositiu
<i>Installation_POI</i>	Instal·lació
<i>PaxTerminalResource_POI</i>	Recurs del terminal
<i>VirtualSpace_POI</i>	Espai virtual
<i>PaxTerminal</i>	Terminal de passatgers
<i>PaxTerminalResource</i>	Recurs del terminal
<i>Arrival_PaxTerminalResource</i>	Recurs d'arribades
<i>BaggageBelt_PaxTerminalResource</i>	Cinta de recollida d'equipatges
<i>BaggageClaimArea_PaxTerminalResource</i>	Area de recollida d'equipatges
<i>Departure_PaxTerminalResource</i>	Recurs de sortides
<i>BoardingArea_PaxTerminalResource</i>	Area d'embarcament de passatgers
<i>BoardingGate_PaxTerminalResource</i>	Porta d'embarcament
<i>CheckinArea_PaxTerminalResource</i>	Area de facturació d'equipatges
<i>CheckinDesk_PaxTerminalResource</i>	Mostrador de facturació
<i>WorldZone</i>	Zona mundial
<i>Time</i>	Temps

Fig. 4.4. Taula de correspondència entre classes i conceptes

4.4 Implementació de les relacions

Les propietats representen relacions i *Protégé* diferencia entre dos tipus de propietats: les relacions entre una instància de classe i un valor d'un tipus de dada com pot ser un número o una cadena de caràcters, a les que anomena propietats de dades (*Data Properties*), i les relacions entre dues instàncies de classe, les propietats d'objecte (*Object Properties*).

4.4.1 *Object Properties*

Ens centrarem primerament en aquest últim tipus de propietat. En definir una *Object Property* entre dues instàncies de classe és possible enriquir el significat de la relació més enllà de la mera definició que l'individu A *s'hi relaciona amb* B. Això és possible mitjançant les característiques aplicables a les relacions. Una *Object Property* pot ser:

- Funcional: Quan donada una instància, aquesta pot estar relacionada com a màxim amb una altra instància. Per exemple, un vol d'arribada només pot tenir un aeroport d'origen.

- Funcional inversa: En definir una relació de tipus funcional, la relació inversa, si existeix, és del tipus funcional inversa. En el nostre cas, tenim la relació funcional [aeroport] *està en* [ciutat] que relaciona un aeroport amb una ciutat, doncs la relació inversa [ciutat] *té* [aeroport] serà del tipus funcional inversa.
- Transitiva: Quan es dona el cas que si A està relacionat amb B i B està relacionat amb C, es pot inferir que A està també relacionat amb C. En la nostra ontologia, temin la relació transitiva [Passatger] *viatja amb* [Passatger], ja que es compleix que si Joan *viatja amb* Pere i Pere *viatja amb* Maria, es pot inferir que Joan *viatja amb* Maria.
- Simètrica: Quan la relació entre dues instàncies implica la mateixa relació en sentit contrari. La relació *viatja amb* és simètrica, ja que es compleix que si Joan *viatja amb* Pere, s'infereix que Pere *viatja amb* Joan.
- Asimètrica: Aquest tipus de relació significa que si hi existeix una relació entre dues instàncies, no pot existir-hi mai la mateixa relació en sentit contrari. En el cas que ens ocupa, tenim que hi han codis de vol que estan vinculats a un altre que és el principal, per tant si 'IBE1234' *és codi principal de* 'VLG3333', mai es podrà donar la situació 'VLG3333' *és codi principal de* 'IBE1234'.
- Reflexiva: Quan la relació implica que una instància ha d'estar relacionada amb sí mateixa. En l'ontologia aeroportuària no tenim cap cas.
- Irreflexiva: Quan la relació implica que una instància no pot estar relacionada amb sí mateixa. La relació [codi de vol] *és codi principal de* [codi de vol] és irreflexiva, perquè un codi de vol no pot tenir com a codi principal ell mateix.

La creació de les *Object Properties* es realitza amb Protégé de manera molt senzilla i intuïtiva mitjançant la pestanya que porta el mateix nom, tal i com es pot veure a la figura 4.5. Tot i que és possible establir una jerarquia de propietats, en el nostre cas no s'ha

considerat necessari i s'han definit totes les propietats al mateix nivell.

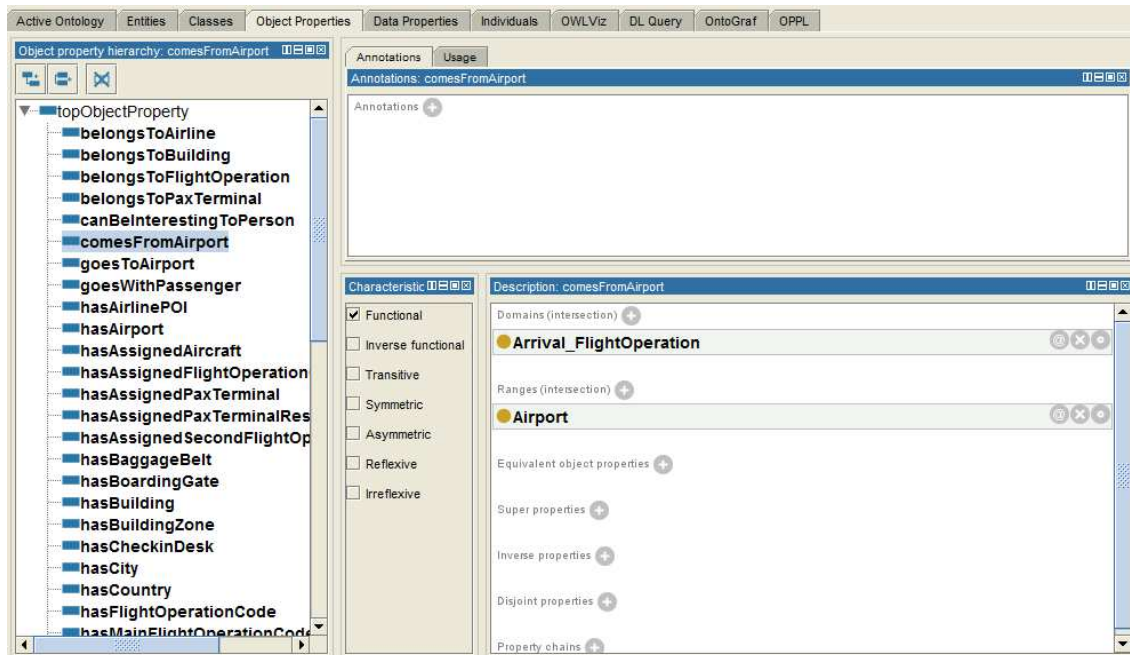


Fig. 4.5. Detall d'una *Object Property* amb *Protégé*

A la figura 4.6 es mostra un quadre resum de totes les *Object Properties* creades amb la informació associada de domini, rang, característiques de la relació i propietat inversa si en té. A continuació, es mostren en un diagrama amb forma de graf dirigit (fig. 4.7) sense incloure-hi les relacions de jerarquia.

Object Property	F	I	T	S	A	R	I	Domini	Rang	Inversa
<i>isOperatedByAirline</i>	X							<i>FlightOperation</i>	<i>Airline</i>	<i>operatesFlightOperation</i>
<i>hasAssignedAircraft</i>	X							<i>FlightOperation</i>	<i>Aircraft</i>	<i>performsFlightOperation</i>
<i>hasAssignedPaxTerminal</i>	X							<i>FlightOperation</i>	<i>PaxTerminal</i>	<i>receivesFlightOperation</i>
<i>hasFlightOperationCode</i>		X						<i>FlightOperation</i>	<i>FlightOperationCode</i>	<i>belongsToFlightOperation</i>
<i>comesFromAirport</i>	X							<i>Arrival_FlightOperation</i>	<i>Airport</i>	
<i>goesToAirport</i>	X							<i>Departure_FlightOperation</i>	<i>Airport</i>	
<i>isAssociatedWithArrivalFlightOperation</i>	X	X						<i>Departure_FlightOperation</i>	<i>Arrival_FlightOperation</i>	<i>isAssociatedWithDepartureFlightOperation</i>
<i>hasMainFlightOperationCode</i>	X				X	X		<i>FlightOperationCode</i>	<i>FlightOperationCode</i>	<i>isMainFlightOperationCodeOf</i>
<i>isMainFlightOperationCodeOf</i>		X			X	X		<i>FlightOperationCode</i>	<i>FlightOperationCode</i>	<i>hasMainFlightOperationCode</i>
<i>belongsToAirline</i>	X							<i>FlightOperationCode</i>	<i>Airline</i>	<i>operatesFlightOperationCode</i>
<i>hasAssignedPaxTerminalResource</i>								<i>FlightOperationCode</i>	<i>PaxTerminalResource</i>	<i>isAssignedToFlightOperationCode</i>
<i>hasPaxTerminal</i>		X						<i>Airport</i>	<i>PaxTerminal</i>	<i>isInAirport</i>
<i>isInCity</i>	X							<i>Airport</i>	<i>City</i>	<i>hasAirport</i>
<i>isInCountry</i>	X							<i>City</i>	<i>Country</i>	<i>hasCity</i>
<i>isInWorldZone</i>	X							<i>Country</i>	<i>WorldZone</i>	<i>hasCountry</i>
<i>hasPaxTerminalResource</i>		X						<i>PaxTerminal</i>	<i>PaxTerminalResource</i>	<i>isInPaxTerminal</i>
<i>hasBuilding</i>		X						<i>PaxTerminal</i>	<i>Building</i>	<i>belongsToPaxTerminal</i>
<i>hasBuildingZone</i>		X						<i>Building</i>	<i>BuildingZone</i>	<i>belongsToBuilding</i>
<i>isCloseToBuildingZone</i>				X				<i>BuildingZone</i>	<i>BuildingZone</i>	
<i>hasBoardingGate</i>		X						<i>BoardingArea_PaxTerminalResource</i>	<i>BoardingGate_PaxTerminalResource</i>	<i>isInBoardingArea</i>
<i>hasCheckinDesk</i>		X						<i>CheckinArea_PaxTerminalResource</i>	<i>CheckinDesk_PaxTerminalResource</i>	<i>isInCheckinArea</i>

hasBaggageBelt		X					<i>BaggageClaimArea_PaxTerminalResource</i>	<i>BaggageBelt_PaxTerminalResource</i>	<i>isInBaggageClaimArea</i>
isOfferedAtPOI							<i>ServiceAndProduct</i>	<i>POI</i>	<i>offersSAP</i>
isLocatedAtBuildingZone	X						<i>POI</i>	<i>BuildingZone</i>	<i>hasPOI</i>
isRelatedToAirline							<i>Airline_POI</i>	<i>Airline</i>	<i>hasAirlinePOI</i>
maybeInterestedInSAP							<i>Person</i>	<i>ServiceAndProduct</i>	<i>canBeInterestingToPerson</i>
hasPersonalProperty							<i>Person</i>	<i>PersonalProperty</i>	
isFromCountry	X						<i>Person</i>	<i>Country</i>	
goesWithPassenger							<i>Companion</i>	<i>Passenger</i>	<i>isAccompaniedBy</i>
travelsWithPassenger			X	X			<i>Passenger</i>	<i>Passenger</i>	
hasAssignedFlightOperationCode	X						<i>Passenger</i>	<i>FlightOperationCode</i>	
hasAssignedSecondFlightOperationCode	X						<i>Transit_Passenger</i>	<i>FlightOperationCode</i>	
operatesFlightOperation		X					<i>Airline</i>	<i>FlightOperation</i>	<i>isOperatedByAirline</i>
performsFlightOperation		X					<i>Aircraft</i>	<i>FlightOperation</i>	<i>hasAssignedAircraft</i>
receivesFlightOperation		X					<i>PaxTerminal</i>	<i>FlightOperation</i>	<i>hasAssignedPaxTerminal</i>
belongsToFlightOperation	X						<i>FlightOperationCode</i>	<i>FlightOperation</i>	<i>hasFlightOperationCode</i>
isAssociatedWithDepartureFlightOperation	X	X					<i>Arrival_FlightOperation</i>	<i>Departure_FlightOperation</i>	<i>isAssociatedWithArrivalFlightOperation</i>
operatesFlightOperationCode		X					<i>Airline</i>	<i>FlightOperationCode</i>	<i>belongsToAirline</i>
isAssignedToFlightOperationCode							<i>PaxTerminalResource</i>	<i>FlightOperationCode</i>	<i>hasAssignedPaxTerminalResource</i>
isInAirport	X						<i>PaxTerminal</i>	<i>Airport</i>	<i>hasPaxTerminal</i>
hasAirport		X					<i>City</i>	<i>Airport</i>	<i>isInCity</i>
hasCity		X					<i>Country</i>	<i>City</i>	<i>isInCountry</i>
hasCountry		X					<i>WorldZone</i>	<i>Country</i>	<i>isInWorldZone</i>

<i>isInPaxTerminal</i>	X						<i>PaxTerminalResource</i>	<i>PaxTerminal</i>	<i>hasPaxTerminalResource</i>
<i>belongsToPaxTerminal</i>	X						<i>Building</i>	<i>PaxTerminal</i>	<i>hasBuilding</i>
<i>belongsToBuilding</i>	X						<i>BuildingZone</i>	<i>Building</i>	<i>hasBuildingZone</i>
<i>isInBoardingArea</i>	X						<i>BoardingGate_PaxTerminalResource</i>	<i>BoardingArea_PaxTerminalResource</i>	<i>hasBoardingGate</i>
<i>isInCheckinArea</i>	X						<i>CheckinDesk_PaxTerminalResource</i>	<i>CheckinArea_PaxTerminalResource</i>	<i>hasCheckinDesk</i>
<i>isInBaggageClaimArea</i>	X						<i>BaggageBelt_PaxTerminalResource</i>	<i>BaggageClaimArea_PaxTerminalResource</i>	<i>hasBaggageBelt</i>
<i>offersSAP</i>							<i>POI</i>	<i>ServiceAndProduct</i>	<i>isOfferedAtPOI</i>
<i>hasPOI</i>		X					<i>BuildingZone</i>	<i>POI</i>	<i>isLocatedAtBuildingZone</i>
<i>hasAirlinePOI</i>							<i>Airline</i>	<i>Airline_POI</i>	<i>isRelatedToAirline</i>
<i>canBeInterestingToPerson</i>							<i>ServiceAndProduct</i>	<i>Person</i>	<i>maybeInterestedInSAP</i>
<i>isAccompaniedBy</i>							<i>Passenger</i>	<i>Companion</i>	<i>goesWithPassenger</i>

Fig. 4.6. Quadre resum de les *Object Properties* implementades

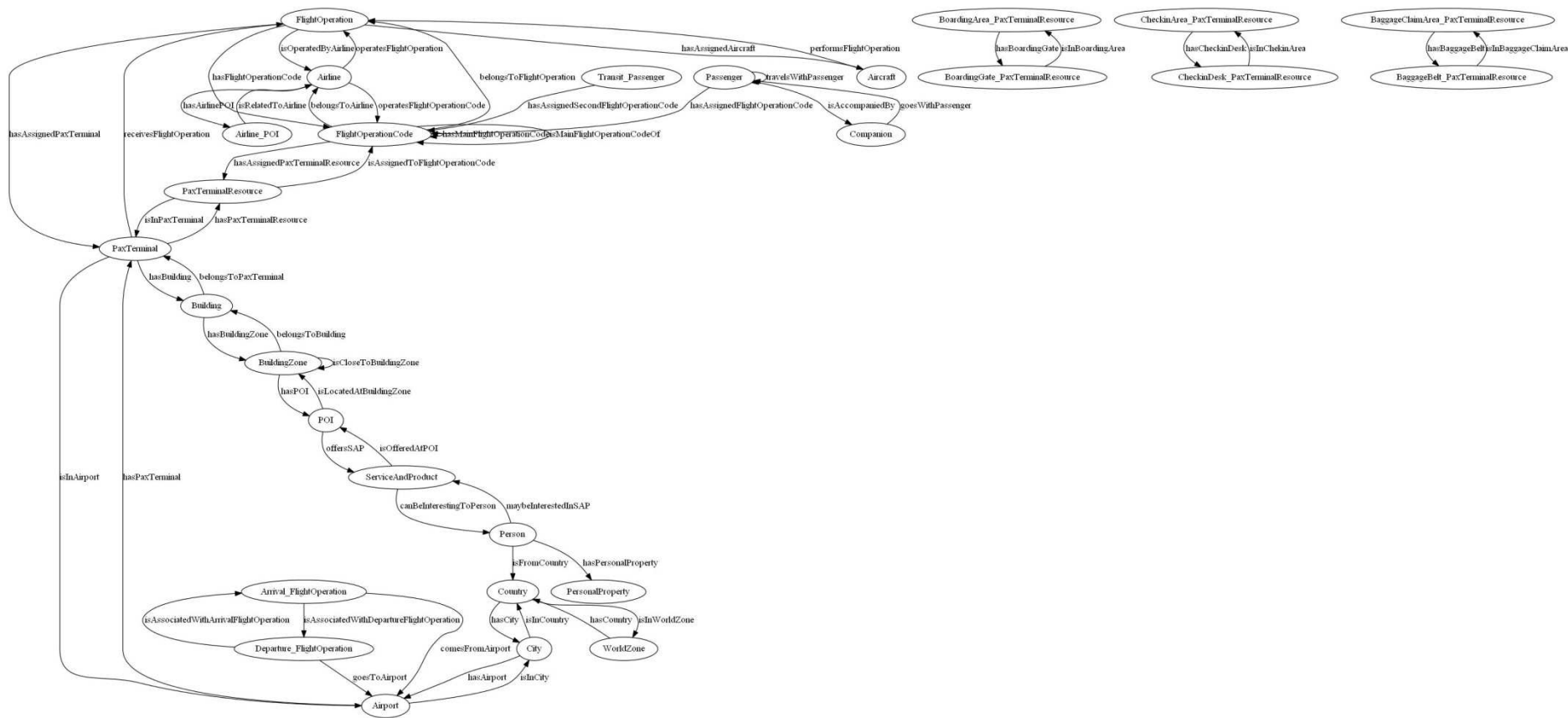


Fig. 4.7. Propietats entre instàncies de classes

4.4.2 Data Properties

En aquest apartat es mostra com s'ha dut a terme la creació de les *Data Properties*, les quals estableixen una relació entre una instància de classe i un valor d'un tipus de dades com ara un número o una cadena de caràcters. En la creació, a més de ser necessari informar el tipus de dades que emmagatzemarà, es pot indicar si aquesta propietat serà o no funcional. Que una *Data property* sigui del tipus funcional vol dir que les instàncies tindran com a màxim un únic valor per aquesta propietat. En el nostre cas, totes les *Data properties* definides són del tipus funcional.

En la creació mitjançant *Protégé* indiquem el nom de la propietat, el domini (classe en què podrem utilitzar-la) i el rang, que en aquest cas és un tipus de dades o un conjunt de literals. A les figures 4.8 i 4.9 s'il·lustren alguns exemples.

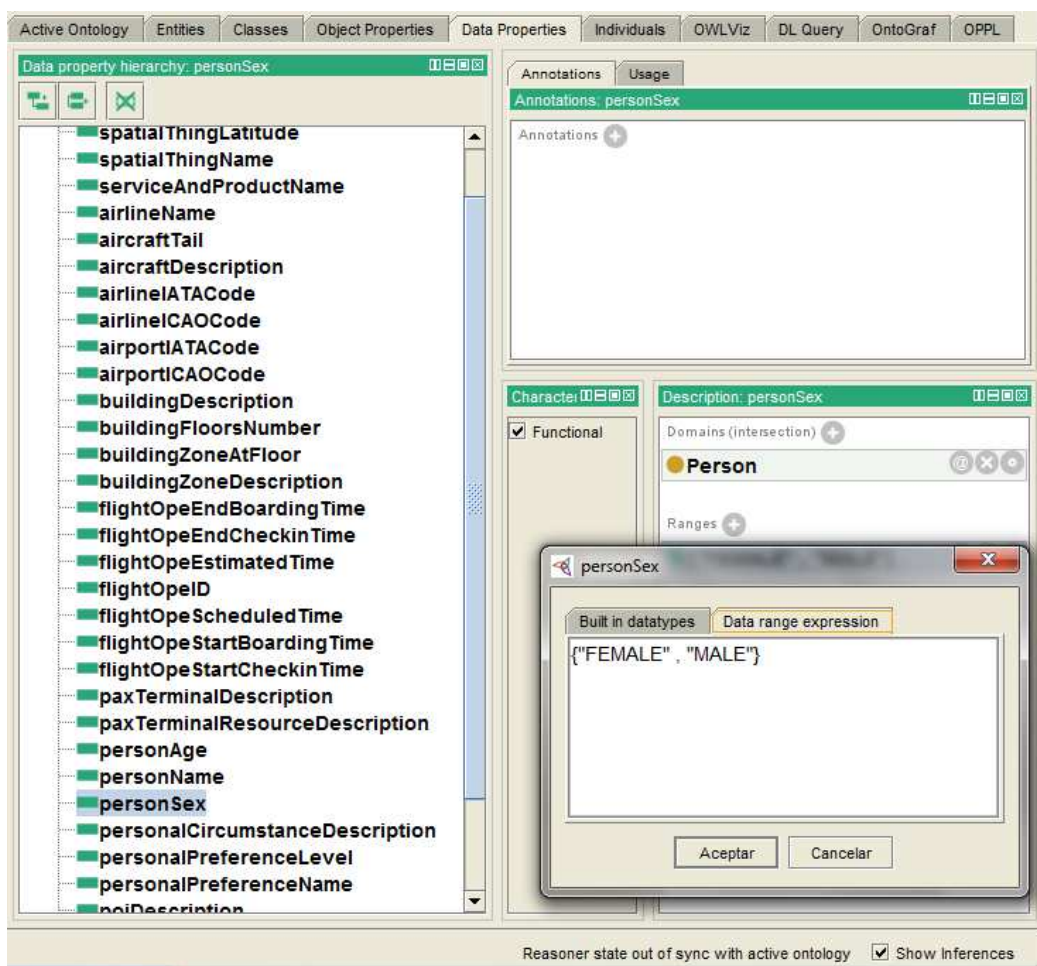


Fig. 4.8. Creació d'una *Data Property* amb rang definit per un conjunt de literals

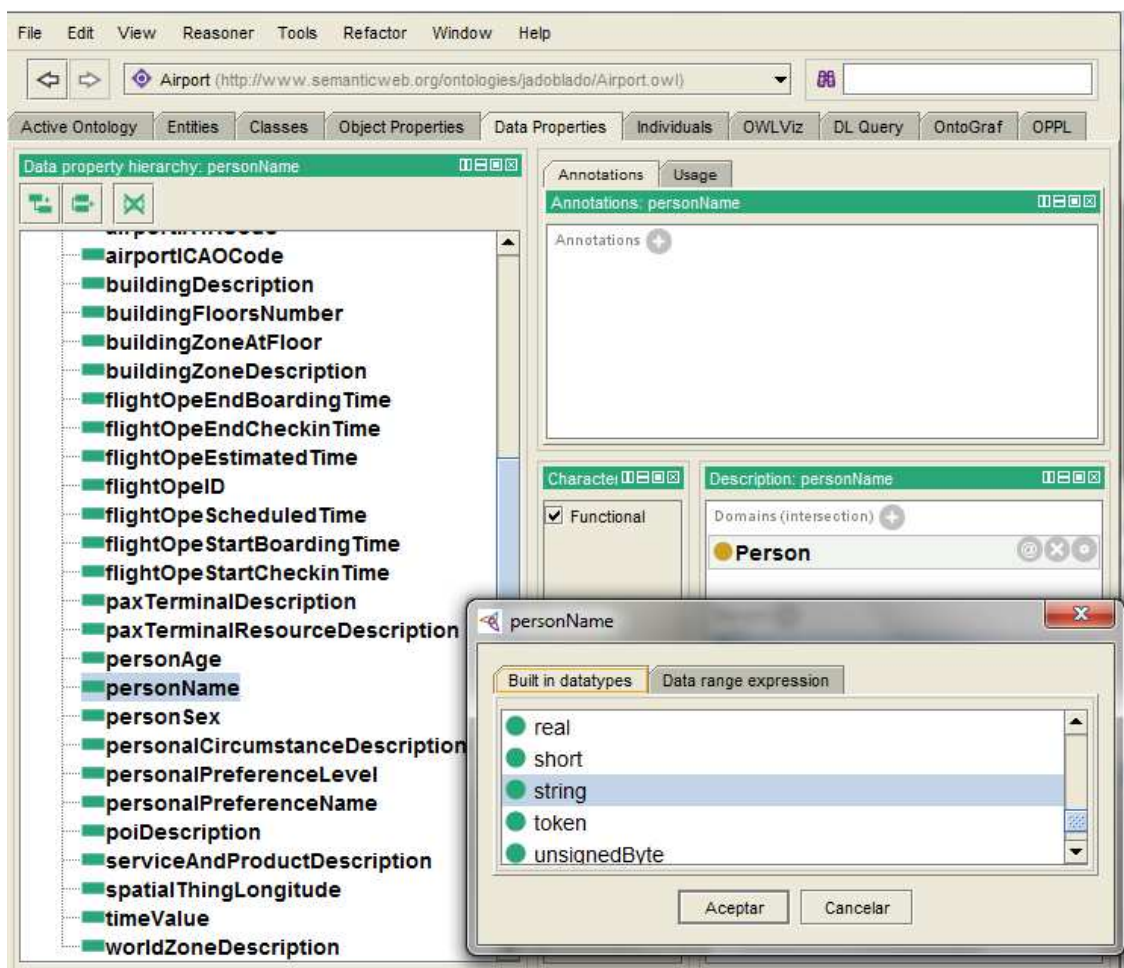


Fig. 4.9. Creació d'una *Data Property* amb rang de tipus cadena de caràcters

El quadre (fig. 4.10) resum totes les *Data Properties* creades amb els seus dominis i rangs.

<i>Data Property</i>	Domini	Rang
<i>aircraftDescription</i>	<i>Aircraft</i>	<i>String</i>
<i>aircraftTail</i>	<i>Aircraft</i>	<i>string</i>
<i>airlineIATACode</i>	<i>Airline</i>	<i>string</i>
<i>airlineICAOCode</i>	<i>Airline</i>	<i>string</i>
<i>airlineName</i>	<i>Airline</i>	<i>string</i>
<i>airportIATACode</i>	<i>Airport</i>	<i>string</i>
<i>airportICAOCode</i>	<i>Airport</i>	<i>string</i>
<i>buildingFloorsNumber</i>	<i>Building</i>	<i>integer</i>
<i>buildingZoneAtFloor</i>	<i>BuildingZone</i>	<i>integer</i>
<i>flightOpeEndBoardingTime</i>	<i>Departure_FlightOperation</i>	<i>dateTime</i>

flightOpeEndCheckinTime	Departure_FlightOperation	dateTime
flightOpeStartBoardingTime	Departure_FlightOperation	dateTime
flightOpeStartCheckinTime	Departure_FlightOperation	dateTime
flightOpeEstimatedTime	FlightOperation	dateTime
flightOpeID	FlightOperation	string
flightOpeScheduledTime	FlightOperation	dateTime
flightOperationCodeID	FlightOperationCode	string
personAge	Person	integer
personName	Person	string
personSex	Person	{"MALE","FEMALE"}
personalPreferenceLevel	PersonalPreference	{"LOW","MEDIUM","HIGH"}
personalPropertyDescription	PersonalProperty	string
personalPropertyName	PersonalProperty	string
serviceAndProductDescription	ServiceAndProduct	string
serviceAndProductName	ServiceAndProduct	string
spatialThingDescription	SpatialThing	string
spatialThingLatitude	SpatialThing	double
spatialThingLongitude	SpatialThing	double
spatialThingName	SpatialThing	string
timeValue	Time	dateTime

Fig. 4.10. Quadre resum de les *Data Properties* implementades

4.4.3 Restriccions

En aquest moment tenim creades les classes i les propietats, però en l'etapa de disseny havíem definit un altre concepte: la cardinalitat de les propietats. La implementació d'aquestes restriccions en Protégé, es fa mitjançant la definició de classes equivalents amb modificadors de cardinalitat. Veurem un exemple:

Domini	Propietat	Rang	Cardinalitat (mín., màx.)	Propietat inversa
Vol d'arribada	Prové d'un aeroport origen	Aeroport	(1,1)	

Per garantir que un vol d'arribada té necessàriament un i només un aeroport d'origen, tal com vam dir a la definició del model conceptual,

en la implementació amb *Protégé* es representa com es veu a la figura 4.11:

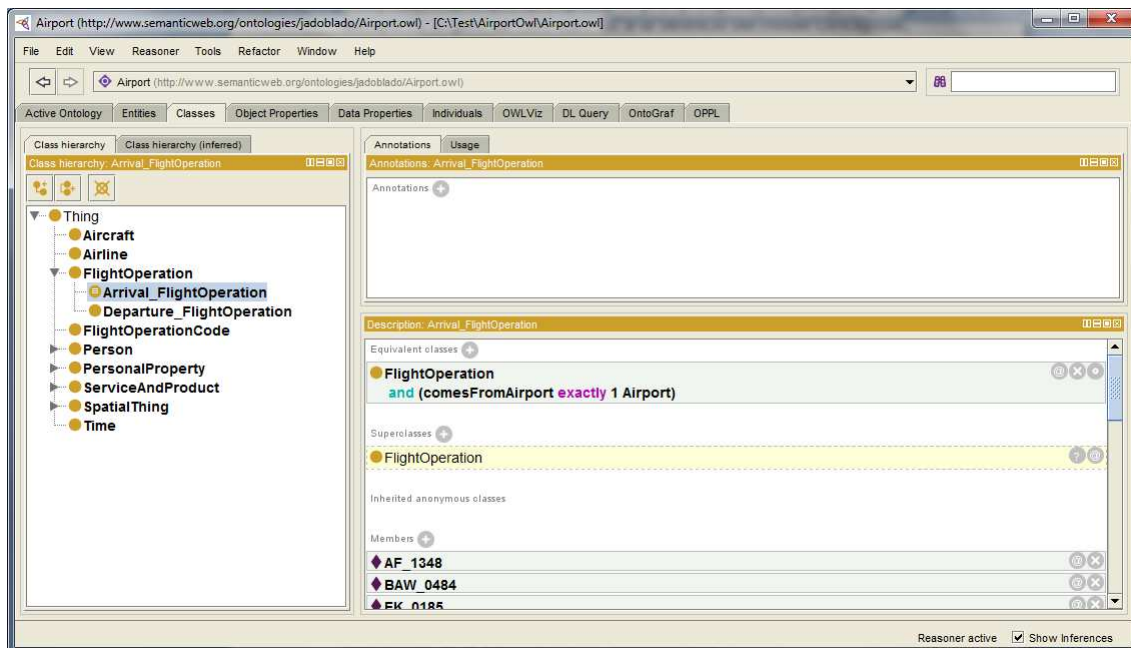


Fig. 4.11. Implementació de restriccions de cardinalitat amb Protégé

4.5 Integració de la informació

Un cop implementades les classes i relacions de l'ontologia, ja només resta crear les instàncies. En el cas que ens ocupa, i atenent a la freqüència amb què la informació haurà d'actualitzar-s'hi, distingirem entre dos tipus d'informació a integrar:

Informació de la infraestructura aeroportuària

Com a informació aeroportuària s'entén tot allò que forma part de la infraestructura d'un aeroport, com ara els edificis terminals, les oficines de venda de bitllets, els mostradors de facturació d'equipatges, les portes d'embarcament, etc. Aquesta informació no anirà canviant en el temps amb molta freqüència, per tant la considerarem de tipus estàtic i només caldrà actualitzar-la quan s'hi introdueixin canvis en les instal·lacions.

Per aquest motiu he considerant apropiat definir un mecanisme semiautomàtic per a la integració d'aquestes dades. Aquest mecanisme consistirà en un programa fet en *Java*, utilitzant les llibreries de *Jena Ontology API* que integrarà la informació en l'ontologia, a partir de dades emmagatzemades en arxius tipus CSV. He triat el format CSV (*Comma Separated Values*) per tal de facilitar l'exportació de dades des de qualsevol font d'informació origen.

El programa s'anomena *popOnt* (abreviatura de *populate Ontology*), i s'executa des de la consola. Admet com a paràmetres el nom de la classe a instanciar i el nom del fitxer CSV que conté les dades. La primera línia del fitxer ha d'estar formada per una capçalera amb els noms de les propietats i el tipus si es tracta de *Data Properties* a les que corresponen els valors que hi segueixen. La sintaxi admesa pel programa és:

```
popOnt <Ontology_OWLFile>,<Ontology_ClassName>,<CSV_DataFile>[,<Separator_Character>]
```

Si volguéssim poblar la classe *Airline* amb el fitxer *airlines.csv* que conté els valors separats pel caràcter ";" i l'ontologia està emmagatzemada a l'arxiu *ElPrat.owl*, es cridaria al programa de la forma:

```
popOnt ElPrat,Airline,airlines,;
```

L'arxiu *airlines.csv* tindria la forma:

```
Id;airlineName:string;airlineOaciCode:string;airlineIataCode:string  
AmericanAirlines;American Airlines;AL;AAL  
Vueling;Vueling;VY;VLG  
Iberia;Iberia;IBE;IB
```

Val a dir que s'ha optat per aquesta opció, després d'haver valorat diferents solucions existents i que es poden trobar per la xarxa com per exemple l'eina *Populous* de la companyia *e-LICO* o el programa *GATE* desenvolupat per la *Universitat de Sheffield* disponible amb llicència *GNU* però no s'hi terminaven d'adaptar, a més que la seva utilització era bastant difícil.

Informació de vols i assignació de medis

Consisteix en les dades relatives a vols d'arribada i de sortida així com els medis aeroportuaris assignats. Per simplificar, partim de la situació que aquesta informació la rebrem periòdicament dels sistemes aeroportuaris, mitjançant un arxiu de tipus CSV com en el cas de la informació aeroportuària.

Una opció per resoldre la integració de la informació dinàmica és mantenir dues versions de l'ontologia. Una versió auxiliar que podríem anomenar *base* amb tota la informació excepte la relativa a vols, i la versió *final* amb aquesta darrera informació incorporada, la qual s'hauria d'anar generant periòdicament, a mesura que s'actualitza la informació de vols. Encara que la millor manera seria

implementar un mecanisme d'integració basat en missatges de canvis des dels sistemes aeroportuaris cap a l'ontologia, però aquest desenvolupament queda fora de l'àmbit del projecte.

Cal dir que per aquesta integració s'ha fet servir també el programari *OPPL API* esmentat anteriorment.

4.5.1 Conjunt de dades

Per poder comprovar la consistència de l'ontologia i la realització de consultes era necessari definir un conjunt de dades a integrar-hi com a instàncies de les classes amb les seves propietats. Per a la formació d'aquest conjunt i pel que fa la part d'infraestructura aeroportuària, s'ha pres com a exemple l'aeroport de Barcelona-El Prat, del qual, per aspectes de manejabilitat, no s'ha considerat la totalitat dels elements públics que el formen i poden consultar-se a la web [1], sinó només uns quants de representatius.

En allò que respecta a la informació de vols, i amb l'objectiu de simular una situació real, s'ha elaborat una mostra fictícia amb les dades de codis de vol, orígens, destinacions, medis assignats, etc, els quals podrien estar vigents en un moment donat.

4.6 Consultes

Un cop poblada l'ontologia i comprovat que no hi existeixen inconsistències és l'hora d'extreure el coneixement representat i comprovar que l'ontologia dóna resposta a les qüestions que havíem plantejat en el començament del disseny.

Aquesta tasca es duu a terme utilitzant el llenguatge de consultes *SPARQL*. La funcionalitat està disponible a *Protégé* 4.2 a través de la pestanya anomenada "*SPARQL Query*" (fig. 4.12).

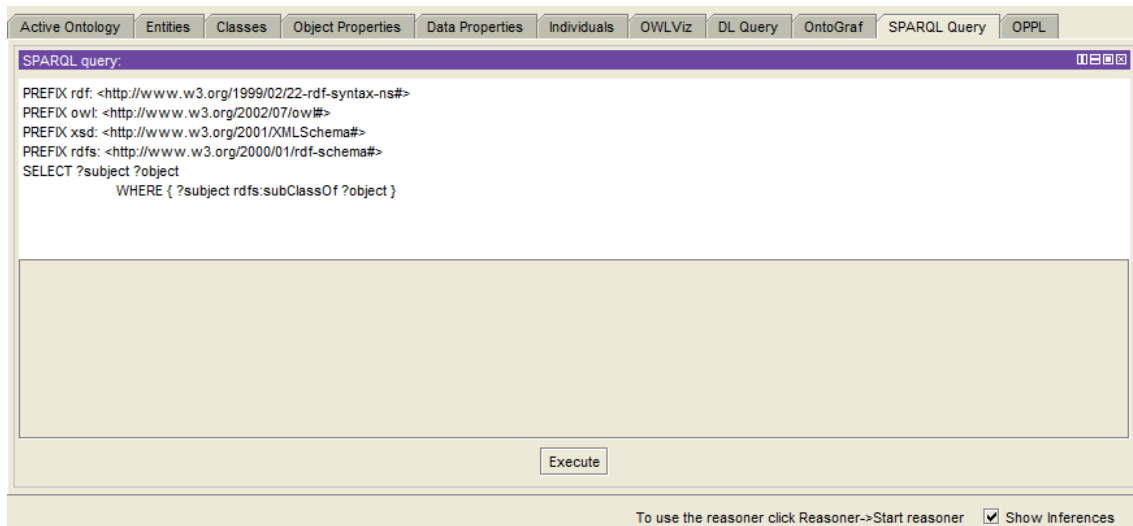


Fig. 4.12. Pestanya SPARQL Query en Protégé 4.2

A continuació s'hi inclouen una sèrie de consultes i els resultats obtinguts (fig. 4.13 a 4.15):

- Consulta sobre els comerços on es poden trobar productes de moda tant per a home com per a dona

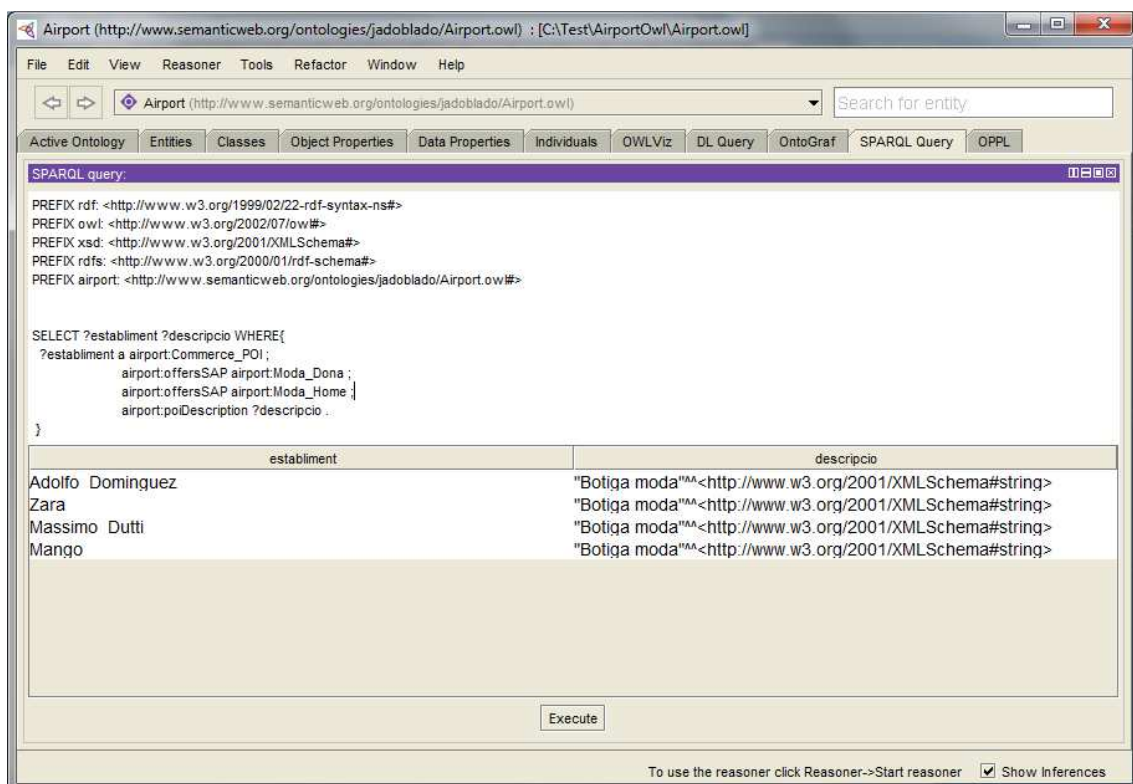


Fig. 4.13. Resultat de consulta SPARQL

- Establiments on es pot prendre cafè a la zona anomenada "Sky Center"

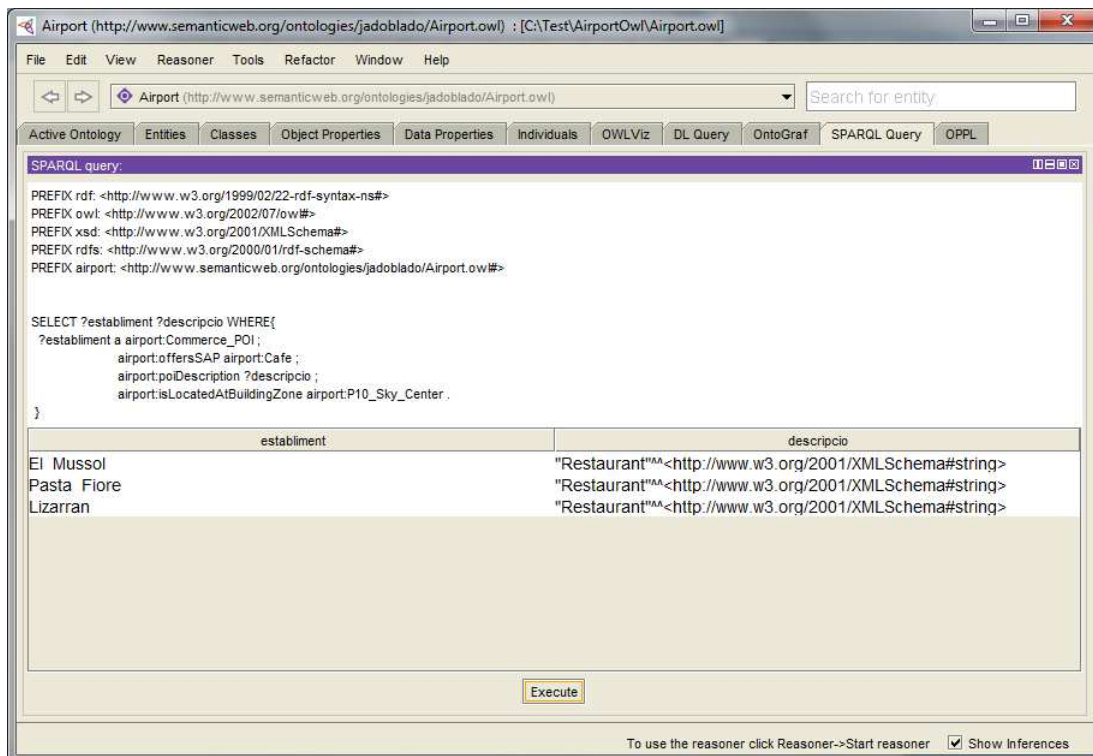


Fig. 4.14. Resultat de consulta SPARQL

- Porta assignada i hora d'inici de l'embarcament pel vol LY-394

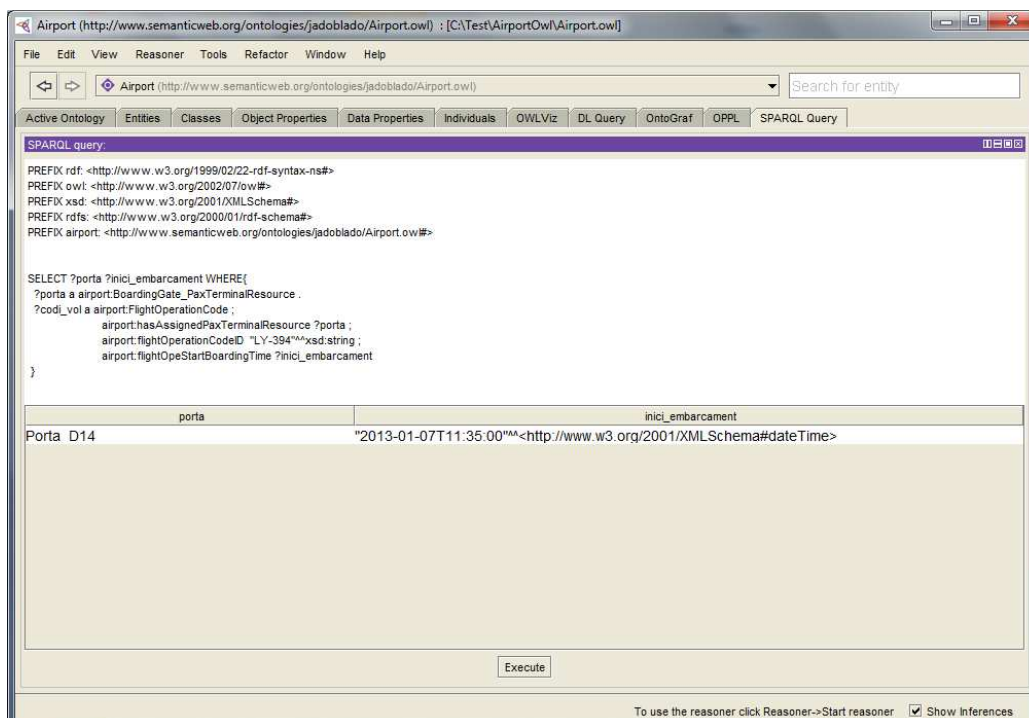


Fig. 4.15. Resultat de consulta SPARQL

5. Conclusions

Un cop finalitzat el treball, les conclusions que es poden extreure-hi són:

Des del punt de vista històric de l'origen de les ontologies, podria semblar que mons tan distants en el temps i en els seus fonaments, com són la filosofia grega i les TIC, mai arribarien a creuar-s'hi. Amb l'ús de les ontologies en la intel·ligència artificial aplicada, queda patent però, que és possible utilitzar les eines que ens proporciona la tecnologia per desenvolupar teories de la metafísica.

Des de la vessant tecnològica, cal destacar els esforços de la W3C en l'elaboració d'especificacions que han esdevingut en estàndards per a la representació de la informació, que han estat clau per a la proliferació d'eines i aplicacions en el camp de la Web semàntica.

Des de la perspectiva de l'abordament del problema plantejat en aquest treball, es pot afirmar que les tècniques de representació del coneixement basades en ontologies són perfectament vàlides per oferir-hi solucions.

Bibliografia

1. AENA AEROPUERTOS. [en línia] <http://www.aena-aeropuertos.es/cartografia/recinto.jsp?poi=&t=&p=&id=7&fplano1237548330376> [data de consulta 12/2012].
2. BERNERS-LEE, T; HENDLER, J; LASSILA, O. (2001). *The Semantic Web*. Scientific American (Maig 2001).
3. EGAÑA, M.; IANNONE, L.; STEVENS, R. (2011). *Ontology Pre-Processor Language (OPPL) User's manual*. School of Computer Science. Technical University of Madrid (UPM).
4. E-LICO. [en línia] <http://www.e-lico.eu/populous.html>
5. GENERAL ARCHITECTURE FOR TEXT ENGINEERING. [en línia] <http://gate.ac.uk/>
6. GOMEZ-PEREZ, A. (1998). *Knowledge sharing and reuse*. Handbook of Expert Systems, CRC (cap. 10). New York.
7. GRAPH VISUALIZATION SOFTWARE. [en línia] <http://www.graphviz.org/>
8. GRUBER, T. R. (1993). *A translation approach to portable ontology specifications*. Knowledge Acquisition (Vol. 5:199-220).
9. GUARINO, N. (1995). *Formal ontology, conceptual analysis and knowledge representation*. International Journal of Human and Computer Studies.
10. HORRIDGE, M. (2011). *A practical guide to building OWL ontologies using Protege 4 and CO-ODE tools*. University of Manchester.
11. KANT, I. (2001) *Lectures on metaphysics - Part III Metaphysik L2*. Cambridge University Press.
12. MIZOGUCHI, R., VANWELKENHUYSEN, J., IKEDA, M. (1995). *Task Ontology for Reuse of Problem Solving Knowledge. Towards Very Large Knowledge Bases*. KnowledgeBuilding and Knowledge Sharing (vol. 46-59)

13. NOY, N.; MCGUINNESS, D. (2005). *Ontology Development 101: A Guide to Creating Your First Ontology*. Stanford University.
14. THE PLACE OF ONTOLOGY IN MODERN PHILOSOPHY. [en línia] <http://ww.formalongology.it> [data de consulta 12/2012].
15. SITA (2012). *Passenger Self-Service Survey 2012*.
16. [STANFORD CENTER FOR BIOMEDICAL INFORMATICS RESEARCH](http://protege.stanford.edu) [en línia] <http://protege.stanford.edu>
17. THE APACHE SOFTWARE FOUNDATION. [en línia] <http://jena.apache.org>
18. THE DARPA AGENT MARKUP LANGUAGE. [en línia] <http://www.daml.org/2001/02/geofile/geofile-ont.daml>
19. THE FRIEND OF A FRIEND PROJECT. [en línia] <http://xmlns.com/foaf/spec/>
20. THE WORLD WIDE WEB CONSORTIUM. [en línia] <http://www.w3.org>
21. VAN HEIJST, G., SCHREIBER, A. T., WIELINGA, B. J. (1997). *Using explicit ontologies in KBS development*. International Journal of Human-Computer Studies (núm. 45).

Annex I. Codi font de *popOnt*

popOnt.java

```
package popont;

import java.io.File;
import java.io.IOException;
import java.io.PrintWriter;
import com.csvreader.CsvReader;
import com.hp.hpl.jena.datatypes.xsd.XSDDatatype;
import com.hp.hpl.jena.ontology.Individual;
import com.hp.hpl.jena.ontology.OntClass;
import com.hp.hpl.jena.ontology.OntModel;
import com.hp.hpl.jena.ontology.OntModelSpec;
import com.hp.hpl.jena.rdf.model.ModelFactory;
import com.hp.hpl.jena.shared.JenaException;

public class PopOnt extends Object {

    static String ontology_OWLFile, ontology_ClassName, csv_DataFile;
    static Character separator_char=',';

    public static void main (String args[]) throws IOException {

        if (args.length < 3 || args.length >4) {
            System.out.println("Usage: popOnt
<Ontology_OWLFile>,<Ontology_ClassName>,<CSV_DataFile>[,<Separator_Character>");
            System.exit(0);
        }
        ontology_OWLFile = args[0];
        ontology_ClassName = args[1];
        csv_DataFile = args[2];
        if (args.length == 4) separator_char = args[3].charAt(0);

        populate();

    }

    public static void populate() throws IOException{

        OntModel model = ModelFactory.createOntologyModel(OntModelSpec.OWL_MEM);
        String ns = "http://www.semanticweb.org/ontologies/jadoblado/Airport.owl#";

        try{
            model.read("file:"+ontology_OWLFile+".owl","RDF/XML");
        } catch (JenaException je) {
            System.out.println("ERROR");
            je.printStackTrace();
            System.exit(0);
        }

        //Llegir l'arxiu .csv
        CsvReader reader = null;
        try {
            reader = new CsvReader(csv_DataFile+".csv",separator_char);

            reader.readHeaders();
            int numCols = reader.getHeaderCount();
            String headers[]=reader.getHeaders();
            String headerType[]=new String[numCols];
            String headerName[]=new String[numCols];
            XSDDatatype type[]=new XSDDatatype[numCols];
            for (int i=1;i<numCols;i++){
                int pos=headers[i].indexOf(':');
                if (pos>=0){
                    headerType[i]=headers[i].substring(pos+1);
                    headerName[i]=headers[i].substring(0,pos);

                    type[i]=XSDDatatype.XSDstring; //default
                    if (headerType[i].equals("string"))
```

```

        type[i]=XSDDatatype.XSDstring;
    else if (headerType[i].equals("date"))
        type[i]=XSDDatatype.XSDdateTime;
    else if (headerType[i].equals("number"))
        type[i]=XSDDatatype.XSDinteger;

    headers[i]=headerName[i];
}
}
OntClass c1 = model.getOntClass( ns + ontology_ClassName );
Individual ind;
while (reader.readRecord()) {
    ind = model.createIndividual(ns+reader.get(0), c1);

    for (int i=1;i<numCols;i++){

        if (model.getObjectProperty(ns+headers[i]) != null){
            ind.setPropertyValue(model.getProperty(ns+headers[i]),

model.getResource(ns+reader.get(i)));
        }
        else
            ind.setPropertyValue(model.getProperty(ns+headers[i]),

model.createTypedLiteral(reader.get(i),type[i]));
    }
}

} catch (Exception ex) {
    ex.printStackTrace();
} finally {
    reader.close();
} // end try

//Guardar l'ontologia
File file = new File(ontology_OWLFile+".owl");
try{
    if (!file.exists()){
        file.createNewFile();
    }
    model.writeAll(new PrintWriter(file),null,null);
} catch(IOException ioe){
    ioe.printStackTrace();
}
}
}
}

```