UOC

ONTOLOGIES TURÍSTIQUES GEOGRÀFIQUES

PLA DE TREBALL

DANIEL BOSCH VEGA
Treball Fi de Carrera

Enginyeria Informàtica

CURS 2010-2011 – Segon semestre

Als meus pares

Índex

ÍNDEX **3**

INTRODUCCIÓ **6**

OBJECTIUS **7**

OBJECTIUS DEL PROJECTE 7
ABAST DEL PROJECTE 8
RESULTATS ESPERATS 8

FONAMENTS **8**

QUÈ ÉS UNA ONTOLOGIA 8
ONTOLOGIA A L'ÀMBIT DE LA INFORMÀTICA 9
COMPONENTS D'UNA ONTOLOGIA 10
PROCÉS DE CREACIÓ D'UNA ONTOLOGIA 11
TIPUS D'ONTOLOGIES 13

LLENGUATGES EMPLEATS PER A DEFINIR ONTOLOGIES **14**

XML 14
Origen *15*
Característiques del XML *15*
Estructura d'un document en XML *16*
Documents ben formats i vàlids *19*
Documents vàlids *20*
Presentació XML *23*
Processament d'un XML *24*
Limitacions del XML amb les metadades *25*
RDF 25
Orígens *25*
Característiques del RDF *26*
El model RDF *27*
Sintaxi RDF i RDF/XML *28*
Esquema RDF *33*
OWL 36
Origen i característiques *36*
Els tres subllenguatges de OWL *36*
Estructura del OWL *38*

LLENGUATGES D'INTERROGACIÓ D'ONTOLOGIES **40**

SPARQL 40

EINES EMPLEADES PER A DEFINIR ONTOLOGIES **41**

PROTÉGÉ 42
RACERPRO 43
OWLVIZ I GRAPHVIZ 43
ONTOVIZ 43

ESTAT DE LES ARTS A LES ONTOLOGIES **44**

TESAURUS I ALTRES ESPECIFICACIONS 44

Thesaurus on Tourism and Leisure Activities (UNWTO) *44*
La OpenTravel Alliance (OTA) *45*
MONDECA TOURISM ONTOLOGY 45
HIGH-TOUCH 46
ONTOLOGY HARMONISE 46
QALL-ME 47
DERI ETOURISM ONTOLOGY 48
EON TRAVELLING ONTOLOGY 49
CDOTT ONTOLOGY 49
TAGA 49
GETESS ONTOLOGY 50
ONTOLOGIA CRUZAR 50
ONTOLOGIA ANOTA 50

CREACIÓ DE L'ONTOLOGIA 51

DESCRIPCIÓ 51
PROCÉS DE CREACIÓ 51
ABAST 52
REUTILITZACIÓ D'ONTOLOGIES EXISTENTS 52
ENUMERACIÓ DELS TERMES IMPORTANTS 53
CREACIÓ DE LES CLASSES 53
Nom de les classes *55*
Descripció de les classes *55*
Jerarquia de classes *58*
Creació dels slots *59*
Tipus de valor dels slots *59*
Noms dels slots *60*
Domini i rang d'un slot *60*
Cardinalitat dels valors dels slots *60*
RESUM DE CLASSES I SLOTS DE L'ONTOLOGIA 61
Classe: DadesContacte *61*
Classe:Document *61*
Classe:DocumentAudio *62*
Classe:DocumentVideo *63*
Classe:DocumentText *63*
Classe:DocumentFotografic *64*
Classe:DocumentWeb *64*
Classe:Entrada *64*
Classe:HorariObertura *65*
Classe:Idioma *66*
Classe:Localitzacio *66*
Classe:LocalitzacioCoordenadesGPS *66*
Classe:LocalitzacioDireccioPostal *67*
Classe:Periode *67*
Classe:PeriodeDates *68*
Classe:PeriodeHores *68*
Classe:Temporada *68*
Classe:PuntInteres *69*

Classe:PuntComercial *69*
Classe: ComercialCentreComercial *70*
Classe: ComercialComerç *70*
Classe: ComercialSupermercat *70*
Classe: ComercialHipermercat *71*
Classe: ComercialMercat *71*
Classe:PuntHostalaria *71*
Classe:HostalariaAllotjament *72*
Classe:HostalariaRestauracio *72*
Classe:PuntOci *72*
Classe: OciCultural *73*
Classe: OciEsportiu *73*
Classe: OciNocturn *73*
Classe: OciParcAtraccions *74*
Classe:PuntSanitari *74*
Classe: SanitariHospital *74*
Classe: SanitariFarmacia *75*
Classe: PuntServeiPublic *75*
Classe: PuntTransport *75*
Classe: TransportEstacio *75*
Classe:PuntTuristic *76*
Classe: TuristicEspaiNatural *76*
Classe: EspaiNaturalHidrografic *76*
Classe: EspaiNaturalZonaVerda *77*
Classe: TuristicMonument *77*
Classe: MonumentHistoric *77*
Classe: MonumentReligios *77*
INSTANCIACIÓ DE LES CLASSES 78
TESTS REALITZATS 79
Proves amb el raonador *79*
Interrogació de l'ontologia amb SPARQL *81*

PREVISIONS DE FUTUR 85

BIBLIOGRAFIA 87

MATERIAL DE L'ASSIGNATURA 87
BIBLIOGRAFIA RECOMANADA 87
TESIS I PROJECTES FI DE CARRERA 90

Introducció

Al llarg de les tres darreres dècades, la industria turística ha sofert un creixement espectacular arribant a convertir-se en una part fonamental de les economies de moltes de regions i països que tenen en el turisme una gran font d'ingressos arribat a esser, en molts de casos, el seu sector estratègic més important. Per això un país depenent del turisme no pot només dedicar-se simplement a donar-se a conèixer mitjançant campanyes publicitàries sinó que li és de vital importància intentar potenciar i orientar al màxim la seva promoció turística i intentar que el nombre de turistes augmenti per a maximitzar el seus ingressos.

Amb l'explosió d'Internet, ha canviat la forma en que les usuaris poden accedir a la informació sobre els llocs a on volen visitar facilitant-los l'accés ràpid i immediat a tot tipus d'informació i introduint també una nova forma de contractació d'allotjaments, cotxes, passatges d'avió o excursions a la destinació que abans forçosament estaven quasi monopolitzats i controlats per les agències de viatges tradicionals i, a més alt nivell, pels touroperadors que proveïen directament una informació limitada als possibles viatgers.

Com que ara a Internet la informació existent que es pot trobar relacionada amb la industria del turisme és immensa quasi absent de cap tipus de normalització, cada cop és més difícil per part dels portals de serveis i dels operadors turístics proporcionar informació sobre qualsevol destí turístic al usuari que la sol·liciti i encertar completament amb la informació que aquest espera rebre d'acord amb les seves preferències és una tasca bàsica que s'ha d'intentar per tal d'assegurar al màxim la reserva però cada pic més complicada.

Per tant i donat el paper fonamental que té avui Internet i la gran quantitat d'informació que conté sobre qualsevol cosa que, a més, ha estat generada sense cap tipus de control o organització, serà necessari una forma òptima de poder analitzar totes les fonts d'informació possibles que siguin necessàries per a poder resoldre la consulta que ha fet el potencial client i serà bàsic intentar aconseguir un eina o tecnologia permeti que, de forma automàtica, els ordinadors puguin inferir qualque resposta vàlida de totes aquestes dades realitzant una localització, processament i integració de tota la informació rellevant disponible sobre el context desitjat pel client.

D'aquesta necessitat i dintre del camp de la tecnologia web a través d'Internet sorgeix el concepte de la Web Semàntica que té per objectiu proporcionar un significat ben definit a la informació, permetent així que els ordinadors tinguin un nivell de comprensió de la Web suficient com per poder realitzar un accés d'una forma natural tota la informació com una tasca relativament senzilla i se'ls hi proporciona la capacitat d'integrar dades provinents de diverses fonts i d'obtenir millors i més precisos resultats a l'hora de fer recerques sobre elles amb l'avantatge de poder aprofitar els processos d'automatització inherents en els computadors per a poder dur a termes les tasques més complexes.

Per a poder representa la informació relacionada amb aquest àmbit s'intenta adaptar el concepte clàssic d'ontologia o model conceptual amb la finalitat d'intentar facilitar la comunicació i l'intercanvi d'informació entre diferents sistemes. Degut a la seva importància

durant els darrers anys, s'ha intentat innovar i desenvolupar al màxim aquest concepte tant en l'àmbit privat com en l'àmbit acadèmic a través de diversos projectes que han generant una sèrie d'ontologies que donen solució a un plegat de problemes comuns a tots els àmbits i que conformen una base més o menys sòlida per a que es poguessin desenvolupar unes altres més específiques i més relacionades amb el turisme(o amb qualsevol altre àrea) que ja puguin donar solucions més concretes als problemes particulars d'aquesta àrea.

Dintre d'aquest camp que és el turisme i enquadrat aparició d'un projecte global que s'ha desenvolupat al llarg de varis anys amb l'objectiu el desenvolupar un marc de treball que pugui donar un suport a l'usuari en les seves rutes turístiques i que el permeti guiar-se en el seu recorregut en funció de la seva posició, les seves preferències i el moment del dia en què es trobi i poder recomanar-li una informació personalitzada segons les seves preferències o característiques particulars com a viatger es pot trobar en aquest projecte.

Aquest document s'exposa la elaboració del Projecte de fi de carrera "Ontologies Turístiques Geogràfiques", corresponent a l'àrea de XML i Web semàntica dels estudis d'Enginyeria Informàtica de la Universitat Oberta de Catalunya que tracta de la creació d'una ontologia turística geogràfica que permeti etiquetar els fitxers que contenguin informació sobre els punts d'interès per a poder representar els aspectes de l'àmbit turístic.

El document constarà d'un recull de fonaments bàsics que proporcionaran un marc de coneixements, un resum de l'estat de les ontologies turístiques més importants existents en l'actualitat i el desenvolupament d'una ontologia amb les característiques esmenades anteriorment i unes conclusions sobre el treball realitzat.

Objectius

Objectius del projecte

En aquest punt es descriuran els objectius que es pretenen assolir amb l'elaboració d'aquest Treball de Fi de Carrera (TFC) i dels objectius singulars del projecte.

Durant aquest projecte es tractaran aspectes directament relacionats amb XML i la Web semàntica aprofundint sobre el camp de les ontologies i adquirint coneixements necessaris per a crear-ne una i a través dels llenguatges de representació i eines més utilitzats.

Els objectius principals d'aquest projecte són clars:

1. Proporcionar un marc de coneixements sobre l'àrea de les ontologies, mecanismes existents per a la seva creació, representació i aplicació a entorns de Web Semàntica.
2. Recopilar l'estat actual de l'art de les ontologies turístiques actuals a data d'aquest article.
3. La creació d'una ontologia dins d'un àmbit molt específic com és el turístic, explicant el seu desenvolupament i característiques particulars.
4. Demostrar el funcionament de l'ontologia i comprovar la seva utilitat a l'hora de catalogar informació relacionada amb aquest àmbit d'acció.

Com a objectiu secundari de d'aquest TFC tindríem que centrar-nos en el marc d'una assignatura que té com a finalitat la realització d'un treball de síntesi dels coneixements adquirits en diferents assignatures de la carrera i mostrar l'assoliment d'una sèrie de coneixements adquirits per mi, l'alumne, durant els meus estudis i més concretament en aquest cas en la carrera d'Enginyeria Tècnica en Informàtica dintre de l'àrea de XML i Web Semàntica realitzant un exercici d'anàlisi d'un problema complex de tipus pràctic relacionat amb aquest àrea i transformant-lo en un projecte informàtic el qual s'haurà de desenvolupar i planificar mitjanant un pla de treball aplicant-hi una metodologia adient i treballar a fons tots els aspectes formals del desenvolupament d'un projecte per acabar sintetitzant una solució al problema plantejat i elaborant una memòria del projecte i una presentació dels resultats fins obtinguts.

Abast del projecte

L'abast d'aquest projecte compren des de l'estudi de les ontologies turístiques existents fins a la creació i implementació d'una ontologia per representar els aspectes relacionats amb l'àmbit turístic amb documents associats a punt d'interès.

Resultats esperats

S'espera obtenir amb la realització d'aquest projecte per una banda una visió de l'estat actual de les ontologies turístiques i les eines i recursos que existeixen per la seva gestió. I d'altra la creació d'una ontologia que permeti representar els aspectes relacionats amb l'àmbit turístic amb l'àmbit així com la latitud i longitud de la informació associada per cadascun dels fitxers.

Fonaments

Què és una ontologia

El concepte d'ontologia té el seu origen a l'antiga filosofia grega, i encara que es pot establir el seu origen en textos aristotèlics, ja que des de un primer moment que va parlar d'una "*philosofia prima*"(filosofia primera) i en la que va incloure, tant l'estudi de l'ens en quan a l'ens, com l'estudi d'un ens principal al qual es subordinaven els altres ens, i comença a distingir el que després s'anomenaria ontologia. La primera utilització del terme ontologia en la seva forma greca, οντολογία (unió del terme grec οντος, genitiu del participi del verb εἰμί, que significa ésser o estar, i de λόγος, que significa ciència, estudi o teoria), va esser al segle XVII quan en Rodolfo Gloclenio, creador del terme *psicologia*, ho va incloure a la seva obra *Lexicon "philosophicum, quo tanquam clave philosophiae fores aperiuntur"* l'any 1613 a on la definí com "*philosophia de ente*"(la filosofia de l'ens). Uns anys després va esser Leibniz, el que utilitzà el terme a la seva obra "*Introductio ad Encyclopaediam arcanam"* (1683-1685) en la qual definí ontologia com a "*scientia de aliquo et nihilo, ente et non ente, re et modo rei, substantiel et accidente*"(ciència del que és i del que no és, de l'ens i el no ens, de les coses i dels seus modes, de la substància i del accident).

El primer en fer servir el terme com a terme tècnic va esser Jean Le Clerc al seu segon tractat titulat "*Ontología sive de ente in genere*"(1692) contingut a les seves "O*pera philosophíca in quatuor volumina digesta*", però fou Cristian Wolff el que popularitzà el terme definint-la a la seva obra "*Philosophia prima sive ontología methodo scientifica pertracta, qua omncs cognitionis humanae principia continentur*" (1730) com una "scientia entis in genere, quatenus ens ens"(ciència de l'ens en general, en quant que ens) i afirmant que utilitza un mètode demostratiu(és a dir, racional i deductiu) i que analitza els predicats de tots els ens coma tals.

Tots aquests sentits i definicions varen contribuir a identificar-la a la branca de la filosofia anomenada metafísica, que es centra en l'estudi de la naturalesa i organització de la realitat, és a dir el que existeix. Les ontologies en filosofia s'ocupen del establiment d'aquelles categories o modes generals de ser que tenen les coses, partint de l'estudi profund de les seves propietats, estructures i sistemes i en com els ens poden ser classificats de diferents formes dins d'unes jerarquies, i subdividits segons les similituds i diferències que presenten.

Ontologia a l'àmbit de la informàtica

Des de la segona meitat del segle 20, han estat debatent a la comunitat filosòfica els mètodes possibles per a poder construir ontologies sense obtenir cap resultat satisfactori. En canvi, al camp de la informàtica amb un debat significativament menor sobre com fer-ho, si que s'han realitzat avanços significatius,construint algunes grans i robustes ontologies com Wordnet o Cic. Des de la segona part dels anys 70, els investigadors en el camp de la intel·ligència artificial(IA) ja van esbrinar que la clau per a poder construir grans sistemes d'IA potents i sòlids és que la captura de coneixement, i que es poden crear ontologies com a models computacionals que permetin certs tipus de raonament automatitzat.

En el camp de la informàtica no es va començar a parlar del terme ontologia fins a la darrera meitat dels anys 80 en que la comunitat de la branca d'IA va començar a utilitzar el terme d'ontologia per a referir-se a les teories per a modelar el món com a un component dels sistemes de coneixement.

Va esser realment a principis de la dècada dels 90 quan van començar a proliferar definicions d'ontologia i la més estesa i acceptada d'elles, és la de T. R. Gruber al 1993 continguda a la seva publicació "*Toward Principles for the Design of Ontologies Used for Knowledge Sharing*", i la seva clarificació posterior, que està basada en el concepte de conceptualització proposat per Genesereth i Nilsson al 1987, i que es refereix al conjunt de conceptes i relacions entre conceptes que son rellevants en un domini o una aplicació.

Gruber defineix una ontologia com "*An ontology is an explicit specification of a conceptualization*" (Una ontologia és una especificació explícita d'una conceptualització). Aquesta definició va esser després clarificada per Borst al 1997: *"An ontology is a formal specification of a shared conceptualization"(*Una ontologia és una especificació formal d'una conceptualització compartida).

Unificant ambdues definicions s'obté com a resultat que "Una ontologia és una especificació formal i explícita d'una conceptualització compartida" i sobre el que es poden inferir una sèrie

d'idees clares, com que fa referència a un model abstracte de qualque fenomen del món en el que s'identifiquen els seus conceptes rellevants i s'especifiquen de forma explícita els seus conceptes i restriccions existents entre ells utilitzant un llenguatge formal, cosa que implica qualque tipus d'organització teòrica de termes i relacions que puguin esser utilitzats com a eina per l'anàlisi de conceptes d'un domini i pugui esser processat per computadors i no sols per persones. A més deixa clar que una ontologia és un coneixement compartit, fruit del consens dintre d'una comunitat.

Gruber afirmava que les ontologies sovint podien esser equiparables a les jerarquies taxonòmiques de les classes, les seves definicions i relacions entre elles i que les ontologies no poden limitar-se a aquestes formes, ni poden ser definides amb terminologies conservadores que no afegeixen cap coneixement del món, i que, per tant, per especificar una conceptualització, existeix la necessitat de definir axiomes que no restringeixin les possibles interpretacions dels termes definits. Aquesta definició és consistent amb l'ús d'ontologia com a un conjunt de conceptes i definicions de forma més general però té ja un sentit diferent al que se li donava en filosofia.

A partir de Gruber s'han succeït moltes altres definicions com la de Breuker al 1999 que defineix una ontologia com "*una representació explícita d'una conceptualització cognitiva", es a dir, la descripció dels components de coneixement rellevants amb l'àmbit del modelat*.

La creació d'una ontologia ofereix l'avantatge de que la categorització dels elements i relacions que intervenen en el model de coneixement es fa de forma explícita i que, per tant, a més de que el model de coneixement pot esser editat i gestionat, és possible la seva transmissió de forma que un sistema entengui la conceptualització que se ha utilitzat a l'altre donant lloc a un coneixement d'un domini reutilitzable i mantenible.

En termes pareguts V.Devedžić al 2002 entengué que una ontologia és "*un sistema de conceptes definits e interpretats d'una manera declarativa. El sistema defineix el vocabulari d'un domini i les restriccions que se'ls apliquen per a que aquests pugin esser combinats en el model del domini*".

N. Guarino al 1998 defineix ontologia com "*una teoria lògica que considera el significat intencional d'un vocabulari formal, és a dir, el seu compromís ontològic per a una conceptualització particular del món*".

Components d'una ontologia

Des d'un punt de vista de l'enginyeria, una ontologia és un artefacte que està construït amb un vocabulari específic per a descriure una certa realitzat a més d'un conjunt de suposicions que tenen en compte el significat d'aquest vocabulari. De forma independent a aquesta realitat i del vocabulari utilitzat, els components d'una ontologia solen esser sempre els mateixos:

Axiomes	Són elements que permeten el modelat de les asseveracions que es compleixen sempre dintre de la realitat, establint una sèrie de condicions i relacions amb la jerarquia de l'ontologia conceptes i atributs definits. Poden esser estructurals i/o no estructurals.

Regles	Són un tipus d'axioma específic que indiquen causalitat. Són del tipus: Si X -> Y i descriuen inferències lògiques.
Classe	És un conjunt o col·lecció d'objectes en que cadascun d'ells és una instancia d'aquesta classe. Des del punt de vista de la lògica els objectes d'una classe es poden descriure especificant les propietats que han de satisfer per a pertànyer a aquella classe. Les classes seran la base de la descripció del coneixement en les ontologies ja que descriuen els conceptes inclosos al domini. Una classe pot esser dividida en subclasses, les quals representaran conceptes més específics que la classe a la que pertanyen. Una classe en la qual els seus components són classes es denomina superclasse o metaclasse.
Individus o instàncies	Són els objectes que pertanyen a una classe. No poden esser dividits sense perdre la seva estructura i característiques funcionals.
Propietats, atributs o slots	Són els aspectes, propietats, característiques o paràmetres que els objectes poden tenir, el conjunt del qual descriu al objecte. Les propietats d'una classe són heretades per les seves subclasses i pels individus que pertanyen a ella.
Restriccions	Són rangs, especificacions i descripcions formals han de complir les propietats o atributs perquè siguin entrades vàlides.
Relacions	Se estableixen entre els conceptes d'una ontologia i volen representar la forma en que interaccionen les classes i els individus entre si. Es solen definir generalment com el producte cartesià de n conjunts: R: C1 x C2 x …x Cn.
Funcions	Són un tipus concret de relacions on s'identifica un element mitjançant el càlcul d'una funció que considera diversos elements d'una ontologia.

Procés de creació d'una ontologia

Dissenyar i crear una ontologia es un procés complexa i no existeix una única ontologia per a representar un domini determinar sinó que dependrà en gran mesura de múltiples variables com el context en que s'ha construït, la persona que l'hagi construït, etc.. Tampoc existeix una metodologia única per a la creació d'ontologies sinó que n'existeixen vàries i encara que no s'ha determinat cap mètode com al correcte per, si que la majoria coincideixen en que hi ha una sèrie de passos s'haurien de considerar en el procés de creació, que són:

Identificar i determinar el seu domini i el seu abast	Definir el domini i l'abast de l'ontologia per a tenir clar els seus objectius mirant de trobar les respostes adequades a preguntes del tipus ¿Quin és el domini que l'ontologia ha de cobrir?,¿Per a què s'utilitzarà?,¿A quines preguntes a de poder donar resposta? o ¿Qui utilitzarà i mantindrà l'ontologia?. Una de les formes de determinar l'abast de l'ontologia és preparar una llista de preguntes no exhaustiva que la base del coneixement basada en l'ontologia hagi d'esser capaç de respondre. Aquesta llista rebrà el nombre de competència i ha de servir de base per a les proves més endavant . Això ajudarà a decidir quin tipus d'informació s'ha d'incloure i quina no és necessària, si calen sinònims, l'explicació dels conceptes, etc... Les respostes a aquestes preguntes poden variar al llarg del disseny de l'ontologia.
Considerar reutilitzar ontologies existents	S'ha de comprovar si existeixen ontologies ja fetes que es puguin utilitzar o reutilitzar-ne parts. La reutilització d'ontologies existents pot esser una bona solució si el sistema o pàgina web que la fa servir ha d'interactuar amb altres aplicacions que utilitzen ontologies particulars.

	Moltes ontologies estan disponibles en format electrònic i poden esser importades directament amb el sistema de desenvolupament que s'utilitzi aprofitant-la tota o les parts que s'estimin convenients. A causa del formalisme amb el que estan expressades generalment les ontologies no es permet la importació i exportació però tot i això, la traducció de les ontologies cap a un altre formalisme no sol esser gaire complicat.
Enumerar els termes importants de les ontologies	Abans de començar la construcció de l'ontologia es convenient fer un llistat de quins seran els termes que es volen construir o parlar a un usuari. S'ha de demanar, De quins termes es volen parlar? Quines propietats tenen? Què es vol dir d'ells? En principi només es farà una llista de termes sense tenir en compte si els termes es solapen amb el que representen, de les relacions entre ells, o les propietats que haurien de tenir les classes a definir.
Definició de la jerarquia de classes	Un cop es té el llistat dels termes, ja es pot passar a definir la jerarquia de classes i per a dur aquesta tasca a terme es tenen dues opcions possibles. Es pot començar definint els conceptes més generals del domini i anar realitzant especialitzacions del mateixos o directament i a partir de les classes més específiques anar agrupant-les en conceptes més genèrics. Es pot també intentar combinar ambdós mètodes si es te més experiència però pot esser bastant complicat. Alhora de definir la jerarquia de classes el que s'ha d'intentar, independentment del mètode triat, és que, partint del llistat realitzat, seleccionar els termes que descriuen objectes i que seran les classes que formaran part de la jerarquia de classes.
Definició de les propietats les classes	Amb jerarquia de les classes definida s'ha de tenir en compte que no només amb ella es pot obtenir prou informació per a contestar les qüestions que s'han plantejat en el primer pas i encara queda definir com serà la seva estructura interna, és a dir, les seves propietats o slots. Del llistat realitzat i un pic extrets els termes que identificaven les classes, els termes restants probablement siguin propietats d'aquestes classes i s'ha de determinar a quina classe descriuen tenint en compte que les subclasses d'una classe hereten les seves propietats i s'assignaran a les classes més generals sempre que sigui possible.
Definició de les restriccions de les propietats.	Un cop es tenen les propietats definides, s'han de definir les seves restriccions deixant ben establerts quins seran els possibles valors que se'ls hi podrà assignar. Les restriccions es farà en termes de cardinalitat, indicant quants valors pot prendre alhora;, de tipus de valor que poden guardar(caràcter, numèric, enter, booleà, etc...); i domini i rang, si el tipus d'una propietat és un altra classe.
Establir les relacions semàntiques	S'ha de realitzar un resum de les possibles relacions existents entre els objectes de l'ontologia i de les possibles interaccions que es pugin presentar entre ells. S'ha d'incloure la transitivitat entre les classes que pugi tenir la relació.
Creació de les instàncies	Aquest procés simplement consta d'escollir la classe de la que es vol crear una instància, crear-la i omplir amb un valor que compleixi les restriccions, les propietats de la classe escollida i respectant les relacions definides.
Avaluació i documentació	Avaluar l'ontologia a partir del disseny definitiu que s'ha creat per a comprovar que és correcta, realitzant les proves que es considerin adequades per a la seva validació. S'ha de documentar també durant tot el procés, a més de l'ontologia en si mateixa, les passes seguides i la justificació de les decisions preses en cada cas. Aquesta documentació de la té per objecte facilitar-ne l'ús i la reutilització.

Tipus d'ontologies

Existeixen moltes formes de classificar les ontologies depenent de múltiples variants proposades per diferents autors. S'exposa a continuació un resum de les més importants.

Segons en M. Uschold (1996) existeixen tres dimensions sobre les quals es poden distingir els tipus d'ontologia:

1. Segons Formalitat, referint-se al grau de formalisme del llenguatge utilitzat per expressar la seva conceptualització. Segons la seva formalitat es pot distingir entre:

Altament informal	Si està expressada en llenguatge natural (per exemple un glossari de termes).
Informal estructurada	Si utilitza llenguatge natural estructurat i restringit, que permeti la reducció de l'ambigüitat.
Semiformal	Si utilitza llenguatge de definició formal.
Rigorosament formal	La definició de termes es du a terme de forma meticulosa utilitzant semàntica formal ,teoremes, i proves d'aquetes propietats com a solidesa.

2. Segons el seu propòsit i la intenció que se li vulgui donar es pot distingir entre:

Per a la comunicació entre persones	En que una ontologia informal no ambigua pot esser suficient.
Per a la interoperabilitat entre sistemes	S'utilitzarà per a dur a terme traduccions entre diferents mètodes, llenguatges, programari, etc... En estos casos l'ontologia s'utilitza com a un format d'intercanvi de coneixement.
Per beneficiar la enginyeria de sistemes	Beneficien les aplicacions suportant aspectes com a la reutilització de components de programari en un domini d'interès, l'adquisició de coneixement, la fiabilitat del sistema proporcionant consistència en el coneixement utilitzat o la especificació dels sistemes de programari identificant els requisits informació.

3. Per últim, amb respecte a al seva matèria es pot distingir:

De domini	Modelen un domini o subdomini determinat caracteritzant disciplines específiques, amb independència dels problemes o tasques rellevants en elles.
De tasca, de mètode o de resolució de problemes	Conceptualitzen el problema o la tasca a resoldre en un domini.
De representació	També anomenades metaontologies en el que l'objecte caracteritzat és un llenguatge de representació del coneixement.

N. Guarino (1998) defineix que s'ha de fer un altra distinció i diferencia entre ontologies refinades(off-line) i no refinades(on-line) . Les no refinades tenen un nombre mínim d'axiomes

i el seu objectiu es esser compartides pels usuaris que concorren en una determinada visió el món. Tenen una major capacitat per a ser compartides i han d'esser utilitzades en línia per a poder donar suport en funcionalitat de sistemes d'informació. Les refinades precisen d'un llenguatge d'alta expressivitat i tenen un gran nombre d'axiomes per la qual cosa s'han d'utilitzar fora de línia i com a referència.

En van Heijst (1997) proposa un altre tipus de classificació de les ontologies segons la seva quantitat i tipus d'estructura distingint:

Terminològiques	Especifiquen els termes que s'han d'utilitzar per a representar el coneixement en l'univers del discurs i que solen esser utilitzats per a unificar vocabulari en un camp determinat.
D'informació	Especifiquen l'estructura d'emmagatzemament de bases de dades oferint un marc per l'emmagatzemament estandaritzat d'informació.
De modelat de coneixement	Especifiquen conceptualitzacions del coneixement.

Per últim es pot fer una darrera classificació en funció de la usabilitat i la reusabilitat tenint en compte que la relació entre les dues característiques és inversament proporcional i quan com més reutilitzable és una ontologia menys usable és, i a l'invers.

En general les ontologies, independentment del tipus al que pertanyin, interessarà que incloguin glossaris de termes més o menys complets i complexos aportant una descripció formal cosa que resultarà de més utilitat als programes proporcionant-los d'una major capacitat d'inferència.

Llenguatges empleats per a definir ontologies

Al llarg d'aquest capítol es comentaran alguns dels llenguatges habituals empleats per a definir les ontologies, explicant els seus aspectes més importants, origen i sintaxi. Inicialment s'explicarà què és el XML, base de la resta i del qual deriven els altres llenguatges que s'hauran d'utilitzar i aprendre per a poder realitzar aquest projecte.

XML

En aquest apartat es farà una introducció per a conèixer i comprendre els fonaments bàsics del llenguatge XML(eXtensible Markup Languatge) fent un repàs al seu origen, la seva estructura i a seva sintaxi, ja que XML representa una primera aproximació a poder definir una ontologia i ,actualment, és l'estàndard més estès avui en dia per a la representació de dades.

XML permet realitzar una estructuració lògica de dades i de documents en forma d'arbres d'etiquetes que tenen valor i atributs, proporcionant una sintaxi que facilita molt l'intercanvi de dades i de documents estructurats entre programes i aplicacions. Aquesta estructura, com es veurà més endavant, ha de seguir unes regles, la qual cosa permet crear uns motors de cerca àgils i eficients que permeten un accés ràpid a les dades contingudes en un document en XML i per tant s'esdevé perfecte per a l'intercanvi d'informació i cooperació entre programes.

Des de que XML es un estàndard reconegut, la majoria dels fabricants de programari l'han acceptat i serveix per a la majoria de les plataformes webs, sistemes operatius, navegadors, llenguatges de programació o bases de dades actuals. Abans de la seva existència cada aplicació utilitzava el seu format específic de dades i realitzar la conversió d'un tipus de dades a un altre podia esser una tasca difícil i depenent de la creació d'un programari addicional que s'encarregués de la traducció del format original al de l'aplicació de destí, amb el cost en diners i temps que pot suposar això. Per aquest motiu el seu èxit com estàndard.

Aquest apartat servirà sobretot, per intentar establir una base i com introducció als apartats següents en els que es veuran els llenguatges RDF i OWL que si aporten la capacitat de poder afegir una semàntica a les dades i possibiliten la definició d'una ontologia.

Origen

L'aparició de la seva primera versió formal va esser al febrer de l'any 1998, encara que ja es feia feina amb ell des d'un parell d'anys enrere ja que va néixer a partir d'un subconjunt simplificat de SGML i és, com ell, un metallenguatge, es a dir, un llenguatge per a definir llenguatges i va esser una recomanació del W3C, que és una especificació estable desenvolupada per un grup de treball del W3C i revisada pels seus membres.

S'ha d'explicar que quan va aparèixer, ja feia temps que ja existia HTML, que era massa poc flexible al esser un llenguatge de marcat basat en etiquetes i no un metallenguatge com XML, i que servia per codificar documents i distribuir-los per la World Wide Web. HTML és un llenguatge que és interpretat pels navegadors que mostren el seu contingut al contrari que XML, que utilitza les etiquetes només per a delimitar peces de dades i no deixa que aquestes es pugin interpretar ja que especifica el significat de cada etiqueta i atribut. Les regles per als arxius XML són més estrictes que en HTML, ja que es poden definir plantilles amb unes normes estrictes que ha de seguir el document. La mancança principal de HTML és que la informació que codifica fa referència només a la seva presentació i no al seu contingut, és a dir, un document HTML, és interpretat per un navegador de forma seqüencial de dalt a baix i només l'indica què i cóm es visualitza. El problema d'intercomunicació entre ordinadors, aplicacions i bases de dades no podia solucionar-se amb HTML.

Al principi, el grup de treball de XML i el W3C, es van plantejar com a objectius que fos directament utilitzable a Internet, suportar gran varietat d'aplicacions i amb un disseny formal i concís que pogués fer que els documents XML i els programes que els hagin de processar fossin fàcils de crear. De fet una de les claus del seu èxit és la seva senzillesa ja que no és més que codi ASCII que és pot editar amb qualsevol navegador.

La solució a les necessitats plantejades va esser la proposta del estàndard multiplataforma anomenat XML, que amb el temps ha estat àmpliament acceptat i adoptat per la majoria de fabricants i, per tant, els sistemes operatius, navegadors i bases de dades el saben interpretar.

Característiques del XML

Així doncs i resumint les característiques més rellevants del XML es pot dir que les característiques més rellevants que té XML serien:

És un dialecte simplificat de SGML però amb les seves propietats més importants i desitjables. Pretén esser raonablement simple.
Al esser un metallenguatge, permet la creació d'altres llenguatges de marques. És extensible, ja que les seves etiquetes no estan predefinides.
Ofereix una representació estructurada de les dades fàcilment implementable i distribuïble.
Garanteix que les dades estructurades siguin uniformes i independents d'aplicacions o fabricants facilitat la interoperabilitat i de la integració de dades procedents de tot tipus de fonts.
S'ha convertit en l'estàndard per a l'intercanvi d'informació a Internet estès i acceptat.
Fàcilment processable pels humans i les màquines, sent utilitzable amb qualsevol alfabet o llenguatge.
Codifica dades d'acord a un esquema semàntic i separa informació de presentació i format.
Està basat en marques sintàctiques que ha de complir una sèrie de regles lèxiques i normes sintàctiques etiquetant els components del document d'acord amb elles.
Se li pot associar una gramàtica per a definir un model d'estructura de document DTD-esquema.

Estructura d'un document en XML

Un document XML té una estructura concreta amb dues parts diferenciades: Pròleg i instancia del document. Així mateix el pròleg consta de 3 seccions: La declaració del XML , La declaració del tipus de document i referència a una fulla d'estil externa. A continuació es pot veure un exemple de les diferents parts comentades:

```
<?xml version="1.0" encoding="ISO-8859-1" standalone="no"?>
<!DOCTYPE Classe SYSTEM "classe.dtd">
<?xml-stylesheet href="classehtml.xsl" type="text/xsl"?>

<classe>
 <alumne>
   <nom>Joan</nom >
   <cognoms>Manresa Pou</ cognoms >
   <edat>33</ edat >
 </alumne>
<classe>
```

Declaració del XML
Declaració de tipus de document
Fulla d'estil externa
} **Pròleg**

} **Instància**

Pròleg

Com s'ha dit abans el pròleg es divideix en tres part:

Declaració del XML	Indica la versió o especificació de XML utilitzada al document; la codificació, que és opcional, representa el joc de caràcters que utilitza el document; i la declaració de document aïllat, també opcional, indica aquest document fa referència o no a una entitat externa.
Declaració del tipus de document [Opcional]	Definirà l'estructura i sintaxi que han de seguir els documents que siguin d'aquest tipus. Ja s'explicarà més endavant i en profunditat com es fa aquesta definició, ara només s'assenyala com s'associa el DTD al XML..

	La declaració de tipus, tindrà el format: <!DOCTYPE nombre DTD> a on NOM és el nom del tipus en qüestió i DTD una referència a la declaració del tipus de document que pot esser: 1. La definició d'un tipus de document o DTD. 2. Una referència a un recurs externs que conté la DTD d'aquest tipus. 3. Una referència a un recurs extern amb la DTD, i la definició de part de la DTD aplicable únicament a aquest document.
Referència a una fulla d'estil [opcional]	Indicarà la forma de representar el document, són instruccions de processament. Com es veurà més endavant, es pot optar per dues tecnologies: Fulles d'estil en cascada (CSS) o Fulles de transformacions XSL . Més endavant ja s'explicarà en detall quina estructura i com funcionen aquestes definicions d'estil en apartats posteriors.

Instància

Dintre de la instància d'un document es troba el contingut del document pròpiament dit. La instància comença amb una marca inici*(<element>)* d'un element (element arrel) i acaba amb la marca de fi d'aquest mateix element*(</element>*). L'arrel del document pot contenir més elements formant una estructura jeràrquica en forma d'arbre contenent. A l'instancia del document també es pot fer referència a un esquema XML, aquest esquema es d'una tecnologia alternativa al DTD anomenada XML-Schema i al igual que al igual que DTD serveix per a definir la gramàtica que ha de seguir el document. Més endavant ja es parlarà d'aquestes dues tecnologies en detall, però a la següent figura es pot veure un exemple del seva utilització:

```
<?xml version="1.0"?>
<Libreria xmlns="http://www.aules.org"
      xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
    xsi:schemaLocation="http://www.aules.org libreria.xsd">
<classe>
<alumne>
   <nom>Joan</nom >
   <cognoms>Manresa Pou</ cognoms >
   <edat>33</ edat >
 </alumne>
</classe>
```

Components bàsics d'una instància

A la instància d'un document es poden trobar el components bàsics són els elements, els atributs i les dades. Un exemple seria el següent:

```
<projecte nom="Ontologiaturistica" default="default" basedir=".">
  <descripcio>
    Aquets projecet té per objecte la de crear na ontologia turística
sobre punts d'interès turístic
  </descripcio>
  <import file="ontoproject/build.xml"/>
  <target name="-post-compile">
    <obfuscate>
      <fileset dir="./documents"/>
    </obfuscate>
  </target>
</projecte>
```

Elements	Constitueixen les unitats bàsiques dels documents i representen els components lògics que el conformen. Cadascun d'ells es descriu a través d'una etiqueta a on hi consta el nom del tipus d'element i una llista d'atributs[opcional] que indicaran les propietats de l'element. El contingut de l'element entre marques d'inici i fi. Es pot observar al exemple, que es declara l'element arrel a través de l'etiqueta projecte i com es situa el seu contingut entre la marca d'inici <projecte [llistat_atributs]> i la marca final </projecte>.
Atributs	Els atributs descriuen les propietats dels elements i formen part de l'etiqueta. Cadascun d'ells es representa a través d'una parella nom-valor en què aquest últim ha d'anar tancat entre cometes com es pot veure al exemple anterior al atribut nom de l'etiqueta del element <projecte>.
Dades	Els elements del document contenen, a més dels atributs, dades. Aquestes es troben entre les marques d'inici i de fi, i poden ser de diferents tipus, entre els quals hi ha d'altres elements. També poden esser buits i no tenir-ne.
Comentaris	A part d'això es podran introduir comentaris a mode de notes o explicacions per part del programador al documents XML que no seran interpretats pels parsers.

Els namespaces

Els namespaces(espais del nom) en els elements de XML proporcionen un mètode per a la unicitat, que marca les diferencies entre diversos autors d'esquemes que utilitzen els mateixos noms d'elements i que seran bàsics a l'hora de definir altres llenguatges com el RDF i el OWL.

Segons la definició inclosa a la *W3C Namespaces Recommendation*, un espai de noms XML és una col·lecció de noms d'elements i atributs qualificats mitjançant un URI (*Uniform Resource Identifier*), i bàsicament s'utilitzen per evitar conflictes de noms quan es combinen dades de diverses fonts XML i n'existeixen dos tipus: els namespaces per defecte i els explícits.

L'espai de nom per defecte, es declara només un pic al document XML i després s'aplica a l'element en que ha estat declarat i a tots els elements que jeràrquicament siguin fills seus.

Els espais de nom explícits, en canvi, només s'aplicaran a l'element on han estat declarats sense estendre's als seus elements fills. Així es poden combinar elements de diverses fonts

Documents ben formats i vàlids

Als documents XML s'ha de tenir molt en compte dues característiques que són molt importants: Estar ben format i ser vàlid.

En primer lloc i per a poder explicar què és per a un document XML estar ben format i què és ser vàlid s'ha de recordar el que s'ha dit als apartats anteriors sobre que un DTD o un XML-Schema defineixen com ha d'esser un document, és a dir, defineixen els seus elements i atributs i com s'estructuren i relacionen. Recordant això es pot dir que el document XML:

Està ben format	Si és un document absent d'errors de sintaxi d'acord am les especificacions de XML versió 1.0.
És vàlid	Si està ben format i a més segueix una definició detallada a un DTD o a del XML-Schema.

Per a comprovar si un document XML compleix aquestes propietats serà necessari el que s'anomena un parser, que serà no validador si només comprova si el document compleix les especificacions o validador si a més contrasta el contingut del document amb les regles contingudes al DTD o el XML-Schema.

Perquè un document XML pugui ser processat i compartit entre diferents aplicacions és necessari que estigui ben format i pot esser interessant també que sigui vàlid si, per exemple, es vol fixar un format estricte per a que es comuniquin dues aplicacions i assegurar que el document que li enviï una a l'altra estigui en un format esperat. Profunditzant un poc més en el tema es definirà que s'entén com a document ben format.

Documents ben formats

Per ser un document XML ben format(*Well-formed XML*), el document ha de complir rigorosament una sèrie de normes. El conjunt de totes les normes que ha de complir un document XML venen donades pel W3C en l' XML 1.0 Recommendation.

Un petit resum de les regles que asseguraria que estigui el document ben format inclouria per a cadascun dels components del document les següents regles generals:

Normes generals per als elements

Com a normes generals per als diferents elements s'inclouen les següents:

- El nom de l'element no pot contenir espais però són vàlides tan lletres majúscules i minúscules. Tampoc no pot començar per un número, cap signe de puntuació, ni cap combinació de la paraula "xml".
- El nom ha de començar després del caràcter "<", sense espai entre ells.
- El nom d'inici i final d'element ha de ser el mateix, tenint en compte que XML distingeix entre majúscules i minúscules.
- El document ha de tenir almenys un element i només en pot existir un d'element arrel amb el qual tots els altres han d'estar jeràrquicament dins d'aquest. Aquest enniuament és estricte i no es pot posar la marca de fi d'un element pare si s'ha posat la marca de fi dels seus elements fills.
- Un element que no té contingut, consisteix en les etiquetes d'inici i fi seguides.

- Si l'element sense contingut és l'element arrel, la seva sintaxi forçosament ha d'esser *<Arrel></Arrel>* encara que sigui un element buit.

Normes per als Atributs

Com a normes generals per als diferents atributs s'inclouen:

- Els atributs han de ser declarats dins l'etiqueta d'inici de l'element i consisteixen en un nom i un valor assignat que va entre cometes simples o dobles i precedit del signe "**=**".
- Un element pot contenir tants atributs com es vulgui i si n'hi ha més d'un han d'estar separats per un espai tenint en compte que dins un mateix element no pot haver-hi més d'un atribut amb el mateix nom i un el seu nom no pot contenir-hi espais.

Normes per als Comentaris.

Respecte els comentaris són aquestes:

- Tot comentari ha de començar amb els caràcters "<!--", o bé, "<!---" i acabar amb els caràcters "-->". Dins el text no hi pot aparèixer la seqüència de caràcters: "--".

Normes per als caràcters reservats.

Al XML existeixen una sèrie de caràcters que són reservats perquè tenen una funció determinada(marquen inici etiqueta, etc...) que els parsers quans els troben directament intentaran interpretar i que per això s'ha d'escriure d'un altra forma representant-los amb una cadena de caràcters diferents que el parser interpretarà com el caràcter. El caràcter en si quedarà reservat per a la possible funció. Es poden resumir en aquesta taula:

Caràcter	Cadena de substitució
&	&
<	<
>	>
'	"
''	'

Una altra manera de representar-los és amb la utilització de la instrucció: *<![CDATA]{ seqüència de caràcters a escriure }]>.* Si s'utilitza aquesta instrucció, tot el contingut no serà interpretat pel parser que ho escriurà com a text pla. Aquesta instrucció és molt útil i permet, introduir seqüències de codi HTML com a possibles valors inclosos als atributs o als elements.

Documents vàlids

Abans ja s'ha explicat que un document XML, per poder-se tractar i intercanviar entre aplicacions, seria convenient que fos també un document vàlid. Serà vàlid si la seva estructura interna compleix una sèrie de normes, que prèviament han estat definits en un esquema.

Aquest esquema és també un document de text que descriu l'estructura que pot tenir el document XML i si l'estructura del document coincideix amb la de l'esquema, llavors es dirà que es tracta d'un document vàlid. Cal tenir present que tot document vàlid és també un document ben format, però no necessàriament tot document ben format és vàlid.

Existeixen bàsicament dos tipus d'esquemes per efectuar la validació d'un document XML:

DTD's	És la validació original descrita en el *W3C XML Recommendation* versió 1.0, i no estan basats en XML.
XML schemas	És la solució actual del W*3C Recommendation* per validar documents XML. Substitueix als DTD's i estan escrits en XML.

Validació mitjançant DTD

Com s'ha explicat abans al DTD es definiran els elements que conformaran aquest tipus de document i com ha d'estar organitzat perquè sigui correcte. Respecte al DTD es pot dir que és capaç de determinar:

1. Els elements possibles que poden aparèixer en el document.
2. Els atributs que poden utilitzar al costat de cada element.
3. Com es poden niar els elements (pares i fills).
4. L'ordre en què han d'aparèixer els elements fills d'un mateix pare.
5. El nombre permès d'elements fills.
6. Si un element pot ser buit o no.
7. Tipus de dades per a elements i atributs.
8. Valors per defecte i fixos per a elements i atributs.

El DTD pot estar inclòs al propi document XML o ser extern, en aquest cas el document XML s'associarà amb el DTD mitjançant l'etiqueta *<! DOCTYPE>* en el pròleg i en ell es podrà especificar un DTD complet (DTD intern), o la URL on trobar el DTD (DTD externa) .

Entre els avantatges que aporta el DTD està que sigui un estàndard robust i madur, això implica que existeixen gran quantitat de desenvolupaments compatibles (programes d'edició, parsers, transformadors, exemples, etc.). Com a contrapartida es podria dir que els DTDs són SGML, no XML, de manera que no recullen conceptes recents com els espais de nom i també se li pot retreure que la definició de tipus és bastant limitada.

Per tot això van sorgir, com una alternativa a les DTD, els XML Schema Definition (Definició d'esquema XML) que estan desenvolupats amb XML i permeten definir els espais de nom.

Validació mitjançant XML-Schema Definition

Com s'ha explicat, les definicions d'esquemes XML (XSD) van sorgir com alternativa als DTD i també serveixen per a descriure l'estructura de documents XML i que se'ls pugi validar. Al estar escrits en XML, són extensibles, més rics i útils que els DTD, suportant els espais de noms i tot tipus de dades. A més, tenen l'avantatge d'esser igualment processables.

Un XSD permet definir el tipus de contingut d'un element o atribut i especificar si ha de ser un número enter, una cadena de text, una data, etc...essent el primer pas que per a poder utilitzar-se, indicar aquest fet al document XML que es vol validar. L'associació del XSD es fa afegint una sèrie d'elements al element arrel del document i es fa en una sèrie d'atributs:

xmlns	Indica sempre l'espai de noms per defecte.
xmlns:xsi	Indica que l'atribut schemaLocation utilitzat és el del espai de noms XMLSchema-instance.
xsi:schemaLocation	Aquest atribut té dos valors separats per un espai, el primer indicarà quin espai de noms utilitzar I el segon es el nom del fitxer que s'ha d'utilitzar per aquest espai de noms.

Un pic s'associa un XSD a un XML s'ha de veure quina estructura ha de tenir un XSD. En primer lloc apareixerà sempre, a més de la capçalera que indica que s'utilitza la versió de XML com als documents XML, es trobarà un element *schema* que contendrà a tots els altres:

xmlns:xs:	Indica que els elements i els tipus de dades utilitzades en l'esquema provenen del espai de noms que indiquem. També especifica que els elements i tipus de dades que provenen de l'URI del espai de noms ha de portar el prefix xs:. Es podria definir més d'un espai de noms al element schema.
targetNamespace	Indica que els elements definits per l'esquema provenen del espai de noms indicat.
xmlns	Indica que l'espai de noms per defecte.
elementFormDefault	Indica el tipus d'herència empleada (default o qualified) , que vol dir que opera mitjançant la utilització de prefixes associats als elements I atributs del document. En aquest cas tots hauran de dur el prefix xs: perquè així s'ha definit l'espai de nom abans(xmlns:xs

A partir d'aquí s'han d'anar definint tots els elements i atributs dels elements de l'esquema i que el document XML haurà de complir per a poder esser validat:

1. Definicions dels tipus dels elements (ElementType)

```
<ElementType name=NomElement content='textOnly'/>
```

A l'hora de definir els tipus dels elements es pot distingir entre tipus ja predefinits que són els més habituals, tipus simples i tipus complexes. En aquetes definicions en definim tan el tipus com el contingut dels elements especificats assignant-li un nom *NomElement*, així com de tots els subelements que pugui contenir i les possibles restriccions que pugin tenir els diferents valors inclosos. Es pot definir també tipus complexes d'elements mitjançant *Complextype*. Això permetrà assignar assenyalar la inclusió d'uns elements dintre d'uns altres i crear tipus més complets. Els tipus d'elements més comuns són: xs:string ,xs:decimal ,xs:integer ,xs:boolean ,xs:date i xs:time .

2. Definicions dels elements simples(Element)

En quant a elements o elements simples venen definits per l'estructura:

```
<xs:element name="xxx" type="yyy"/>
```

A on el valor del camp *name*, *xxx*, serà el nom que li donem al element i el valor del camp *type*, yyy, que serà el tipus del element. S'ha de tenir en compte que només es pot definir un element d'un tipus d'element que ja haguem definit Un pic definit pot aparèixer contingut dins

d'un altre element. Als elements també se'ls pot assignar un valor per defecte o fitxar-ne un que no es podrà canviar amb els atributs *default* i *fixed* respectivament

```
<xs:element name="Nomelement" type="tipuselement" default="valorperdefecte"/>
<xs:element name=" Nomelement" type="tipuselement" fixed="valorfitxe"/>
```

Es poden també definir elements complexes sense necessitat de definit un tipus nou d'element de forma implícita i directament si s'inclou dintre de l'etiqueta element.

3. Definicions dels tipus dels atributs (AttributeType).

S'assigna un tipus i unes condicions a l'atribut especificat en *NomAtribut*. Al igual que en el cas dels elements també es podrà fixar el valor o donar un valor per defecte i afegir totes les restriccions que considerin oportunes. Els tipus d'elements més comuns són: xs:string, xs:decimal, xs:integer, xs:boolean, xs:date, xs:time.

4. **Definicions dels atributs (Attribute**).

A la declaració d'un atribut, s'especifica el tipus de l'atribut del qual en es vol que formi part indicat pel camp *type*. Un cop definit pot aparèixer com atribut d'un element determinat.

Presentació XML

Anteriorment s'ha definit com a una característica dels documents XML, que la seva informació no te format i que només conté dades sense indicar com presentar-les. Per a poder fer-ho i presentar un document XML formatjat, s'ha d'informar sobre el format i la presentació utilitzant els anomenats fulls d'estil, que no són res més que un altre document on s'especifica el format que es vol aplicar. Encara que no siguin un objectiu directe d'aquest treball, a continuació es presentaran els mètodes dels quals es disposa actualment per realitzar aquesta transformació de dades quès són els fulls d'estil CSS i XSLT.

Fulls d'estil CSS

Són la forma més popular de donar format a arxius de HTML, encara que també es poden utilitzar per representar documents XML en qualsevol navegador web.

Per afegir un full d'estil a un XML i que permeti el seu processament com s'ha dit s'ha d'incloure una referència. Això farà que el navegador al visualitzar el contingut de l'XML, accedeixi a la URL de la fulla d'estil i la faci servir per processar el document. Bàsicament un fitxer CSS, és un fitxer de text pla en què especifiquem una llista de regles CSS amb la següent sintaxi:

```
selector {propiedad_1: valor_1;....; propiedad_n: valor_n;}
```

A on cada selector serà un element al qual se li aplicaran les propietats definides per a ell.

Existeixen una sèrie de desavantatges d'utilitzar CSS, ja que, per exemple, un fitxer CSS no té sintaxi XML i si s'estan utilitzant tecnologies XML no té sentit fer-ne servir un altra que no en sigui XML. També, i al igual que passava amb DTD i XML Schema, és una eina massa simple de

visualització ja que només permet utilitzar un navegador i a més a més no hi ha interacció amb el document XML. Simplement tracta de presentació. Aquests inconvenients van obligar a W3C a cercar una alternativa que és XSLT.

XSLT

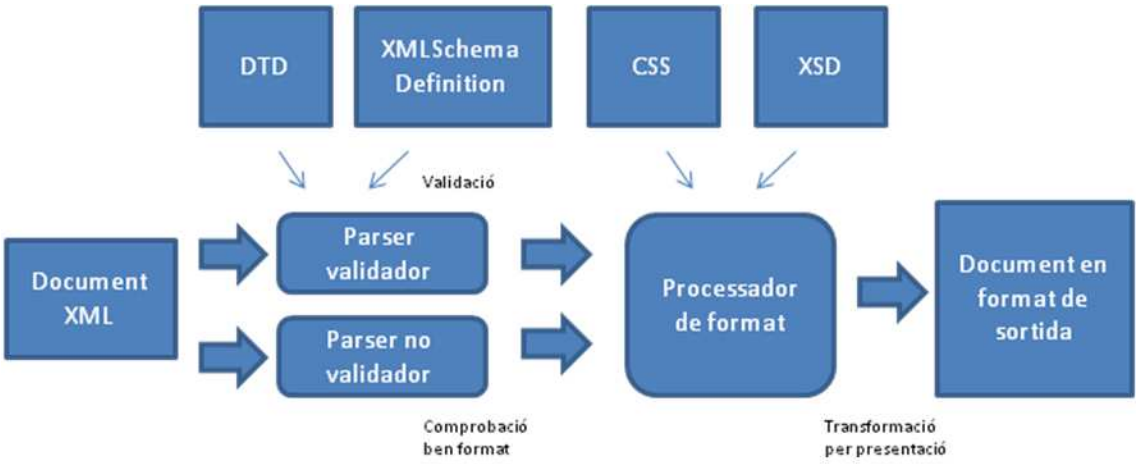
La alternativa creada per W3C és amb fitxers d'estils XSL. L'eXtensible StyleSheets Language és un llenguatge que permet definir una presentació o format per un document XML. Un mateix document XML pot tenir més d'un full d'estil XSL que el mostrar-lo en diferents formats (HTML, PDF, RTF, etc.) i, bàsicament, XSL és un llenguatge que defineix una transformació entre un document XML d'entrada, i un altre document de sortida i és a XML el que CSS és als documents HTML(XSLT):

XSL és un llenguatge XML amb unes etiquetes especifiques destinades a la presentació de les dades contingudes dins dels fitxers XML i quan el sistema operatiu executa un fitxer amb extensió *.xsl* ja s'invoca el parser de XSL que l'interpretarà per poder presentar-ho.

XSL no només permet especificar com es volen presentar les dades d'un document XML, sinó que també serveix per filtrar les dades d'acord amb certes condicions que haguem definit i possibilita l'execució de bucles, sentències del tipus condicionals, seleccions per comparació, operacions lògiques, ordenacions de dades, utilització de plantilles i altres qüestions similars.

Processament d'un XML

Per a fitxar conceptes i a mode de resum, en el següent esquema resum es pot veure tot el procés que segueix un XML alhora del seu processament contemplant les diferents opcions per a la seva validació com document ben format o vàlid així com el processament de la seva presentació i la seva visualització.



Per a visualitzar un document XML abans se l'ha de validar si està ben format i si és necessari validar-ho també(mitjançant un DTD o d'una definició d'esquema XML). Un pic validat se l'ha de presentar d'acord amb un CSS o unes transformacions XSLT associades en un fitxer .xsl.

Limitacions del XML amb les metadades

Les metadades, que es poden definir com a les dades sobre les dades, descriuen propietats i característiques sobre les dades. Aquestes descripcions poden esser de gran ajuda a l'hora de recuperar dades i per desgràcia XML presenta una sèrie de deficiències i problemes per a poder fer-ho de forma adequada.

Els problemes principals que té XML són la falta tan de flexibilitat i d'escalabilitat ja que, en la majoria de casos, l'ordre en que apareixen els elements a un document XML és significatiu i necessari cosa que no és lògica en el món de les metadades en que l'important és poder accedir-hi i no l'ordre de la seva presentació, ordre que pot provocar un gran cost de manteniment si el nombre d'elements és elevat. A més té el problema de que els arbres jeràrquics són difícils de combinar entre si i accedir a les dades pot esser problemàtic si existeixen vàries rutes per a poder accedir a una mateixa dada provocant una baixa escalabilitat en els seus accessos si l'arbre és gran. Per últim es té que les gramàtiques XML, que permeten validar les estructures de documents XML no permeten interpretar de forma general les dades que contenen i per a fer-ho és necessari implementar una aplicació que faci.

Per totes aquestes limitacions va aparèixer RDF que té un model de dades més flexible basat un triplets de subjecte+predicat+objecte en els que els seus elements estan identificats amb URIs que permeten l'accés a la seva semàntica i a la seva interpretació de les dades.

RDF

En aquest punt es presentarà, bàsicament, el llenguatge RDF(*Resource Description Framework*). S'explicaran d'aquest llenguatge el seu origen com tots els concepte s bàsics que han de permetre conèixer i comprendre tots els fonaments d'aquest llenguatge: la seva sintaxi, la seva estructura i model de dades que representa.

Orígens

RDF és una de les tecnologies claus en el món de la web semàntica i al igual que moltes d'altres, és una proposta del W3C per a definir i processar metadades en la web i està suportada per XML. El seu origen es remunta a l'agost de l'any 1997 en el que amb la finalitat de crear un format per aconseguir la compatibilitat entre els diferents sistemes de metadades existents, es va crear sota la supervisió del *World Wide Web Consotium*.

Per crear RDF es va comptar amb la influència i protecció d'una sèrie de comunitats, com a poden esser la comunitat de Normalització Web en forma de metadades, la de base de dades o la bibliotecària, entre d'altres, que de forma consensuada es varen posar d'acord des d'un bon principi en els principis bàsics sobre com havia d'esser la representació de les metadades .

Característiques del RDF

El llenguatge RDF, bàsicament, proporciona un tipus d'arquitectura genèrica per a organitzar les metadades i que facilitarà les possibilitats de crear mètodes automatitzats que millorin el nombre de descobriments de nous recursos a partir de les dades inferides i de poder indicar recursos d'una forma intel·ligent amb un grau de precisió superior del que havia fins al moment i es podria dir que el seu objectiu general és definir un mecanisme per a descriure recursos, sense que s'assumeixi cap regla o norma sobre un domini d'aplicació en particular per defecte, ni tampoc es defineixi prèviament la semàntica de cap domini d'aplicació.

La facilitat que posseeix RDF per processar metadades permet la interacció entre diverses aplicacions i proporciona un mecanisme perfecte d'intercanvi d'informació a través de la web, destacant també, gracies a la seva flexibilitat a l'hora d' utilitzar-se, en multitud d'àmbits del entorn web com a per exemple poden esser la recuperació de recursos per millorar els motors de recerca, la catalogació per a descriure contingut i les relacions a una plana web, facilitar l'intercanvi i compartir coneixement entre diferents agents de programari intel·ligents, qualificar el contingut per a la descripció de col·leccions de planes que representen un document lògic individual, la descripció dels drets de propietat intel·lectual de les planes web o el que un usuari pugui expressar les preferències de privacitat, així com les polítiques de seguretat d'un domini web.

S'ha parlat anteriorment sobre els espais de noms com a una part fonamental de la infraestructura web i que representen una part crítica de la infraestructura necessària per a desplegar sistemes de metadades modulars web fent possible la seva modularitat i extensibilitat. Es pot dir que qualsevol conjunt d'elements de metadades és un espai de noms limitat per les regles i convencions determinades per la seva pròpia agencia de manteniment i que les declaracions d'aquests espais de noms permetran als dissenyadors d'esquemes de metadades definir el context per a un terme en particular, assegurant que la definició sigui única per al terme dintre dels límits de l'espai de noms declarat.

Utilitzant aquesta infraestructura, els dissenyadors dels sistemes de metadades poden seleccionar els elements desitjats dins d'un conjunt d'elements de metadades convenientment establert, evitant així tornar a inventar conjunts de noves metadades per a cada nou domini.

El llenguatge RDF representa un model de metadades que dirigeix per referència molts dels aspectes de la codificació que requereix l'emmagatzematge i transferència de fitxers i existeixen molts d'aquests aspectes en que RDF utilitza XML; per això que molt sovint es diu que RDF i XML són complementaris ja que RDF és el model de metadada i solament dirigeix per referència molts dels aspectes de codificació que requereix el seu emmagatzemament i transferència d'arxius. Per aquests aspectes, RDF té com a suport el XML. És important també entendre doncs, que aquesta sintaxis en XML es solament una sintaxis possible per a que surtin formes alternatives per a representar el mateix model de dades RDF.

RDF es beneficia del XML de la seva flexibilitat a l'hora de generar nous conjunts d'etiquetes, de la seva orientació multiplataforma, i del mecanisme semàntic que proporciona per a expressar la descripció de qualsevol tipus de recurs, encara que s'ha de destacar, que aquesta

sintaxi XML és només una de les possibles a utilitzar dintre de RDF, i que actualment es pot optar per altres alternatives, com N3, o el seu subconjunt, les N-Triples, o bé, en un futur, poden sorgir altres formes per a representar el mateix model de dades RDF.

Resumint doncs, es pot dir que el llenguatge RDF especifica una gramàtica lògica que aconsegueix que els dissenyadors de planes web puguin descriure les propietats semàntiques dels seus documents en una notació estàndard i comú per qualsevol tipus de metadades, i que per poder-ho fer es basa en tres pilars fonamentals com les metadades, els espais de noms i un llenguatge que permeti proporcionar una estructura a les dades i documents de la web.

Als següents apartats s'explicaran els conceptes bàsics del model a utilitzar per a representar les metadades mitjançant RDF, així com una sintaxi per a codificar i transmetre-les maximitzant la interoperabilitat entre els servidors i clients web desenvolupats independentment.

El model RDF

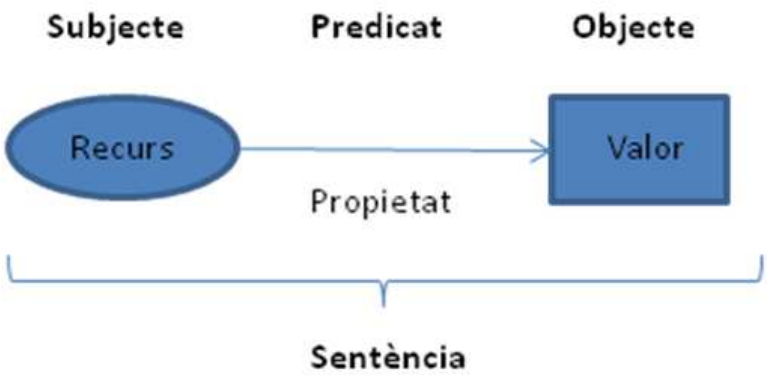
La base de RDF és la creació d'un model per a representar recursos, propietats i els valors corresponents a aquestes propietats, incloent-hi també, les relacions existents entre ells.

El model de dades del llenguatge RDF es basa en una nomenclatura neutral i independent que serveix per a representar expressions RDF.

Bàsicament el model de dades bàsic el constitueixen tres tipus conceptes:

Recursos (resources)	Totes les coses descrites per expressions RDF es denominen recursos. Llocs, persones, una plana web completa o una part o una col·lecció d'elles o objectes del món, real seran recursos. Cadascun d'ells que s'utilitzi ha d'estar identificat inequívocament mitjançant un URI.
Propietats (properties)	És qualsevol característica, atribut o relació utilitzada per descriure recursos. A cada cadascuna d'elles se li assigna un valor, defineix els valors permesos, es limita els tipus de recurs a descriure, així com se li determina les seves relacions amb altres propietats.
Sentències (statements)	Són una combinació formada per un recurs específic, una propietat determinada i el valor assignat a ella. A aquestes tres parts individuals d'una sentència se'ls anomena, respectivament, subjecte, predicat i objecte. El significat de les dades es sol expressar per a un conjunt d'aquetes sentències utilitzant marques XML.

Una sentència RDF es sol representar utilitzant: ovals per a representar els subjectes, rectangles per a representar els objectes i fletxes que van des del subjecte al objecte per a representar els predicats.



Representació del format d'una sentència RDF.

El valor d'una propietat pot ser tant un altre recurs, com un simple literal (qualsevol tipus de dada). El diagrama bàsic de sentència és podria llegir com que el "subjecte" té com "predicat" a "objecte>" ,o també: "objecte" és "predicat" del "subjecte".

Tots els gràfics segueixen un model comú: la descripció de nodes mitjançant les propietats i la seva personalització a través dels identificador o URI's. La transformació d'aquestes representacions a RDF es basa en un etiquetat dels URI's que permeti qualificar als nodes i a les fletxes de manera unívoca. La uniformitat forma part de les característiques més destacades del model de dades i el fet de tenir un l'únic vocabulari que s'utilitza sigui el format per URI's permet l'ús d'una mateixa URI com a node i com a fletxa i poder realitzar auto referències. Amb la combinació de les representacions descrites anteriorment, es poden representar pràcticament totes les estructures o models de dades que es desitgin.

Sintaxi RDF i RDF/XML

Existeix més d'un tipus de sintaxis RDF i actualment es poden utilitzar tan l'N3, o el seu subconjunt, les N-Triples i XML. La utilitzada més habitualment és una serialització de XML coneguda com RDF/XML que és la que es passarà a explicar a continuació.

La sintaxi RDF/XML va esser dissenyada per a poder agrupar més d'una sentència sobre un mateix recurs, ja que generalment una sentència RDF no apareix aïllada, sinó que forma o construeix un agrupament de propietats per a qualificar a un determinat recurs.

A continuació es pot veure un exemple de com es plasmarien a un document RDF dues entrades a una llista de CDs:

Títol	Artista	País	Companyia	Preu	Any
Empire Burlesque	Bob Dylan	USA	Columbia	10.90	1985
Hide your heart	Bonnie Tyler	UK	CBS	9.90	1988

```
<?xml version="1.0"?>

<rdf:RDF
xmlns:rdf="http://www.w3.org/1999/02/22-rdf-syntax-ns#"
xmlns:cd="http://www.recshop.fake/cd#">

<rdf:Description
rdf:about="http://www.recshop.fake/cd/Empire  Burlesque">
 <cd:artista>Bob Dylan</cd:artista>
 <cd:pais>USA</cd:pais>
 <cd:companyia>Columbia</cd:companyia>
 <cd:preu>10.90</cd:preu>
 <cd:any>1985</cd:any>
</rdf:Description>

<rdf:Description
rdf:about="http://www.recshop.fake/cd/Hide  your heart">
 <cd:artista>Bonnie Tyler</cd:artista>
 <cd:pais>UK</cd:pais>
 <cd:companyia>CBS Records</cd:companyia>
 <cd:preu>9.90</cd:preu>
 <cd:any>1988</cd:any>
</rdf:Description>
.
.
.
</rdf:RDF>
```

RDF a la seves especificacions defineix dos sintaxis XML per a codificar una instància o objecte específic d'una categoria del model de dades.

Sintaxi serialitzada

En primer lloc es te la sintaxi serialitzada que expressa les capacitats totals del model de dades d'una forma molt regular i estructurada. La sintaxi serialitzada basa en dues instruccions <RDF> .. </RDF> i <Description> .. </Description> . La primera s'utilitza per a indicar que es tracta d'un document RDF i la segona per a descriure cadascun dels recursos que formen el document RDF. El format d'aquestes instruccions és el següent:

Element rdf

Format:['<RDF>'] Description ['</RDF>']*

< RDF> és l'element arrel d'un document RDF seguint el format d'un document XML i conté també una referència a l'espai de noms RDF. RDF du terme la funció de contenidor que marca els límits al document XML, el contingut pot ser mapejat com una instància del model de dades RDF. Si el contingut pot considerar-se com a RDF des del context de l'aplicació podria obviar-se. Si s'analitza aquesta instrucció del exemple anterior:

```
<rdf:RDF
xmlns:rdf="http://www.w3.org/1999/02/22-rdf-syntax-ns#"
xmlns:cd="http://www.recshop.fake/cd#">
```

Es pot veure la definició dels espais de noms, i els seus corresponents prefixes, que s'utilitzaran en el document. En aquest cas els prefixes són "*rdf*" i "*cd*", que corresponen a document i esquema RDF respectivament.

Element Description

Format: <Description Atributabout> Propietats</Description>*

Descriu cadascun dels recursos que s'utilitzen al document considerant-se el lloc a on s'ha de realitzar la identificació d'un recurs descrit i proporcionar una forma de donar nom simultàniament a vàries sentències d'un recurs. A partir d'ells s'aniran definint i agrupant cadascuna de les propietats que han de qualificar a cadascun dels recursos en concret. Al tag *Description* del exemple pot tenir dos paràmetres que poden tenir orígens molt diversos que són:

***Atributabout* (*about*).**	Identificar el recurs.
Propietas:	A on es poden afegir i descriure les diferents propietats del recurs. Les propietats contingudes poden definir-se com atributs com al exemple anterior o com a recursos en que en lloc de tenir assignat un valor, té una referència a un recurs que conté informació sobre la propietat.

Sintaxi abreviada

El segon tipus de sintaxi, és la sintaxi abreujada que inclou termes addicionals per tal de proporcionar una forma més compacta de representar un subconjunt del model de dades a que, encara que la sintaxi serialitzada mostra l'estructura d'un model RDF d'una forma molt clara i fàcil d'entendre, normalment és millor utilitzar una forma XML més compacta i per això es farà servir la forma abreujada igual a l'anterior però aplicant tres tipus d'abreviacions:

1. Si no existeixen propietats repetides dintre d'un element Description i els valors de les propietats són literals, les propietats es poden expressar com atributs XML de l'element *Description*. En aquest cas:

```
<rdf:Description
rdf:about="http://www.recshop.fake/cd/Empire Burlesque">
 <cd:artista>Bob Dylan</cd:artista>
 <cd:pais>USA</cd:pais>
 <cd:companyia>Columbia</cd:companyia>
 <cd:preu>10.90</cd:preu>
 <cd:any>1985</cd:any>
</rdf:Description>
```

```
<Description
 about="http://www.recshop.fake/cd/Empire Burlesque"
 artista="Bob Dylan"
 pais="USA"
 companyia="Columbia"
 preu="10.90"
 any="1985" />
```

2. La segona forma es pot utilitzar amb sentències específiques on l'objecte de la declaració és un altre recurs, i el valor de qualsevol propietat donada per aquest segon recurs són

cadenes de text afecta als elements *Description*. En aquest cas, s'aplica una transformació on les propietats de l'element *Description* d'aquest segon recurs poden posar-se com atributs XML de l'element *propertyElt* del primer recurs especificat:

```
<rdf:Description about="http://..../Programa-1">
  <rdf:Arxius resource="http//.../Arxiu-1"/>
 <rdf:Description>

 <rdf:Description about="http://.../Arxiu-1">
  <rdf:Tipus>indexat</rdf:Tipus>
 </rdf:Description>
```

```
<rdf:Description about="http://.../Programa-1">
   <rdf:Arxius resource="http://.../Arxiu-1"
    Tipus="indexat"  />
 </rdf:Description>
```

3. El darrer tipus d'abreviació s'aplica només en el cas de que un element *Description* que contingui una propietat type. En aquest cas el tipus de recurs definit com a valor de la propietat type pot utilitzar-se directament com un nom d'element en lloc de l'element Description en què està inclòs.

```
<rdf:Description
rdf:about="http://www.recshop.fake/cd/Empire
Burlesque">
<type resource="http://.../schema/cd"/>
.
.
.
</rdf:Description>
```

```
<rdf:cd
rdf:about="http://.../schema/cd"/>
.
.
.
</rdf:cd>
```

El seu ús a un mateix documents és possible ja que els intèrprets RDF implementen les dues.

Contenidors

Els elements contenidors normalment s'utilitzen per a referir-se a una col·lecció de recursos en lloc de a un recurs en concret, per exemple, és més habitual treballar amb la relació de programes informàtics que integren un inventari, que no treballar amb un programa en concret. Per poder mantenir aquestes llistes de recursos o literals, l' RDF utilitza els anomenats contenidors. N'existeixen tres tipus:

1. **Bag(rdf:bag)**. És una llista desordenada de recursos o literals i s'utilitzen per indicar que una propietat té múltiples valors, es valor poden estar en qualsevol ordre i se n'admeten duplicats.

```
<?xml version="1.0"?>

<rdf:RDF
xmlns:rdf="http://www.w3.org/1999/02/22-rdf-syntax-ns#"
xmlns:cd="http://www.recshop.fake/cd#">

<rdf:Description
rdf:about="http://www.recshop.fake/cd/Beatles">
 <cd:artista>
  <rdf:Bag>
   <rdf:li>John</rdf:li>
   <rdf:li>Paul</rdf:li>
   <rdf:li>George</rdf:li>
   <rdf:li>Ringo</rdf:li>
  </rdf:Bag>
 </cd:artista>
</rdf:Description>

</rdf:RDF>
```

2. **Sequence(rdf:Seq)**. És una llista ordenada de recursos o literals que s'utilitza per a declarar que una propietat té múltiples valors però, al contrari que bag, l'ordre si és significatiu. També admet valors duplicats.

```
<?xml version="1.0"?>

<rdf:RDF
xmlns:rdf="http://www.w3.org/1999/02/22-rdf-syntax-
ns#"
xmlns:cd="http://www.recshop.fake/cd#">

<rdf:Description
rdf:about="http://www.recshop.fake/cd/Beatles">
 <cd:artista>
  <rdf:Seq>
   <rdf:li>George</rdf:li>
   <rdf:li>John</rdf:li>
   <rdf:li>Paul</rdf:li>
   <rdf:li>Ringo</rdf:li>
  </rdf:Seq>
 </cd:artista>
</rdf:Description>

</rdf:RDF>
```

3. **Alternative(rdf:Alt)**. És una llista de recursos o literals que representa les possibles alternatives per a un valor individual d'una propietat. L'aplicació que utilitzi una propietat on el valor és una col·lecció d'alternatives sap que pot elegir com a correcte qualsevol dels valors continguts a la llista. En el següent exemple, format, només podria prendre 1 dels valors inclosos.

```
<?xml version="1.0"?>

<rdf:RDF
xmlns:rdf="http://www.w3.org/1999/02/22-rdf-syntax-
ns#"
xmlns:cd="http://www.recshop.fake/cd#">

<rdf:Description
rdf:about="http://www.recshop.fake/cd/Beatles">
 <cd:format>
  <rdf:Alt>
   <rdf:li>CD</rdf:li>
   <rdf:li>Record</rdf:li>
   <rdf:li>Tape</rdf:li>
  </rdf:Alt>
 </cd:format>
</rdf:Description>

</rdf:RDF>
```

Esquema RDF

Com s'ha explicat, el model de dades RDF defineix un model simple per descriure els recursos i las relacions entre ells mitjançant la utilització de propietats i els seus valors assignats. Aquestes propietats es podrien entendre com atributs del recurs corresponent-se en aquest sentit al típic parell atribut-valor. A part d'això, les propietats RDF també poden representar relacions entre recursos i amb aquesta estructura, el model de dades RDF es semblaria a un diagrama entitat-relació, encara que no proporciona mecanismes per a declarar aquestes propietats, ni tampoc per a definir les relacions entre les propietats i altres recursos.

Per a poder dur a terme aquesta funció o poder definir les classes específiques de l'aplicació i les propietats s'utilitzarà una extensió del RDF com és l'Esquema RDF (RDFS).

Un esquema RDF no defineix les propietats sinó que proporciona un marc de treball per a poder fer-ho i definir-ne totes les propietats d'un recurs determinat i tots els tipus de recursos que es descriuran.

Els esquemes RDF podrien equiparar-se doncs als DTD's o als Esquemes XML, però amb la diferència que permeten especificar restriccions en l'estructura d'un document XML i proporcionen informació sobre la interpretació que s'ha de donar a una sentència en un model de dades concret i, a l'igual que l'RDF, també utilitzen l'XML per a la seva codificació d'intercanvi i per això es podran definir amb ell relacions jeràrquiques de classes de recursos, especificant les propietats i relacions que s'admeten entre elles. En RDF les classes, relacions i les pròpies sentencies són també recursos, i per tant, es poden tractar com a tals.

Estructura d'un esquema RDF.

Com s'ha esmentat anteriorment, un esquema RDF és una col·lecció de recursos RDF que es poden utilitzar per descriure les propietats d'altres recursos RDF i que permeten definir vocabularis RDF per aplicacions específiques aquest RDFS seguirà una estructura definida i per

definir el nucli del vocabulari utilitzat en un document RDF, dins el propi document RDF s'indicarà l'espai de noms i identificant-ho amb un URI. Per exemple a:

```
<rdf:RDF
xmlns:rdf="http://www.w3.org/1999/02/22-rdf-syntax-ns#"
xmlns:rdfs="http://www.w3.org/2000/01/rdf-schema#"
xml:base=" http://www.recshop.fake/cd#">
```

Aquesta URI serà "http://www.recshop.fake/cd" i a més es pot veure en aquesta especificació que també es defineix el prefix *rdf* per a referir-se a l'espai de noms principal RDF : *http://www.w3.org/1999/02/22-rdf-syntax-ns#*

Com s'ha descrit abans en l'especificació del model i la sintaxi, els recursos poden ser instàncies d'una o més classes, i això s'indicava amb la propietat *type*. A més, s'ha vist que, en ocasions, les classes es poden organitzar de forma jeràrquica donant lloc a classes filles o subclasses *subClassOf*.

El sistema d'escriptura dels esquemes RDF es bastant paregut al dels llenguatges de programació orientada a objectes, però RDF es diferencia d'ells en què en lloc de definir una classe en termes de propietats, utilitza l'esquema RDF per definir les propietats en termes de classes dels recursos als que s'apliquen i ho fa a través de les restriccions *domain* i de *range*.

Classes

Les classes que formen directament com part del vocabulari en un esquema RDF, i que cada model RDF que es crea sobre l'espai de noms de l'esquema RDF inclou implícitament, són:

Resource	Totes les coses descrites per expressions RDF s'anomenen recursos i es poden considerar instàncies de la classe Resource.
Property	Representa el subconjunt de recursos RDF que són propietats.
Class	De forma pareguda als llenguatges orientats a objectes, es correspon amb el concepte genèric d'un tipus (*Type*) o categoria (*Category*. En el moment que a un esquema es defineix una nova classe, el recurs que representa aquesta classe ha de tenir una propietat type, el valor de la qual és el recurs Class.

Propietats

Són objectes específics d'una categoria de la classe *Property* i proporcionen un mecanisme per a expressar les relacions entre les diferents classes. Són:

type	Indica que un recurs és membre d'una classe i, per tant, té totes les característiques previstes per a un membre d'aquesta classe.
subClassOf	Especifica una relació pare-fill entre classes.
subPropertyOf	És una instància de *Property* que s'utilitza per a especificar que una propietat és una especialització d'una altra.

seeAlsO	Especifica un recurs que podria proporcionar informació addicional sobre el subjecte.
isDefinedBy	És una subpropietat de seeAlso, i indica el recurs que defineix el subjecte.

Restriccions

A esquema RDF les restriccions s'utilitzen bàsicament per a poder descriure les limitacions de les classes i propietats. Aquestes restriccions no indiquen però si, ni com, una aplicació pot processar la informació restringida.

La nomenclatura de les restriccions més habituals queda recollida a la següent:

ConstraintResource	Defineix una subclasse de *Resource*, les instàncies de la qual són construccions de l'esquema RDF implicades en l'expressió de les restriccions.
ConstraintProperty	Defineix una subclasse de *Property* a on totes les instàncies són propietats utilitzades per especificar restriccions.
range	Les restriccions de *rang* només s'apliquen a les propietats a la qual se li defineixen un rang de valor que ha de complir. Només pot haver 1 restricció de rang.
domain	S'utilitza per indicar la classe, els membres de la qual poden utilitzar una propietat. Una propietat pot tenir cap, una, o més d'una classe com a domini. Si no existeix restricció de domini, pot utilitzar-se amb qualsevol recurs.

Limitacions del Esquema RDF

RDFS no té capacitat per a descriure una semàntica de conceptes i relacions més allà de les li proveu els mecanismes de l'herència, això fa que sigui un llenguatge molt dèbil encara per al més senzill dels sistemes basats en coneixement. RDFS disposa solament de les primitives bàsiques per definir ontologies.

En contrast amb els típics llenguatges de representació de coneixement, RDFS no ha estat concebut per ser una resposta definitiva en la representació del coneixement en un domini particular sinó un nucli que pot ser extés i encara que RDFS té suport per al tractament de conceptes ontològics i relacions, no en té per als axiomes que són un ingredient clau en la definició d'una ontologia.

Es pot dir doncs que RDFS es pot utilitzat directament per descriure una ontologia, aclarint abans que per culpa d'haver esser dissenyat com a estàndard web, no disposa de primitives suficients com per resultar un llenguatge ontològic que sigui acceptable expressivament parlant sinó que és un nucli concebut per ser ampliat com es veurà al capítol següent.

OWL

En aquest capítol s'explicaran els conceptes bàsics del llenguatge OWL i s'explicarà el que ha suposat la seva a aparició per al desenvolupament de la web semàntica i la seva gran aportació per a la representació d'ontologies.

Origen i característiques

El Ontology Web Language (OWL) va aparèixer degut a que, per les limitacions d'expressivitat que tenen RDF/RDFS, van començar a definir-se nous llenguatges ontològics menys limitats. Entre aquest llenguatges es troben SHOE, DAML-ONT, OIL o DAML+OIL, aquest últim fou el precursor de l'actual OWL. OWL té també el seu origen en una recomanació del W3C (*World Wide Web Consortium*) i el seu naixement data del 10 de febrer de 2004.

OWL va esser ideat per a la definició d'ontologies estructurades al web i es va pensar directament entre diverses comunitats per a poder oferir una integració i interoperabilitat de les dades molt més complerta que les existents fins al moment, ja que quan es va definir es va fer de forma que fos compatible amb l'arquitectura general existent de la Word Wide Web i, en particular, de la Web semàntica.

OWL està pensat per ser utilitzat quan la informació continguda en els documents necessita ser processada per aplicacions i pot ser utilitzat per representar explícitament el significat de termes en vocabularis i les seves interrelacions, és a dir ontologies, té també major capacitat per expressar significat i semàntica que XML, RDF, i RDFS.

EL llenguatge OWL es construeix sobre el RDF i RDFS i encara que es basa en els ja coneguts URIs per a definir els noms dels recursos, afegeix més vocabulari per a descriure propietats i classes. Entre d'altres:

- Relacions entre classes(inclusió, equivalències)
- Cardinalitat(per exemple, només un)
- Igualtat i desigualtat entre les classes o les instàncies.
- Nous tipus de propietats
- Característiques per a les propietats
- Classes enumerades.

Els tres subllenguatges de OWL

Aquestes noves característiques fan que sigui molt més adequat per a poder representar ontologies que no RDF/RDFS ja que milloren molt el seu grau d'expressivitat però suposen també una sèrie de problemes, per exemple, limiten la capacitat dels raonadors per a inferir nou coneixement no garantint que el raonador pugui trobar totes les conclusions vàlides (completitud computacional), ni que trobi la resposta adequada per a qualsevol entrada en un temps finit(decidibilitat computacional). Per això es van crear tres subllenguatges que són variants a partit de OWL i cadascuna d'elles presenta un diferent grau en la seva relació

expressivitat/decidibilitat i cadascun d'ells està pensat per esser utilitzat per diverses comunitats de desenvolupadors i usuaris.

De menor a major capacitat expressivitat, que implica de major a menor eficiència de raonament es tenen: OWL Lite, OWL DL i OWL Full. Cadascun d'aquests subllenguatges és una extensió del seu predecessor i garanteix el tractament de qualsevol ontologia o estructura creada correctament per ell(però no el seu invers).

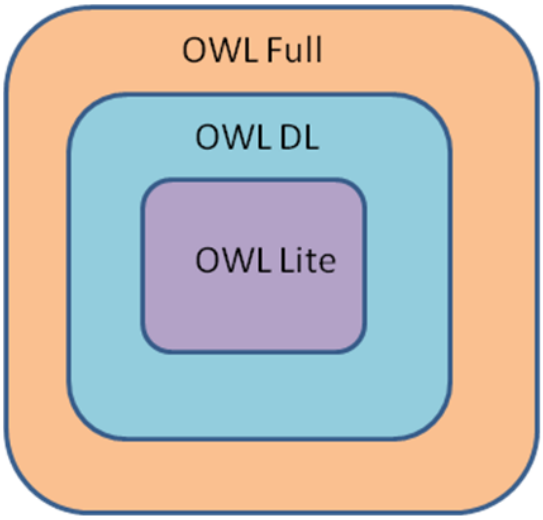


Diagrama d'inclusió dels tres subllenguatges

OWL Lite

Permet definir i declarar estructures jeràrquiques de classificació en les que es poden aplicar restriccions simples. És el que ofereix una complexitat més baixa i el més fàcil d'entendre i d'implementar sacrificant poder d'expressivitat. Està pensat per a proporcionar un suport bàsic als usuaris.

OWL DL

OWL DL o OWL de Lògica Descriptiva, fa possible una major expressivitat però garantint la seva completitud i la seva decidibilitat computacional, problemes que s'ha explicat que eren les limitacions de OWL. Això ho pot fer incorporant tots els constructors de OWL però només permet utilitzar-los baix certes restriccions que afecten sobretot, a la separació de tipus. Per exemple, no admet que un element individual d'una classe sigui al mateix temps una altra classe, o que una propietat sigui al mateix temps una classe.

OWL Full

És la versió més complexa de les que presenten aquests tres subllenguatges, i permet la màxima expressivitat amb la llibertat sintàctica que ofereix RDF però a canvi no ofereix cap garantia en termes de computabilitat. OWL Full permet augmentar el significat del vocabulari definit en una ontologia, provocant que no tots els agents de software siguin capaços de garantir el tractament de totes les funcionalitats d'aquest subllenguatge.

Es pot dir que OWL Full es comparable a una extensió de l'RDF, mentre que OWL Lite i OWL DL equivaldrien més a extensions d'un model RDF restringit. Tots tres creen documents en format

RDF, i cada document RDF és un document OWL Full, però només alguns documents RDF seran documents OWL Lite o OWL DL vàlids.

Estructura del OWL

Per a explicar l'estructura bàsica d'un document OWL, que definirà l'ontologia, es farà servir el més simple de tots ells, és a dir, el llenguatge OWL Lite i a continuació s'explicaran les característiques afegides per OWL DL i OWL FULL. En aquest capítol s'intentaran introduir tots els elements essencials que formen part d'una ontologia OWL com a són les classes, propietats, instàncies de classes i relacions entre aquestes instàncies.

L'estructura d'un document OWL estarà formada per una capçalera, que pot servir per a poder identificar l'ontologia, i una sèrie de classes, propietats i restriccions , que són las que serviran per a declarar-la. No s'estableix cap ordre en la seva aparició en el document deixant l'estructura final del mateix a criteri el desenvolupador.

Capçalera

Tot document OWL pot contenir una capçalera, però també pot no tenir-ne cap o tenir-ne més d'una. Dintre de cadascuna d'elles s'hi defineix la versió del llenguatge utilitzat i els elements que s'importen, indicant si es vol en ella:

- **La versió de OWL**: Marcada per la instrucció <*versionInfo*>, està formada per una cadena de text que representa la versió utilitzada de OWL.
- **Comentaris:** Si es volgués escriure cap comentari s'utilitzaria la instrucció <*comment*> permet introduir un comentari RDF.
- **Imports**: Que fan referència a una o més ontologies que contenen definicions que s'utilitzen en l'ontologia que s'està definit. Cadascuna d'aquestes ontologies està identificada per un URI,que especifica la ubicació des de on s'importa .

Classes

Dintre de OWL són bàsiques les classes ja que proporcionen un mecanisme d'abstracció per a poder agrupar recursos que tinguin característiques similars i com passava a RDF, tota classe OWL té associat un conjunt d'elements individuals que en formen l'extensió. En OWL Lite i en OWL DL, cadascun d'aquests elements no pot ser al mateix temps una altra classe però en OWL Full no s'aplica aquesta restricció.

Segons OWL es pot descriure correctament una classe de diferents formes :

- Amb un identificador o URI.
- Amb una enumeració de paràmetres individuals que formen la instància de la classe.
- Amb restriccions. Que permeten filtrar les condicions que han de complir els elements individuals de la classe.
- Amb la intersecció de dos o més descripcions de classes ja existents.
- Amb la unió de dos o més descripcions de classes ja existents.
- Amb el complement d'una descripció de classe. Que permet descriure una classe amb els elements individuals que no formen part d'una altra classe ja existent.

Per defecte, en tot document OWL existeixen sempre dues classes ja definides implícitament:

Thing	El conjunt d'elements individuals conforma la classe *Thing* per la qual cosa es dedueix que totes les classes que es defineixin seran subclasses de *Thing*
Nothing	És el conjunt buit, i és una subclasse de totes les classes que es poden anar definint

Per a la definició de les classes s'utilitzen les instruccions: Class i *subClassOf*. Amb aquestes dues estructures es poden definir i encadenar de forma jeràrquica totes les classes que formen una ontologia.

Propietats

Les propietats permetran declarar els aspectes generals que han de complir cadascun dels membres individuals que conformen una classe i especificar-ne, si es vol, característiques concretes. En OWL existeixen dos tipus de propietats:

DataType	Representa una relació existent entre les instàncies d'una classe i els literals RDF i esquemes XML. Instrucció *ObjectProperty*
Object	Representa una relació existent entre les instàncies de dues classes. Instrucció *DatatypeProperty*

Com passava a les classes, tota propietat es pot definir una especialització d'una altra ja existent amb la instrucció *subPropertyOf* durant la seva declaració pot restringir-se-li l'abast de la seva relació en quan al domini i al rang a través de les instruccions *domain* o *range* .

Restriccions a propietats

Es possible que mitjançant restriccions delimitar el rang de qualsevol. Per a definir una restricció es fa través de la instrucció *Restriction*, que indica el tipus de restricció a efectuar, i la instrucció *onProperty*, que delimita la propietat a la que s'efectuarà la restricció. Les restriccions mes importants es poden veure en aquesta taula:

AllValuesFrom	Indica una restricció a totes les instàncies de la classe que contenen la propietat especificada.
SomeValuesFrom	Indica que almenys un valor de la propietat indicada ha de complir.
cardinality	Indica exactament el nombre d'elements que formen part d'una relació. Es pot complementar amb la combinació de les instruccions *maxCardinality* i *minCardinality*, que permeten definir el valor més alt i més baix que admet la restricció cardinality.
hasValue	Indica que la classe està basada en la existència d'uns determinats valors de les propietats, és a dir, un membre individual formarà part de la classe, si almenys, una de les seves propietats és igual a l'especificada a la restricció.

Relacions entre classes i/o propietats.

A més de totes les estructures vistes anteriorment, existeixen encara una sèrie d'estructures a OWL que fan referència a la relació i combinació de classes i propietats a i entre diferents ontologies. Les més importants es poden trobar en aquesta taula:

equivalentClass	Indica s'indica que una determinada classe d'una ontologia, és equivalent a una altra classe d'una segona ontologia.
equivalentProperty	Indica que una determinada propietat d'una ontologia és equivalent a una altra propietat d'una segona ontologia.
sameAs	Indica que dos membres individuals de dues ontologies diferents són iguals.
differentFrom	Indica que diversos membres individuals de dues ontologies, són diferents.
allDifferent	Indica que tots els membres individuals d'una determinada ontologia, són diferents de tots els membres d'una altra ontologia.
intersectionOf	Indica que una determinada classe està formada pels membres que pertanyen a la intersecció dels membres individuals de les classes especificades.
unionOf	Indica que una determinada classe està formada per la unió dels membres de les classes especificades.
complementOf	Indica que una determinada classe està formada per tot el conjunt de membres individuals que no pertanyen a la classe especificada.
oneOf	Indica la relació d'elements individuals que formaran la classe.
disjointWith	Indica i garanteix que un membre individual d'una classe no sigui al mateix temps, una classe i d'un altre.

Llenguatges d'interrogació d'ontologies

Existeixen una sèrie de llenguatges es poden utilitzar per a interrogar ontologies i intentar obtenir com a resultats llistats d'individus que compleixin les restriccions plantejades a través de les preguntes realitzades. Encara que existeixen diversos llenguatges per a realitzar aquestes consultes essent més adient la utilització d'un o d'un altre depenent també de les característiques de l'ontologia i de com estigui definida en aquest capítol només s'explicarà el SPARQL, que és el que ha estat el utilitzat en les proves i test realitzats al projecte.

SPARQL

SPARQL és l'acrònim de l'anglès SPARQL Protocol and RDF Query Language i és un llenguatge estandarditzat per a la consulta de grafs RDF. Va esser normalitzat pel RDF Data Access Working Group (Dawg) del W3C. És una tecnologia clau en el desenvolupament de la Web Semàntica que es va constituir com recomanació oficial del W3C al 15 de gener de 2008.

Igual que succeeix amb SQL, s'ha de distingir entre el llenguatge de consulta i el motor per a l'emmagatzematge i recuperació de les dades. Per aquest motiu, hi ha múltiples

implementacions de SPARQL, generalment lligats a entorns de desenvolupament i plataforma tecnològiques.

Al principi SPARQL únicament incorpora funcions per a la recuperació sentències RDF. No obstant això, algunes propostes també inclouen operacions per al manteniment (creació, modificació i esborrat) de dades. SPARQL especifica quatre variacions de consulta :

SELECT	S'utilitza per extreure els valors en brut d'un extrem de SPARQL, els resultats es retornen en un format de taula.
CONSTRUCT	S'utilitza per extreure la informació des del punt final SPARQL i transformar els resultats en RDF vàlids
ASK	S'utilitza per a proporcionar un simple Verdader / Fals resultat d'una consulta en un extrem de SPARQL.
DESCRIBE	S'utilitza per extreure un graf RDF des del punt final SPARQL, el contingut es deixa al criteri de valoració per decidir sobre la base del que el responsable consideri com informació útil.

Cadascuna d'aquestes formes de consulta conté un bloc WHERE per a restringir la consulta, encara que en el cas de la consulta DESCRIBE el WHERE és opcional.

Un exemple de consulta SPARQL seria la pregunta "Hi ha qualque àudio en català?

```
SELECT ?Nom ?Ext
WHERE {
?id rdf:type POIUOC:Idioma .
?id POIUOC:hasIdiomaDocument ?doc.
?doc rdf:type POIUOC:DocumentAudio.
?doc POIUOC:hasDocumentNom  ?Nom.
?doc POIUOC:hasDocumentExtensio  ?Ext.
?id POIUOC:hasIdiomaNom "Català"^^xsd:string.
}
```

Les variables s'expressen amb un ? i per a realitzar consultes concises. SPARQL permet també la definició de prefixos i URIs base.

Tampoc s'entrarà amb més detall en l'explicació de com s'utilitza aquest llenguatge ja que es podrà veure el ús de forma més gràfica al capítol de proves sobre l'ontologia creada i es recullen totes les fonts utilitzades a la bibliografia del treball.

Eines empleades per a definir ontologies

Existeixen una sèrie d'eines dotades d'entorns gràfics que permeten la codificació i construcció d'una determinada ontologia de forma senzilla. Aquestes eines permeten definir l'estructura base a la que es classificarà una determinada informació facilitant molt la tasca de definició i la de desenvolupament distribuït d'ontologies així com la unió d'ontologies diferents. Entre elles es poden destacar algunes com *WebOnto*, *OILed*, *Ontolingua*, o *WebODE*, però l'eina per a la

construcció d'ontologies més utilitzada en l'actualitat és Protégé i d'aquesta eina es parlarà al llarg del pròxim capítol ja que serà la que es faci servir per a desenvolupar aquest projecte.

PROTÉGÉ

Protégé és un eina desenvolupada pel *Stanford Medical Informatics* de la Universitat de Stanford en un grup dirigit per *Mark Munsen*. Protégé és un marc de treball, una plataforma flexible i de fàcil configuració que facilita al desenvolupador realitzar la seva tasca i poder crear aplicacions i components basats en models. L'arquitectura de Protégé és oberta i el seu codi font es gratuït amb llicencia de codi obert Mozilla Public License(*MPL*), això possibilita afegir-hi fàcilment noves extensions que augmentin les seves funcionalitats.

El nucli central de l'aplicació, conegut com a Protégé-Frames, permet treballar amb sistemes basats en marcs (frames), en què els conceptes que són objectes es representen mitjançant classes, les propietats de les classes com a camps (slots), i les restriccions sobre les propietats o relacions entre classes i/o slots pel que rep el nom de facetes del camp (*slot faces*). Així doncs, amb ella, es poden crear classes i jerarquies, declarar propietats sobre les classes, crear instàncies i introduir valors, tot això en un entorn de menús, botons, quadres de diàleg i representacions gràfiques fàcils d'usar, possibilitant el desenvolupament en RDF i OWL de manera transparent. A més, Protégé disposa també d'un editor OWL que proporciona les utilitats necessàries per a editar, visualitzar, analitzar i consultar ontologies OWL.

Resumint, les tasques principals que poden desenvolupar-se a través de Protégé són tres:

- Desenvolupament d'ontologies
- Optimització de dades d'entrada de formularis
- Inserció de dades

Per a crear una ontologia amb Protégé s'han de seguir una sèrie de passes:

1. Crear un projecte: El que primer s'ha de fer al inicialitzar Protégé és crear un projecte per a poder treballar. Aquest projecte es guardarà amb extensió .pprj.
2. Crear les classes i les subclasses: Si van creant classes, per defecte, aniran subordinades a la classe :THING, que, com ja s'ha explicat al capítol anterior, totes les classes són subclasses d'aquesta si no es diu el contrari. Per a crear una subclasse, només s'ha de seleccionar una classe de les creades i la classe nova, es crearà com a subclasse d'ella.
3. Crear els atributs de les classes: Seleccionant una classe, fàcilment es poden introduir els atributs a través d'un formulari.
4. Crear les instàncies: Les dades reals del coneixement base. És convenient no introduir-ne fins a no tenir totalment creada l'estructura de classes i atributs.

Un pic definida l'ontologia s'emmagatzema en una estructura de directoris que s'ha d'haver definit anteriorment amb fitxers d'extensió .rdf i .rdfs, encara que pot esser exportada a altres

formats. En aquest projecte s'utilitzarà un .owl per emmagatzemar-la i que anirà associat amb el mateix nom del fitxer .pprj que és el del projecte particular encara que l'ontologia pot carregar-se al programa tan sols amb el fitxer .owl.

Protégé és un entorn obert i fàcil d'estendre, que ha generat al seu voltant tota una comunitat que contribueix activitats ment a ampliar l'entorn amb tot tipus de contribucions en forma de plug-ins, fent d'aquesta eina un entorn summament potent.

Resumint-ho tot, es pot dir que Protégé és una eina de desenvolupament d'ontologies que permet la construcció d'ontologies OWL tenint en compte totes les característiques del domini i que gràcies a ell elaboració de propietats i restriccions permet generar inferència sobre el domini, generant conceptualització i coneixement d'aquest.

RacerPro

RacerPro (*Rename Abox and Concept Expression Reason Professional*) és un producte comercial, que integra un conjunt d'eines, encara que per a desenvolupar aquest projecte s'ha demanat un tipus de llicència universitària temporal a través de la seva pàgina web.

RacerPro s'utilitza per manejar les ontologies basades en OWL com a motor de raonament per als editors d'ontologies. En concret, RacerPro és el raonador per defecte de Protégé, que s'ha de configurar per indicar que s'està utilitzant RacerPro es pot utilitzar bàsicament per:

- Verificació de consistència: Permet controlar si existeix consistència entre les propietats d'objectes i restriccions assignades a les classes.
- Classificació taxonòmica: Troba la jerarquia d'inferència de l'ontologia basada en la jerarquia de asserció, la qual és la classificació que s'obté en la construcció de l'ontologia. Generarà una jerarquia d'inferència en l'ontologia.
- Computació de tipus inferits: Permet computar els tipus d'objectes dins de l'ontologia.

En resum, RacerPro permet realitzar de forma ràpida la verificació d'una ontologia tenint en compte elements de consistència, a més de realitzar la classificació taxonòmica anomenada jerarquia d'inferència, derivada del model construït en l'ontologia anomenat jerarquia de asserció. Amb base a aquests elements es fa comprensible la importància de l'ontologia dins del concepte de la web semàntica.

OWLViz i Graphviz

OWLViz és una expansió de Protégé que permet representar de forma gràfica una ontologia i s'ha d'instal·lar a posteriori. Per a poder-se utilitzar, cal instal·lar abans un altre eina que permet realitzar components gràfics de l'ontologia anomenada Graphviz. Un cop instal·lat OWLViz i Graphviz, s'ha de seleccionar la ruta de Graphviz. Permet generar l'ontologia en forma gràfica amb la jerarquia de asserció o la jerarquia d'inferència. Serà molt útil alhora d'explicar i tenir una perspectiva més clara de l'ontologia

OntoViz

Igual que OWLViz, OntoViz és una eina que permet la representació gràfica de l'ontologia. No obstant això, OWLViz representa només les classes de l'ontologia mentre que OntoViz representa les classes i els individus o instàncies de l'ontologia. OntoViz s'instal·la com una

expansió encara que ja ve inclòs en la última versió de Protégé(4.1) i al igual que OWLViz, serà molt útil per a obtenir una visió global i representativa d'una ontologia.

Estat de les arts a les ontologies

A l'àmbit del turisme s'han intentat anar desenvolupant diferents repositoris de dades i taxonomies per intentar facilitar a les empreses la gestió de la informació turística, però el seu ús sempre ha estat intern i sense implementació de cap tipus d'estàndard o normalització. Darrerament s'han realitzat vertaders esforços de generació d'estàndards més globals per a facilitar l'organització i l'intercanvi de dades entre aquestes empreses com són el tesaurus de turisme i oci desenvolupat per l'Organització Mundial del Turisme, que s'explicarà més endavant.

No ha estat fins a les dues darreres dècades i coincidint amb l'anomenat boom del turisme, i amb els inicis de l'aplicació de les ontologies al domini turístic que s'ha donat un important impuls l'estandardització de les dades que ha provocat el desenvolupant diferents ontologies turístiques, públiques i privades. Aconseguint que algunes d'elles s'hagin consolidat com a estàndards assolint unes grans quotes d'acceptació i utilització a la representació tant d'aspectes genèrics de l'àmbit del turisme, com d'àmbits més específics per a descriuen escenaris més concrets.

En aquest capítol es farà un repàs d'algunes de les ontologies especialitzades més importants en l'àmbit turístic de les quals es pot trobar informació ara mateix repassant també els tesaurus i especificacions que varen donar lloc a moltes d'elles.

Tesaurus i altres especificacions

Existeixen una sèrie d'especificacions que no poden considerar-se com ontologies en el sentit estricte de la paraula, però que han servit de base per al desenvolupament de moltes de les ontologies és conegudes com són:

Thesaurus on Tourism and Leisure Activities (UNWTO)

Aquest "*Thesaurus on Tourism and Leisure Activities*" és el fruit de més de 20 anys de feina intentant desenvolupar un llenguatge documentació específica per ajudar a la recerca d'informació relativa a les activitats turístiques i ha estat desenvolupat per la l'Organització Mundial de Turisme (UNWTO).

Pot ser utilitzat com a una guia per al turisme, així com per l'estandardització i normalització de la indexació i del idioma comú de recerca a nivell internacional. Conté termes molt específics amb el turisme han estat àmpliament definides per a les persones que no estan familiaritzats amb aquest vocabulari i pot utilitzar el diccionari de sinònims.

La UNWTO va crear un tesaurus de conceptes turístics i activitats d'oci per tal d'ajudar a la classificació d'informació relacionada amb les activitats i llocs turístics. Aquest diccionari pot

ser utilitzat com a guia terminològica relacionada amb el turisme tractant de normalitzar els termes per aconseguir una classificació estàndard de qualsevol document de tipus turístic.

El Diccionari està traduït al anglès, al francès i al espanyol, i consta de 1.800 descriptors a cadascun d'ells, essent la primera aproximació multilingüe en aquesta àrea que intenta evitar la simple traducció de conceptes i expressions, per obtenir un veritable diccionari turístic real per als tres idiomes. Aquest tesaurus ha servit de base per al desenvolupament de les principals ontologies existents en l'actualitat, com són Harmonise o DERI e-Tourism.

La OpenTravel Alliance (OTA)

La OTA és una organització formada per organitzacions que representen a tots els segments de la indústria del turisme (línies aèries, cadenes hoteleres i d'empreses de lloguer de cotxes, etc.), al costat de proveïdors de tecnologia i serveis.

La seva funció és la de facilitar la interoperabilitat en el desenvolupament de programari amb sistemes d'altres companyies de l'àmbit turístic desenvolupant i mantenint una biblioteca d'esquemes XML per al seu ús en la indústria del turisme, que conformen l'especificació OpenTravel XML. OpenTravel aglutina més de 200 documents XML agrupats en paquets que defineixen conceptes i entitats del sector turístic, i que permeten normalitzar el format dels missatges utilitzats a la comunicació entre sistemes de diferents companyies per a facilitar-ne el seu ús i reduir temps i costos.

Mondeca Tourism Ontology

Parlant pròpiament d'ontologies, una de les que s'ha de parlar és la desenvolupada per la companyia Mondeca que va desenvolupar l'ontologia turística desenvolupada per introdueix una sèrie de conceptes que resulten fonamentals en l'àmbit del turisme ja que questa ontologia inclou conceptes turístics del tesaurus definit per l'Organització Mundial del Turisme (UNWTO), descrivint les diferents dimensions utilitzades per a la determinació dels perfils del concepte turisme, els objectes turístics i culturals (lloc, museu, restaurant, allotjament, transport, esdeveniment), paquets turístics i contingut multimèdia turístic.

Mondeca té una sòlida experiència en el modelatge de l'ontologia i la gestió de l'ontologia en l'àmbit de la indústria, els mitjans de comunicació, autoritats del govern locals per gestionar temes ben organitzats i el contingut a la llar. L'ontologia conté al voltant de 1000 conceptes i està descrita en OWL. És d'àmbit privat i no es permet la seva utilització lliure. Mondeca segueix essent la companyia que ara mateix és líder en tecnologies relacionades amb la Web Semàntica, proporcionant solucions per a la creació de taxonomies, gestió d'ontologies, etc i

compta com a clients destacats, per exemple, a la PSA, l'EADS, Hachette Filipacchi Media, el departament francès de Policia, Ministeri de Turisme americà, Lexis Nexis, o Wolter Kluwers.

A més també és un membre actiu del W3C contribuir als requeriments de OWL estàndard, una de les empreses fundadores darrere dels estàndards XML Tema Mapes i va participar en diversos projectes europeus de recerca com poden ser Hi-Touch(ontologia que s'explicarà a continuació) o el projecte Moisès.

High-Touch

Hi-Touch (High-Touch) és va crear dins el programa IST/CRAFT Europea, que tenia l'objectiu de desenvolupar metodologies Web Semàntica i eines per al turisme sostenible dins d'Europa.

Hi-Touch, va ser desenvolupat principalment per Mondeca en OWL utilitzant com a base el "*Thesaurus on Tourism and Leisure Activities*" que s'ha vist al principi del capítol. Al seu nucli es van incloure inicialment les dades de les que constava el repositori de la companyia.

L'ontologia es centra en els productes i expectatives dels clients del turisme classificant les entitats turístiques relacionades a partir d'una xarxa de relacions semàntiques proporcionades per un mapa de temes. El seu ús vol garantir la coherència de la categorització del turisme recursos gestionats en diferents bases de dades distribuïdes i millorar les recerques entre els productes turístics nombroses proporcionar funcionalitats semàntic de la consulta.

L'ontologia conté els següents tres classes de nivell superior:

- Document: Que és refereix a qualsevol tipus de documentació o publicitat d'un producte turístic.
- Objecte: Que és refereix a una oferta turística, que es divideix en cinc subclasses com ara Medi Ambient, Activitats, La imaginació, l'ètica, i Logística.
- Publicació: Que és un document creat a partir dels resultats d'una consulta; les respostes d'una consulta es poden combinar per formar un document PDF.

Ontology Harmonise

L'ontologia Harmonise va esser desenvolupada com a part del projecte Harmonise que era un projecte que va dur a terme la Unió Europea juntament amb diversos estats i entitats internacionals.

Com que les normes i formats dels missatges que estan actualment en ús no sempre permeten una transferència de dades sense fissures, perquè els socis comercials solen aplicar diferents formats de missatges, es va iniciar un projecte amb l'objectiu de fer front als problemes d'interoperabilitat d'informació per reduir el temps i costós del procés en l'àmbit del turisme i poder proporcionar així un servei de mediació obert per a l'intercanvi d'informació dins els membres de la indústria de viatges i turística.

Harmonise és l'element central d'harmoNET (*Harmonisation Network for the Exchange of Travel and Tourism Information*), que té com objectiu la creació d'una xarxa internacional per a la normalització i l'intercanvi de dades en la indústria turística Harmonitzar que després es va estendre a nous subdominis en el projecte d'Harmonització de la Xarxa Transeuropea de turisme (Harm-TEN).

La xarxa harmoNET va ser fundada a l'any 2006 com a successora dels projectes Harmonise i Harm-TEN i, en l'actualitat, està formada per més de 20 membres de la indústria del turisme i d'entitats dedicades a la investigació de les tecnologies de la informació, com la Comissió Europea de Turisme (ETC), la Iniciativa Travel Technology (TTI) o l'Organització Mundial de Turisme (UNWTO) i pretén crear per als seus membres una xarxa internacional per a l'harmonització i l'intercanvi de dades sense fissures en els viatges i el turisme.

harmoNET té els següents objectius oferir un servei de mediació oberta per a proporcionar tecnologies i serveis únics que permet una ràpida i assequible d'intercanvi d'informació, fàcil entre els interlocutors de negociació reunint a persones o organitzacions amb interès en l'intercanvi de dades sense fissures en els viatges i el turisme i dóna suport l'intercanvi d'idees i les últimes novetats en aquest sector.

Harmonise és el model de dades centralitzat que va plasmar tots els models de dades dels participants i cerca proporcionar a les organitzacions turístiques la capacitat d'intercanviar informació sense necessitat de modificar les seves estructures internes. Com que el nombre de participants en aquesta xarxa era molt elevat, no era possible la creació d'un mediador genèric entre dues entitats per la qual cosa es va d'haver d'establir un conjunt mínim de conceptes que representessin les idees més rellevants dins de la indústria del turisme i, a partir d'ell desenvolupar l'ontologia IMHO (*Interoperable Minimum Harmonization Ontology*), que identifica i defineix els conceptes més rellevants de la indústria del turisme i que actua com a mediadora entre diferents ontologies turístiques de cada membre, actuant d'enllaç semàntic entre sistemes i permetent al receptor interpretar la font de dades com una extensió de la seva pròpia base de dades sense preocupar-se de com estan aquestes dades representats.

L'ontologia Harmonise està implementada en RDF i RDFS, i conté al voltant de 200 conceptes i propietats per descriure entitats turístiques i es centra més específicament en les següents dos sub-dominis com són els esdeveniments(conferències, espectacles, esports, etc...) i els allotjaments (hotels, càmpings i apartaments de vacances, etc...)

QALL-ME

QALL-ME és un projecte finançat per la Unió Europea amb l'objectiu establir una infraestructura compartida per diversos idiomes i respondre a preguntes multimodals en l'àmbit del turisme.

QALL-ME
Question Answering Learning technologies in a multiLingual and Multimodal Environment

El sistema QALL-ME permet als usuaris platejar qüestions naturals en diversos idiomes (tant en text i la parla modalitat) utilitzant una gran varietat de dispositius d'entrada com, per exemple, telèfons mòbils, i retorna una llista de respostes específiques amb el format de la modalitat més apropiada, que van des petits textos, mapes, vídeos i fotos.

El principal objectiu de l'ontologia QALL-ME és proporcionar un vocabulari comú per a l'àmbit del turisme com així com una especificació informatitzat del significat dels termes utilitzats en el vocabulari per permetre l'intercanvi de coneixements i la reutilització dels socis i els investigadors posteriors. Cobreix molts dels aspectes de l'àmbit del turisme:

- Destinacions turístiques: Ciutats, pobles,etc.
- Llocs turístics: Allotjaments, llocs d'interès, gastronomia del lloc,etc.
- Esdeveniments: Representacions, esdeveniments esportius, concerts,etc.
- Mitjans de transport: Tren, bus, metro, etc.

L'ontologia QALL-ME està escrita en OWL DL i desenvolupada amb Protégé amb el raonador RacerPro. L'ontologia conté 122 classes, 55 propietats de tipus de dades i l'objecte 52 diferents propietats que indiquen les relacions entre les classes. Les 122 classes es van classificar en 15 classes de nivell superior amb una jerarquia de classes d'un màxim profunditat de 4.

DERI eTourism Ontology

El grup eTurism de DERI (Digital Enterprise Research Institute) va crear una ontologia de turisme com a part del projecte OnTour. L'ontologia DERI eTourism va ser desenvolupada pel institut STI de la Universitat d'Innsbruck amb l'objectiu de crear un portal anomenat eTourism Semantic Web Portal, que s'utilitza per recerques en webs turístiques utilitzant tecnologies de Web Semàntica. L'ontologia està centrada en la descripció d'allotjaments i infraestructures turístiques, per tal de proporcionar tota la informació que pugui ser rellevant a l'hora de cercar un paquet turístic, incloent tota la informació necessària sobre activitats d'oci i dades geogràfiques. Es centra principalment en:

- Allotjaments: Classifica totes les instal·lacions com hotels, cases d'hostes i apartaments.
- Activitats: Activitats a desenvolupar com l'esquí, pista de bitlles, snowboard, etc.
- Infraestructures: es refereix a les instal·lacions previstes les activitats abans esmentades, com ara sales de bitlles, estacions d'esquí, centre eqüestre i pistes de tennis.

Aquesta ontologia al centrar-se en allotjament i infraestructures turístiques té com objectiu intentar relacionar ambdós conceptes per millorar i facilitar la tasca dels viatgers a l'hora d'escollir l'allotjament més proper al seu destí turístic a on pugin dur a terme les activitats que desitgin. Amb ella es planteja poder contestar una sèrie de preguntes basades en els predicats bàsics What(Què), Where(A on), When(Quan), and How(Com?). Preguntes típiques serien: Què pot fer un turista durant la seva estada

a l'illa de Madeira? A on es troben els llocs interessants per veure i visitar? Quan es pot visitar un lloc en particular? Com pot el turista arribar al seu destí per a fer una activitat?

Està desenvolupada amb Protégé i escrita en llenguatge OWL. I es va basar el seu desenvolupament en l'expansió del tesaurus de l'Organització Mundial de Turisme (UNWTO), afegint-hi més propietats i relacions als conceptes escollits.

EON Travelling Ontology

L'ontologia EON Travelling Ontology va ser una de les primeres ontologies turístiques en desenvolupar-se i es fruit de les investigacions realitzades per l'*Institut National de l'Audiovisuel* a França. Descriu conceptes turístics classificats en diverses categories, com a reserves de vols i hotels, mitjans de transport, llocs d'interès turístic i allotjaments. En l'actualitat sembla haver caigut en desús atesa la poca informació existent sobre aquesta.

cDott Ontology

L'ontologia cDOTT (The Core Domain Ontology for Travel and Tourism), desenvolupada al 2009, està basada en les ontologies Harmonise i EON Travelling Ontology esmentades anteriorment. El seu objectiu també és el d'obtenir una ontologia comuna per al sector turístic per a donar suport a la interoperabilitat dels seus actors en operacions a baix nivell. Com s'ha vist a varies de les ontologies anteriors, el fet de mantenir una única ontologia turística per a tots els propòsits no és sempre factible i la idea és proposar una estructura modular de diverses ontologies de propòsit més específic que estiguin, interconnectades entre si.

cDOTT és el node central de l'estructura que conté les dades genèriques sobre el sector turístic i al qual es defineixen objectes, esdeveniments i destinacions turístiques sobre el que es connectaran unes ontologies de propòsit més específic i unes d'altres independents del domini turístic per estendre el seu abast i funcionalitat com a, per exemple, ontologies sobre allotjaments o gastronòmiques, climatològiques, monetaris, de perfil d'usuari, etc.

Gràcies a l'estructura modular es permet una millora de la personalització de les dades proporcionades a l'usuari com a poder escollir diferents models turístics en funció del tipus i les preferències del viatger, facilitant també l'ús de models estàndard com l'ontologia W3C Time per a representar conceptes temporals i permetre connectar diversos models geogràfics a l'hora de classificar diferents tipologies geogràfiques, com regions turístiques, geopolítiques o regions naturals, permetent escollir el viatger la que millor s'ajusti a les seves necessitats. Aquesta ontologia encara està en fase de desenvolupament**.**

TAGA

El projecte *Travel Agent Game in Agentcities*(TAGA) és un marc de treball per a la simulació del mercat mundial de viatges en web. A TAGA, tots els proveïdors de serveis de viatges poden vendre els seus serveis al web (per exemple, vols i hotels) i per tant constitueixen un mercat per webs de viatges, agències que permet ajudar als clients a comprar un paquet de viatge a Internet segons les seves preferències, per exemple, prenent vols econòmics, allotjaments a prop del centre de la ciutat, o menjar a un italià. Dins TAGA es defineixen dues ontologies de

domini per ser utilitzades en les simulacions. La primera es una ontologia de viatges, Travel Ontology que cobreix els conceptes bàsics d'un viatge com itineraris, clients, agències de viatges, i reserves de serveis. La segona és refereix al camp de les subhastes i defineix diferents tipus de subhastes, rols dels participants en elles, protocols utilitzats, etc.

GETESS Ontology

El projecte *German Text Exploitation and Search System* (GETESS) estava destinat a desenvolupar una eina intel·ligent per a la Web de recuperació d'informació en l'àmbit del turisme. GETESS volia habilitar el llenguatge natural com a base per a l'usuari de realització de les consultes de cerca a través de la navegació en una ontologia de domini específic i presentar els resultats de forma entenedora. L'ontologia GETESS conté 1.043 conceptes i 201 relacions i proporciona termes bilingües (anglès i alemany) per a cada concepte. És el servei central per al emmagatzematge i consulta de contingut semàntic per tal de determinar quins fets poden ser extrets de textos, quin esquema de base de dades ha de ser utilitzat per emmagatzemar aquestes dades i quina informació està disponible en el nivell semàntic.

Ontologia Cruzar

Aquesta ontologia s'ha construït a través de la fundació CTIC al novembre de 2007 dins del projecte finançat per l'Ajuntament de Saragossa: "*Un visitante, una ruta*", una aplicació informàtica per al càlcul de rutes turístiques a la ciutat de Saragossa sobre la base del perfil i context de l'usuari. L'ontologia permet modelar l'organització general del domini turístic i garantir la seva interoperabilitat amb altres sistemes similars, i assegurar també la compatibilitat semàntica futura amb altres extensions, en cas que sigui necessari afegir informació sobre nous recursos relacionats amb el turisme. Per a la representació de l'ontologia es va utilitzar el llenguatge OWL DL.

Es va crear a assumint els compromisos ontològics de l'ontologia d'alt nivell, DOLCE (Descriptive Ontology for Linguistic and Cognitive Engineering), desenvolupada pel Laboratory for Applied Ontology de l'Institute of Cognitive Sciences and Technology (ISTC. Els mòduls de la llibreria DOLCE es van utilitzar com a marc de desenvolupament per l'ontologia específica proporcionant interpretacions sobre les classes i propietats més representatives del domini. L'ontologia Cruzar consta de 34 classes i més de 60 propietats i inclou de tres tipus d'entitats:

- Recursos turístics: Monuments, parcs, allotjaments, activitats de lleure, etc.
- Perfils i preferències d'usuaris : Atributs de situació i de preferències d'usuari.
- Rutes turístiques calculades:Descrivint les característiques d'una visita turística (punts d'interès o els esdeveniments relacionats)

Ontologia ANOTA

L'ontologia ANOTA es un projecte desenvolupa dins del projecte ANOTA que està cofinançat per la Unió Europea i el Ministeri d'Educació d'Espanya i en que participen 4 socis:

- Fundación IBIT (Illes Balears Innovació Tecnològica)
- Fundación Robotiker

- Instituto Andaluz de Tecnología (IAT)
- Universitat Oberta de Catalunya (UOC)

L'objectiu principal és evolucionar la tecnologia que permet l'ús de dos tipus d'anotacions, semàntiques i externes a Internet, per facilitar el seu ús entre les PIMES perquè els coneixements i les eines sobre gestió de coneixement i semàntiques eren massa cares per a les empreses petites. L'ontologia ANOTA reutilitza parts de ontologies turístiques existents, així com els esquemes XML de l'especificació de la OpenTravel Alliance (OTA), per definir conceptes en el camp de l'allotjament, de les activitats, dels vols, etc. I encara que pretenia crear 2 prototipus de cost assequible per a les PYMES Des de la seva presentació a Setembre de 2005 el projecte pareix aturat i no hi ha noticies noves ni actualitzacions a la seva web.

CREACIÓ DE L'ONTOLOGIA

Descripció

Després de les investigacions realitzades al apartat sobre l'estat de les ontologies turístiques, queda patent la gran quantitat d'ontologies existent i la gran complexitat d'alguna d'elles. En aquest projecte no s'intenta crear una ontologia turística que pugui representar tota la casuística I objectes representatius existents, sinó que a l'hora de crear-la s'ha intentat adaptar el seu grau de complexitat al objecte d'aquest projecte, es a dir, a l'àmbit de les relacions entre uns punts interès turístics i uns documents associats, essent conscients que es podria haver elevat el grau de complexitat de l'ontologia per a intentar cobrir el màxim d'àmbits possibles; en lloc d'això s'ha desenvolupat pensant en que pugui combinar-se o integrar-se amb unes d'altres ja creades o amb nous desenvolupaments per intentar abastar tot l'àmbit turístic en una ontologia que cobreixi tots els aspectes de forma minuciosa i pugui proporcionar una eina completa per a desenvolupar un marc de treball que permeti aconsellar i donar un bon suport al usuari en les preguntes sobre les seves visites turístiques.

Pel que fa a la mètrica de l'ontologia, aquesta compta amb 95 classes diferents repartides a partir d'una jerarquia de 9 classes d'alt nivell, 24 propietats d'objecte (ObjectType Property) i 59 propietats de dades (DataType Property). Compta amb propietats inverses, simètriques i funcionals. Si bé existeixen algunes propietats que bé podrien esser classes s'ha decidit, indicar-ho i no estendre més l'ontologia havent de tallar per qualque punt el seu abast.

Procés de creació

En quant al mètode utilitzat a l'hora de realitzar-la definició de l'ontologia s'ha fet servir els conceptes recopilats al apartat de procés de creació de l'ontologia que contempla les següents passes:

1. Identificar i determinar el seu domini i el seu abast.
2. Considerar reutilitzar ontologies existents
3. Enumerar els termes importants de les ontologies
4. Definició de la jerarquia de classes
5. Definició de les propietats o slots de les classes.
6. Definició de les restriccions de les propietats.
7. Establir les relacions semàntiques.

8. Creació de les instàncies
9. Avaluació i documentació

Abast

Si bé es pot dir la primera passa a realitzar ja s'ha completat en l'apartat d'Objectius d'aquest treball deixant l'abast i el domini ben identificats, una de les formes de determinar l'abast de l'ontologia és esbossant una llista de qüestions que la base de coneixements basada en l'ontologia hauria d'esser capaç de respondre, preguntes de competència (Gruninger and Fox 1995). Aquestes preguntes serviran després com prova de control de qualitat: L'ontologia conté suficient informació per respondre aquests tipus de preguntes? Les respostes requereixen un nivell particular de detall o representació en un àrea particular? Les preguntes de competència són només un esbós i no necessiten ser exhaustives.

En el domini en que s'està treballant en aquest projecte, el dels punts d'interès turístic i els seus documents associats , les següents preguntes són possibles preguntes de competència:

- A on està l'Ajuntament?
- Existeix qualque arxiu de vídeo de platges?
- Quines són les dades de contacte del Hospital General?
- Què puc trobar aquí a prop del ONO Estadi?
- Quan val entrar a Pacha?
- Quin horari té el museu Es Baluard?
- Quins parcs hi ha a Palma?
- Quant val entrar a la discoteca Pacha?
- Hi ha qualque vídeo en català?

A partir d'aquesta llista de preguntes definides, s'ha d'aconseguir que es pugin contestar a través de l'ontologia creada que haurà la informació de diverses característiques de punt d'interès, horaris, documents, etc... per a poder resoldre-les, cosa que es comprovarà en última instància en forma de interrogatoris per a veure que s'han complit els objectius platejats. La resta del procés seguit es detallarà a continuació.

Reutilització d'ontologies existents

Com a segon pas a realitzar en el desenvolupament de l'ontologia que s'ha creat, s'ha identificat la informació que es vol representar, i per a tal de realitzar una identificació el més exacta possible, es considera el més adequat prendre com a base el coneixement adquirit al investigar algunes de les ontologies turístiques existents ja que realitzades per tot tipus d'experts en el domini en qüestió per tal de intentar aprofitar totes les possibles categoritzacions o classificació que apareixen en elles.

Per a la definició d'alguns aspectes de l'ontologia s'ha fet servir conceptes extrets d'algunes d'elles però no s'han reutilitzat directament cap part d'elles ja que la majoria eren o massa complexes, no eren de codi lliure, no s'ajustaven al àmbit turístic en concret o no concordaven en amb la jerarquia o definició de les seves classes amb l'abast i les preguntes que es volien intentar respondre amb la construcció d'aquesta ontologia. Tot i així han estat de molta utilitat a l'hora de realitzar tot aquest procés de definició.

Les ontologies de les quals s'han pogut extreure conceptes o idees són, per exemple, l'ontologia QALL-ME, l'ontologia Cruzar creada per l'ajuntament de Saragossa, entre d'altres.

Totes elles queden recopilades a tant a l'apartat anterior d'estat de l'art de les ontologies com a la bibliografia d'aquest document.

A més, preferia crear una ontologia des de zero per a poder aprendre i realitzar tot el procés sense botar-me cap passa ni cap part.

Enumeració dels termes importants

En aquest punt, es farà un llistat de tots els termes que es puguin considerar importants segons el domini i l'abast d'aquesta ontologia sense haver preocupar-se de moment del que representen, només s'ha de realitzar un llistat que després ja es classificarà i es començarà la creació de les classes, slots i instàncies a partir d'ells. Així tenim que, en primera instància van aparèixer els termes següents:

Put d'interès	Document associat	Vídeo	Cultura
Fitxer	Format	Àudio	Contacte
Text	Pàgina web	Foto	Monument
Localització	Dades contacte	GPS	Entrada
Preu	Catedral	Cementiri	Hostal
Pensió	Residència	Càmping	Bed&Breakfast
Estació	Comerç	Direcció	Història
Horari	Espai Natural	Servei	Oci
Temporada	Ajuntament	Hotel	Restaurant
Policia	Hospital	Obrir	Temps
Església	Castell	Torre	riu
Centre Comercial	Supermercat	Discoteca	Pub
Bar	Estadi	Poliesportiu	Turístic
Par atraccions	Descripció	Plaça	Font

Un pic s'ha acabat aquest primer resum de termes es pot continuar amb la passa següent essent conscients de que poden aparèixer alguns de nous i desaparèixer alguns d'altres, cosa que segurament passarà al llarg del desenvolupament.

Creació de les classes

Després del primer resum de termes s'ha fet una petita agrupació per a poder facilitar la creació de classes identificant un poc tant les possibles classes que contemplaria l'ontologia com a una precària distribució jeràrquica de les mateixes:

- Document
    - Àudio
    - Vídeo
    - Web
    - Text
    - Web
- Període
    - Temps
    - Dates
- Temporada

- Horari Obertura
- Localització
    - Coordenades GPS
    - Direcció Postal
- Punt Interès
    - Comercial
        - Mercat
        - Tenda
        - Supermercat
        - Centre Comercial
- Interès Turístic
    - Monument
        - Històric
            - Torre
            - Castell
            - Plaça
            - Font
            - Religiós
            - Catedral
            - Església
            - Cementiri
    - Espai Natural
        - Parc
        - Platja
        - Riu
        - Llac
    - Hostaleria
        - Allotjament
            - Hotel
            - Residència
            - Hostal
            - Pensió
            - Càmping
            - Bed&Breakfast
        - Restauració
            - Bar
            - Restaurant
    - Oci
    - Parc d'atraccions
    - Esport
        - Estadi
        - Poliesportiu
    - Nocturn
        - Pub
        - Discoteca
    - Cultural
        - Cinema

        - Teatre
        - Museu
    - Sanitari
        - Hospital
        - Farmàcia
    - Servei
        - Ajuntament
        - Oficina Postal
        - Policia
    - Transport
- Entrada

A partir d'aquesta descripció inicial de les classes s'ha realitzat la definició de les classes començat amb una definició primera de classes base de primer nivell que inclouran altres subclasses més específiques. A més s'introduiran a partir dels termes anomenats nous termes que donaran lloc a noves classes(sobretot subclasses) no contemplades a la figura anterior per tal de donar un poc més de completitud a l'ontologia. De totes formes i com s'ha explicat abans només es crearan les classes que s'han trobat més representatives, cosa que afecta, sobretot a les subclasses relacionades amb els punts d'interès que podrien multiplicar-se de forma exponencial si s'incoessin tots els possibles tipus de punt que pot existir.

Les classes generals que formen part de l'ontologia són 9: DadesContacte, Document, Entrada, HorariObertura, Idioma, Localitzacio, Periode, PuntInteres, Temporada.

A continuació s'explicarà cadascuna d'elles així explicant què representen i les principals subclasses d'elles i el criteri que s'ha seguit per a desglossar-les. Al apartat posterior de resum de classes i slots es podran veure en detall totes elles amb la seva descripció i les seves associades.

Nom de les classes

Alhora de nombrar las classes i les subclasses derivades s'ha aplicat coma format de posar el nom de la classe superior davant del nom que descriuria a la subclasse més particular. Així, per exemple el cas de Document i la seves subclasses DocumentAudio o DocumentVideo.

Descripció de les classes

Classe:DadesContacte

La classe DadesContacte representa les dades de contacte que pot tenir un punt d'interès, constarà de les característiques típiques que poden formar part d'unes dades de contacte com poden esser, nombre de telèfon, correus electrònics etc... Els slots de la classes poden esser múltiples per representar que a unes dades de contacte pot haver-hi més d'un telèfon o correu electrònic associat.

A unes dades de contacte poden associar-se a més d'un punt d'interès però al menys amb un. No té sentit tenir-ne unes sense punt associat. Que unes dades pugin associar-se amb més

d'un punt d'interès s'entén, per exemple, amb hotels de la mateixa cadena, serveis d'atenció al client comuns o públics de la institució que administri varis monuments etc...

Classe:Document

La classe Document representa un document que pot estar associat a un punt d'interès. La classe Document constarà d'una sèrie de propietats per a indicar les característiques més comuns d'un document:Nom, Descripcio, Grandaria, DataCreacio, DataModificacio, Extensio, Autor i URI. Autor, DataModificacio i URI(o ubicació) són les úniques que pot esser associar-se a múltiples valors.

Aquesta classe estarà dividida en 5 subclasses que representen cadascun dels diferents tipus de document que s'ha contemplat: àudio, vídeo, fotografia, text web. Són: DocumentAudio, DocumentVideo, DocumentText, DocumentFotografic i DocumentWeb i seran disjuntes entre si. Les subclasses diferents que mantindran els slots de la superclasse però que tindran algunes característiques pròpies de la subclasse corresponents. Algunes de les subclasses estaran relacionades amb, al menys un element de la classe idioma per indicar en quin idioma està el document però només en les que té sentit(tots menys DocumentFotografic). Un Document de qualsevol classe, port estar relacionat amb 1 o més punts d'interès.

S'ha de comentar també que un document pot estar relacionat a més d'un punt d'interès i que un document de vídeo pot tenir també qualque document d'àudio associat que denotaria que el vídeo no té l'àudio embegut. Si un document d'àudio està associat a un de vídeo, només pot estar-ne amb un.

Classe:Periode:

La classe Periode que intenta representar un període amb un inici i un final. Existiran dues subclasses PeriodeDates i PeriodeHores que representaran conceptual a un període comprés entre dues dates i un període entre dues hores respectivament. Un període de dates comptarà amb una data d'inici i una de fi i podrà esser associat com a mínim a una temporada. Un període d'hores comptarà amb una hora data d'inici i una de fi i podrà esser associat com a mínim a una a un horari.

Classe:Temporada

La classe Temporada representa una temporada que expressa qualque període de temps contemplat entre dues dates i per tant en pot tenir de relacionats un o varis d'ells. Una temporada es podrà associar a un o més horaris. Es contempla que pugi tenir associats més d'un període de dates ja que, per exemple, la temporada alta d'un hotel pot agrupar varis períodes de dates disjunts al llarg del any.

Classe:HorariObertura

La classe HorariObertura representa un horari d'obertura d'un punt d'interès, és a dir, indica un o més horaris en que es pot accedir a ell. Ha de tenir un nom definit.

A un horari d'obertura pot estar a un punt d'interès(o a més, ja que és molts establiments comparteixen el mateix horari). També pot tenir associades temporades definides o no(cosa que voldrà dir que es manté tot l'any) i com a mínim ha de tenir associat un període d'hores i

un o més valors per la propietat dies de la setmana, encara que pot estar associat a varis períodes d'hores i per exemple es parla d'un horari partit.

Classe:Idioma

La classe Idioma representa un idioma i que pot estar associat a 1 o més de les subclasses de document en que té sentit(Àudio, Vídeo, Text o Web). Ha de tenir un nom definit i s'ha decidit que sigui una classe per si en un futur es volgués relacionar, per exemple amb una classe país.

Classe:Localitzacio

La classe Localitzacio representa una localització física. Té dues subclasses per a representar dues formes diferents de representar una localització com son LocalitzacioCoordenadesGPS o LocalitzacioDireccioPostal que la representen una localització a través d'unes coordenades GPS i a través d'una direcció postal, cadascuna inclourà les propietats característiques del model de representació, és a dir, Latitud i Longitud, en el cas de les LocalitzacioCoordenadesGPS i carrer, pis, etc... en el cas de LocalitzacioDireccioPostal.

Una localització pot estar, o no a prop d'un altra mitjançant una relació simètrica, però no té perquè implicar una transivitat entre varis localitzacions que ho estiguin.

Classe:Puntinteres

La classe PuntInteres representa un punt d'interès, és la classe més complexa i juntament amb la classe Document és la més important de totes les que formen l'ontologia.

Un punt d'interès tindrà 1 nom associat i possiblement una descripció. Podrà tenir també associats documents de qualsevol tipus, unes dades de contacte, unes coordenades GPS, una direcció postal i, si escau, un horari d'obertura mitjançant una relació d'objecte amb les classes que les representen. A excepció dels horaris(depenent, per exemple de la temporada), dades de contacte(pot tenir un hotel, per exemple un correu electrònic propi, però un nombre de telèfon o pàgina web corporativa comú a altres de la cadena), i dels documents(diverses fotos, vídeos, etc...) que poden esser múltiples, un punt d'interès, per raons evidents, només tindrà unes coordenades GPS i una direcció postal.

La classe PuntInteres contindrà 7 subclasses que representen els diferents tipus d'interès i àmbit que pot tenir un punt d'interès: Comercial, Oci, Turístic, Sanitari, Servei, Transport i d'Hostaleria. Aquestes subclasses contindran també unes altres més específiques com ja es veurà a la jerarquia de classes i al resum de classes. Algunes d'aquestes subclasses tindran, associada una Entrada però no es una relació que es pugui aplicar a la superclasse com es veurà al explicar la classe Entrada.

En l'àmbit dels punts comercials s'ha inclòs una propietat que reflexa la possibilitat de que un comerç, o alguns tipus de punts d'oci, com a per exemple els cinemes, estiguin inclosos a un centre comercial.

Classe:Entrada

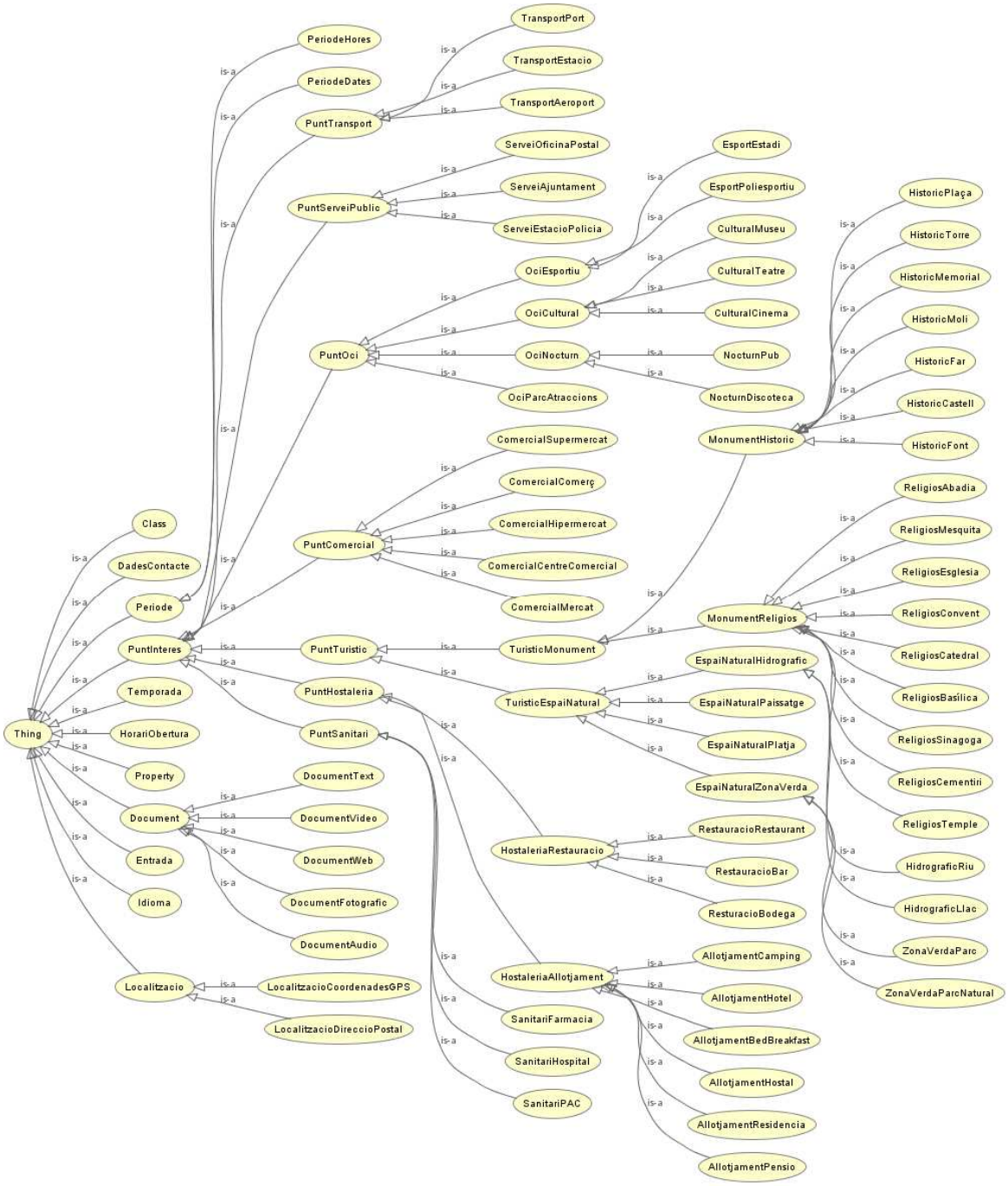
La classe Entrada representa una entrada o tiquet que pot esser necessari per a poder accedir a qualque punt d'interès. Dir que constarà de un nom i d'un preu i que estarà directament

associada al menys un punt d'interès concret(varis punts, com per exemple els cinemes d'una cadena poden utilitzar la mateixa entrada). Només es contempla pagar una entrada per a punts turístics o d'oci.

Jerarquia de classes

Per a realitzar la jerarquia de classes s'ha seguit un procés de desenvolupament top-down, primer definint els conceptes més destacats i després especialitzat apropiadament.

En l'apartat anterior ja s'han explicat breument les classes principals i les seves subclasses immediates però a continuació es pot veure la jerarquia completa de les classes de l'ontologia.



Com es pot veure s'han anant creant subclasses mes específiques en les subclasses de filles de PuntInteres fins a arribar a unes classes de tipus més específiques que indiquen un tipus concret de punt d'interès. Com ja s'ha dit abans es podrien haver creat moltes classes específiques de darrer nivell però només s'han afegit les més importants.

La jerarquia de classes, tal i com es veu a la figura anterior, es representa amb una relació "is-a" ("es-un, es-una"): que significa que una classe A es una subclasse de B i que cada instancia de B es també una instancia de A. Per exemple, PuntOci és una subclasse de PuntInteres com es reflexa a la figura.

Una concepte molt important que s'ha d'entendre que les subclasses hereten les propietats o slots de la classe superior i que si es volen modificar, afegir o eliminar restriccions o eliminar-les completament s'ha de fer expressament. Per exemple, PuntOci hereta tots els slots de PuntInteres.

Una altra tema que s'ha de tenir en compte es definir les classes que són o disjuntes. Què vol dir que siguin disjuntes? Doncs que una instància o individu no pot pertànyer a dues classes disjuntes a l'hora. Per exemple són disjuntes totes les classes principals.

Creació dels slots

Com que les classes aïllades no proveeixen de la suficient informació per respondre les preguntes a les quals vol respondre l'ontologia, un cop s'ha definit les classes i la seva jerarquia, s'ha de descriure l'estructura interna dels conceptes creats.

Per a cada propietat detectada al llistat de termes que no s'hagi convertit en classe s'ha de determinar què descriu i es convertirà en un slot adossat a una classes. Per exemple, 'nom', 'descripció', solen esser típiques propietats que es converteixen en slots en moltes de les classes. La resta de slots de les classes principals s'han definit per tal de completar la representació que vol significar la classe.

A més a més, s'ha de dir, que molts dels slots nous per estendre una classes, serà el que distingiran unes classes de les seves subclasses.

Tipus de valor dels slots

Alhora de crear els slots(propietats) s'ha tingut en compte també els tipus de valor que tindrà. S'ha d'explicar que al hora de crear un slot, pot esser un slot de tipus data (una cadena de text com a per exemple 'nom') o un de tipus objecte que, com veurem representa una relació entre individus.

A la creació de l'ontologia s'ha utilitzat com a tipus de valor els més típics que són:

- **String**: És el més comú, indica una cadena de caràcters i serà utilitzat per slots com ara nom, descripció, etc...
- **Number**: (o bé, Float i Integer per a ser més específics) descriuran slots amb valors numèrics. Per exemple, any o preu.

- **Boolean**: Indiquen només si o no mitjançant els valors booleans típics de 'true' o 'false'. Per exemple s'utilitzaran per indicar si un allotjament té pàrquing, etc...

- **Instance**: Admeten la definició de relacions entre individus. Els slots amb tipus de valor Instance han també definir una llista de classes admeses de les quals les instàncies poden provenir. Per exemple, el slot de isAllotjamentDirecciópostal de la classe HosteleriaAllotjament pot tenir una instància de la classe LocalitzacioDirecciópostal com els seus valors.

Noms dels slots

Alhora de nombrar els slots de data s'ha aplicat el format de posar el prefix '*has*' (té), seguit del nom de la classe davant del nom que descriuria al slot en el cas. Les subclasses que hereten el mateixos noms no els canvien. Així, per exemple el cas de Document i la seves subclasses hasDocumentNom o hasDocumentExtensio. En el cas dels slots de tipus Instancia el prefix dependrà del tipus de relació entre individus que es vulgui representar i el prefix serà '*is*' o '*has*' depenent si de l'origen de la relació.

Domini i rang d'un slot

Les classes admeses per als slots de tipus Instance són sovint anomenades rang d'un slot que es poden restringir per a una classe en particular. Les classes a les quals un slot està adossat són nomenades domini del slot. No hi ha necessitat d'especificar el domini separadament.

Les regles per determinar un domini i un rang d'un slot són similars: Quan es defineix un domini o rang d'un slot, s'ha de trobar les classes o classe més generals que puguin ser respectivament el domini o rang dels slots. Es possible que el domini estigui format per més d'un tipus de classe.

Cardinalitat dels valors dels slots

El darrer punt a aclarir en el procés de creació dels slots de les classes creades, és definir quants pot tenir un slot. Les cardinalitats vendrà descrites per una parella XX:YY a on XX serà la cardinalitat mínima del slot i YY la cardinalitat màxima.

Per indicar que un slot pot no tenir un valor especificat per aquest slot s'utilitzarà 0 com a la seva cardinalitat mínima(valor optatiu), en canvi per indicar que al menys, a de tenir un valor, s'utilitzarà la com a cardinalitat mínima 1.

Per indicar que com a màxim es pot tenir un valor s'utilitzarà com a cardinalitat màxima 1 i per indicar que cardinalitats múltiples (admetent qualsevol quantitat de valors) s'utilitzarà el símbol *(o un nombre superior a 1 si en volem delimitar el nombre màxim però no és habitual) com a cardinalitat màxima.

Per tant tindrem les següents cardinalitats:

Cardinalitat		Expressió
0,1	És optatiu i com a màxim pot tenir 1 valor assignat	MaxCardinality 1

n	És obligatori i només ha de tenir n valor.	cardinality n
0,*	És optatiu i pot tenir més d'un valor assignat	MinCardinality 0
1,*	És obligatori i pot tenir més d'un valor assignat però com a mínim 1.	MinCardinality 1

Resum de classes i slots de l'ontologia

En aquest apartat podem veure un resum de totes les classes creades i els seus slots. Es llistaran cada classe principal i a continuació les seves subclasses.

S'expandirà l'arbre de classes fins a una profunditat de3 totes i baixant més nivells si les subclasses tenen altres subclasses o slots nous o diferents dels heretats de la seva superclasse.

Com que s'indica quines subclasses té cadascuna d'elles(sin en té) només s'inclouran els slots no heretats de la seva superclasse.

Classe: DadesContacte

Nom de la classe	DadesContacte	Label	Dades de contacte
Superclasses	Owl:Thing	Subclasses	-
Disjunt amb	Entrada, HorariObertura, Temporada, Idioma, Localitzacio, Periode, Puninteres, Document		
Descripció	Representa unes dades de contacte		

Slots					
Nom	Descripció	Tipus	Val/class	Car	Invers
hasDadesContacteCorreuElectronic	Indica un correu electrònic d'unes dades de contacte	String	-	0,n	-
hasDadesContacteFax	Indica un número de fax a unes dades de contacte. S'expressa en format +XX YYYYYYYYY a on XX és el codi del país i YYYYYYYYY el número de fax	String	-	0,n	-
hasDadesContactePaginaWeb	Indica una direcció web a unes dades de contacte. Conté la URI completa de la pàgina web.	String	-	0,n	-
hasDadesContacteTelefon	Indica un número de telèfon a unes dades de contacte. S'expressa en format +XX YYYYYYYYY a on XX és el codi del país i YYYYYYYYY el número de fax	String	-	0,n	-
isDadesContactePuntInteres	Indica un punt d'interès a les que estan associades unes dades de contacte.	Instance	PuntInteres	1,n	hasPuntInteresDadesContacte

Classe:Document

Nom de la classe	Document	Label	Document
Superclasses	Owl:Thing	Subclasses	DocumentAudio, DocumentFotografic, DocumentText, DocumentVideo, DocumentWeb

Disjunt amb	Entrada, HorariObertura, Temporada, Idioma, Localitzacio, Periode, Puninteres, DadesContacte				
Descripció	Representa un document amb un nom, un format i una URI a on està situat				
Slots					
Nom	**Descripció**	**Tipus**	**Val/class**	**Car**	**Invers**
hasDocumentAutor	Indica un autor d'un document. Un document pot tenir-ne més d'un.	String	-	0,n	-
hasDocumentdataCreacio	Indica la data de creació d'un document. És única	dateTime	-	1	-
hasDocumentDataModificacio	Indica una data de modificació d'un document.	dateTime	-	0,n	-
hasDocumentDescripcio	Indica la descripció d'un document	String	-	0,1	-
hasDocumentExtensio	Indica l'extensió d'un document (txt, jpg, etc...). No s'ha de posar el punt abans de l'extensió.	String	-	1	-
hasDocumentGrandaria	Indica la grandària que ocupa el document en bytes.	Integer	-	1	-
hasDocumentNom	Indica el nom d'un document	String	-	1	-
hasDocumentURI	Indica una ubicació URI a on es troba un document.	String	-	0,n	-
isDocumentAssociat	Indica un punt d'interès al que està associat un document. Pot estar associat a més d'un.	Instance	PuntInteres	0,n	hasPuntInteresDocumentAssociat

Classe:DocumentAudio

Nom de la classe	DocumentAudio	**Label**	Document d'àudio
Superclasses	Document	**Subclasses**	-
Disjunt amb	DocumentFotografic, DocumentText, DocumentVideo		
Descripció	Representa un document d'àudio amb un nom, un format i una URI a on està situat		
Slots			

Nom	**Descripció**	**Tipus**	**Val/class**	**Car**	**Invers**
hasDocumentAudioBitrate	Indica la velocitat de transmissió (bitrate) del document d'audio. Es medeix en kbits/s.	Integer	-	1	-
hasDocumentAudioResolucioAudio	Indica el nombre de bits per enmagatzemar una muestra de la senyal de un document d'àudio.	Integer	-	1	-
hasDocumentAudioTipusBitrate	Indica el tipus de bitrate de un document d'àudio. Pot esser constant (CBR) o variable (VBR)	String	CBR VBR	1	-

hasDocumentAudioVelocitatMostreig	Indica la velocitat de mostreig de la senyal d'audio d'un document d'àudio. En KHz	Integer	-	1	-
hasDocumentCodec	Indica el códec amb el que ha estat codificat el document (MPEG,etc...)	String	-	1	-
hasDocumentDurada	Indica la durada del document audiovisual en segons	Duration	-	1	-
hasDocumentNombreCanals	Indica el nombre de canals d'audio que té el document	Integer	-	1	-
hasDocumentdioma	Indica un idioma associat un document	Instance	Idioma	1,n	hasIdiomaDocument
isDocumentAudioAssociat	Indica un documents de vídeo associat a un document d'audio	Instance	DocumentVideo	0,n	hasdocumentVideoAudioAssociat

Classe:DocumentVideo

Nom de la classe	DocumentVideo	Label	Document de vídeo
Superclasses	Document	Subclasses	-
Disjunt amb	DocumentFotografic, DocumentText, DocumentAudio		
Descripció	Representa un document de vídeo amb un nom, un format i una URI a on està situat		
Slots			

Nom	Descripció	Tipus	Val/class	Car	Invers
hasDocumentAlçada	Indica l'alçada d'un document en píxels	Integer	-	1	-
hasDocumentAmplada	Indica l'amplada de visualització d'un document en píxels	Integer	-	1	-
hasDocumentCodec	Indica el códec amb el que ha estat codificat el document (MPEG,etc...)	String	-	1	-
hasDocumentDurada	Indica la durada del document audiovisual en segons	Duration	-	1	-
hasDocumentNombreCanals	Indica el nombre de canals d'àudio que té el document	Integer	-	1	-
hasDocumentdioma	Indica un idioma associat un document	Instance	Idioma	1,n	hasIdiomaDocument
hasDocumentVideoAudioAssociat	Un document d'àudio associat a un document de vídeo. pot tenir associat més més d'un àudio per si l'àudio està a un fitxer independent i hi ha un audio per a cada idioma per exemple.	Instance	DocumentAudio	0,n	isDocumentAudioAssociat

Classe:DocumentText

Nom de la classe	DocumentText	Label	Document de Text
Superclasses	Document	Subclasses	-

Disjunt amb	DocumentFotografic, DocumentVideo, DocumentAudio				
Descripció	Representa un document de text amb un nom, un format i una URI a on està situat				
Slots					
Nom	**Descripció**	**Tipus**	**Val/class**	**Car**	**Invers**
hasDocumentTextNombre Pagines	Indica el nombre de pàgines que té un document de text	Integer	-	1	-
hasDocumentTextFormatC odificacio	Indica el format de la codificació de un document de text (utf8, etc...)	String	-	1	-
hasDocumentdioma	Indica un idioma associat un document	Instance	Idioma	1,n	hasIdiomaDocument

Classe:DocumentFotografic

Nom de la classe	DocumentFotografic	**Label**	Document fotogràfic
Superclasses	Document	**Subclasses**	-
Disjunt amb	DocumentVideo, DocumentText, DocumentAudio		
Descripció	Representa un document fotogràfic amb un nom, un format i una URI a on està situat		
Slots			
Nom	**Descripció**	**Tipus**	**Val/class**	**Car**	**Invers**
hasDocumentAlçada	Indica l'alçada d'un document en pixels	Integer	-	1	-
hasDocumentAmplada	Indica l'amplada de visualització d'un document en pixels	Integer	-	1	-

Classe:DocumentWeb

Nom de la classe	DocumentWeb	**Label**	Document web
Superclasses	Document	**Subclasses**	-
Disjunt amb	-		
Descripció	Representa un document web amb un nom, un format i una URI a on està situat		
Slots			
Nom	**Descripció**	**Tipus**	**Val/class**	**Car**	**Invers**
hasDocumentWebEtiqueta Metadata	Indica una etiqueta o tag de metadata de un document web	String	-	0,n	-
hasDocumentdioma	Indica un idioma associat un document	Instance	Idioma	1,n	hasIdiomaDocument

Classe:Entrada

Nom de la classe	Entrada	**Label**	Entrada
Superclasses	Owl:Thing	**Subclasses**	-

Disjunt amb	DadesContacte, HorariObertura, Temporada, Idioma, Localitzacio, Periode, Puninteres, Document				
Descripció	Representa una entrada o ticket				
Slots					
Nom	**Descripció**	**Tipus**	**Val/class**	**Car**	**Invers**
hasentradaDescripcio	Indica una descripció d'una entrada i pot incloure condicions associada a una entrada.	String	-	0,1	-
hasEntradaNom	Indica el nom assignat a una entrada	String	-	1	-
hasEntradaPreu	Indica el preu d'una entrada. S'especifica en euros	Integer	-	1	-
isEntradaAssociadaPuntInteres	Indica un punt d'interès associat a una entrada. Només es pot utilitzar per aquest punt d'interès.	Instance	PuntOci PuntTuristic	1,n	-

Classe:HorariObertura

Nom de la classe	HorariObertura	**Label**	Horari d'obertura
Superclasses	Owl:Thing	**Subclasses**	-
Disjunt amb	DadesContacte, Entrada, Temporada, Idioma, Localitzacio, Periode, Puninteres, Document		
Descripció	Representa un idioma que es pot associar a un idioma		
Slots			

Nom	**Descripció**	**Tipus**	**Val/class**	**Car**	**Invers**
hasHorariOberturaDiaSetmana	Indica un dia de la setmana en un horari	String	• Dilluns • Dimarts • Dimecres • Dijous • Divendres • Dissabte • Diumenge	1,n	-
hasHorariOberturaNom	Indica un nom assignat a un horari	String	-	1	-
hasHorariOberturaPeriodeHores	Indica un període d'hores associat a un l'horari d'obertura	Instance	PeriodeHores	1,n	isPeriodeHoresHorariAssociat
hasHorariOberturaTemporada	Indica una temporada per a un horari d'obertura	Instance	Temporada	0,n	hasTemporadaHorariOberturaAssociat
isHorariOberturaPuntInteres	Indica un punt d'interès associat a un horari d'obertura. Només pot tenir-ne 1.	Instance	PunInteres	0,n	hasPuntInteresHorariObertura

Classe:Idioma

Nom de la classe	Idioma	Label	Idioma
Superclasses	Owl:Thing	Subclasses	-
Disjunt amb	DadesContacte, HorariObertura, Temporada, Entrada, Localitzacio, Periode, PunInteres, Document		
Descripció	Representa un idioma que es pot associar a un idioma		
Slots			

Nom	Descripció	Tipus	Val/class	Car	Invers
hasIdiomaNom	Indica un nom per l' idioma	String		1	-
hasIdiomaDocument	Indica un document que tingui associat un idioma	Instance	DocumentAudio, DocumentVideo,	0,n	-

Classe:Localitzacio

Nom de la classe	Localitzacio
Label	Localització
Superclasses	Owl:Thing
Subclasses	LocalitzacioCoordenadesGPS, LocalitzacioDireccioPostal
Disjunt amb	DadesContacte, HorariObertura, Temporada, Entrada, Idioma, Periode, PunInteres, Document
Descripció	Representa una localització
Slots	

Nom	Descripció	Tipus	Val/class	Car	Invers
hasLocalitzacioAprop	Indica que una localització està aprop d'un altra. És simètrica	Instance	Localitzacio	0,n	-

Classe:LocalitzacioCoordenadesGPS

Nom de la classe	LocalitzacioCoordenadesGPS	Label	Coordenades GPS
Superclasses	Localitzacio	Subclasses	-
Disjunt amb	DireccioPostal		
Descripció	Representa una localització mitjançant unes coordenades GPS		
Slots			

Nom	Descripció	Tipus	Val/class	Car	Invers
hasCoordenadesGPSLatitud	Valor que indica la latitud en graus per indicar la posició segons les coordenades	String	-	1	-

	GPS				
hasCoordenadesGPSLongitud	Valor que indica la longitud en graus per indicar la posició segons les coordenades GPS	String	-	1	-
isCoordenadesGPSPuntInteres	Indica un punt d'interès associat a unes coordenades GPS. Unes coordenades GPS poden tenir associats més d'un punt d'interès	Instance	PuntInteres	1,n	hasPuntInteresCoordenades

Classe:LocalitzacioDireccioPostal

Nom de la classe	LocalitzacioDireccioPostal	Label	Direcció Postal
Superclasses	Localitzacio	Subclasses	-
Disjunt amb	CoordenadesGPS		
Descripció	Representa una localització mitjançant una direcció postal		
Slots			

Nom	Descripció	Tipus	Val/class	Car	Invers
hasDireccioPostalCarrer	Indica el carrer d'una direcció postal	String	-	0,1	-
hasDireccioPostalPis	Indica el pis d'una direcció postal	String	-	0,1	-
hasDireccioPostalCodiPostal	Indica la codi postal d'una direcció postal	Integer	-	0,1	-
hasDireccioPostalProvincia	Indica la província d'una direcció postal	String	-	0,1	-
hasDireccioPostalMunicipi	Indica el municipi d'una direcció postal	String	-	0,1	-
hasDireccioPostalPais	Indica la país d'una direcció postal	String	-	0,1	-
isDireccioPostalPuntInteres	Indica un punt d'interès associat a una direcció postal Una direcció postal poden tenir associats més d'un punt d'interès	Instance	PuntInteres	1,n	hasPuntInteresDireccioPostal

Classe:Periode

Nom de la classe	Periode	Label	Període
Superclasses	Owl:Thing	Subclasses	PeriodeDates,PeriodeHores
Disjunt amb	DadesContacte, HorariObertura, Entrada, Idioma, Localitzacio, Temporada, PunInteres, Document		
Descripció	Representa un període amb un inici i un fi		
Slots	-		

Classe:PeriodeDates

Nom de la classe	PeriodeDates	Label	Període de dates
Superclasses	Periode	Subclasses	-
Disjunt amb	PeriodeHores		
Descripció	Representa un període de dates amb una data d'inici i una de fi		
Slots			

Nom	Descripció	Tipus	Val/class	Car	Invers
hasPeriodeDatesdataInici	Indica la data d'inci de un període de temps	Date	-	1	-
hasPeriodeDatesDataFi	Indica la data de fi de un període de temps	Date	-	1	-
isPeriodeDatesTemporada	Indica una Temporada al que el període de dates està associat	Instance	Temporada	1,n	hasTemporadaPeriodeDates

Classe:PeriodeHores

Nom de la classe	PeriodeHores	Label	Període d'hores
Superclasses	Periode	Subclasses	-
Disjunt amb	PeriodeDates		
Descripció	Representa un període d'hores amb una hora d'inici i una de fi		
Slots			

Nom	Descripció	Tipus	Val/class	Car	Invers
hasPeriodeHoresHoraInici	Indica l'hora d'inci de un període d'hores	Time	-	1	-
hasPeriodeHoresHoraFi	Indica l'hora de fi de un període d'hores	Time	-	1	-
isPeriodeHoresHorariAssociat	Indica un l'horari d'obertura al que el periode està associat	Instance	Horariobertura	1,n	hasHorariOberturaPeriodeHores

Classe:Temporada

Nom de la classe	Temporada	Label	Temporada
Superclasses	Owl:Thing	Subclasses	-
Disjunt amb	DadesContacte, HorariObertura, Entrada, Idioma, Localitzacio, Periode, Puninteres, Document		
Descripció	Representa una temporada que compren un o més períodes de dates		
Slots			

Nom	Descripció	Tipus	Val/class	Car	Invers

hasTemporadaNom	Indica un nom per a la temporada	String	-	1	-
hasTemporadaPeriodeDates	Indica Representa els possibles períodes de dades que conte la temporada	Instance	PeriodeDates	0,n	isPeriodeDatesTemporada
hasTemporadaHorariObertura	Indica que una temporada està associada a un horari d'obertura	Instance	Horariobertura	0,n	hashorariOberturaTemporada

Classe:PuntInteres

Nom de la classe	PuntInteres	**Label**	Punt d'interès
Superclasses	Owl:Thing	**Subclasses**	PuntComercial, PuntHostalaria, PuntOci, PuntSanitari, PuntServei, PuntTransport, PuntTuristic
Disjunt amb	DadesContacte, HorariObertura, Entrada, Idioma, Localitzacio, Periode, Temporada, Document		
Descripció	Representa un punt d'interès turístic que consta d'una sèrie de característiques generals		

Slots

Nom	Descripció	Tipus	Val/class	Car	Invers
hasPuntInteresNom	Indica el nom d'un punt d'interès	String	-	1	-
hasPuntInteresDescripcio	Indica la descripció d'un punt d'interès	String	-	0,1	-
hasPuntInteresDadesContecte	Indica unes dades de contacte associades a un punt d'interès.	Instance	DadesContacte	0,n	isDadesContactePuntInterès
hasPuntInteresCoordenadesGPS	Indica les coordenades GPS associades a un punt d'interès.	Instance	CoordenadesGPS	0,1	isCoordenadesGPSPuntInteres
hasPuntInteresDireccioPostal	Indica la direcció postal associada una punt d'interès	Instance	DireccioPostal	0,1	isDireccioPostalPuntInteres
hasPuntInteresDocumentAssociat	Representa un document associat al punt d'interès. Pot tenir-ne associat més d'un.	Instance	Document	0,n	isDocumentAssociatPuntInteres
hasPuntInteresHorariObertura	Indica un horari d'obertura associat al punt d'interès. Pot haver-hi més d'un	Instance	HorariObertura	0,n	isHorariOberturaPuntInteres

Classe:PuntComercial

Nom de la classe	PuntComercial	**Label**	Punt comercial
Superclasses	PuntInteres	**Subclasses**	ComercialCentreComercial, ComercialComerç, ComercialHipermercat, ComercialMercat, ComercialSupermercat
Disjunt amb	PuntHostalaria, PuntOci, PuntSanitari, PuntServei, PuntTransport		
Descripció	Representa un punt d'interès des del punt de vista comercial		

Slots

Nom	Descripció	Tipus	Val/class	Car	Invers

hasPuntInteresCadena	Indica que el punt d'interès pertany a una cadena o companyia	String	-	0,1	-
hasPuntInteresCapacitat	Capacitat en persones del punt d'interès	Integer	-	0,1	-

Classe: ComercialCentreComercial

Nom de la classe	ComercialCentreComercial	**Label**	Centre Comercial
Superclasses	PuntComercial	**Subclasses**	-
Disjunt amb	ComercialComerç, ComercialHipermercat, ComercialMercat, ComercialSupermercat		
Descripció	Representa un centre comercial compost de altres tipus de punt de comerç		

Slots

Nom	Descripció	Tipus	Val/class	Car	Invers
hasCentreComercialPuntContingut	Indica que un punt d'interès està contingut en un centre comercial . Poden formar part d'ell els comerços típics que es poden trobar a un centre comercial	Instance	ComercialComerç, ComercialHipermercat, ComercialMercat, ComercialSupermercat, PuntOci, HostalariaRestauracio, SanitariFarmacia	0,n	isPartCentreComercial

Classe: ComercialComerç

Nom de la classe	ComercialComerç	**Label**	Comerç
Superclasses	PuntInteres	**Subclasses**	-
Disjunt amb	ComercialCentreComercial, ComercialHipermercat, ComercialMercat, ComercialSupermercat		
Descripció	Representa un punt comercial de tipus comerç de tipus tradicional o tenda		

Slots

Nom	Descripció	Tipus	Val/class	Car	Invers
isPartCentreComercial	Indica que un punt d'interès forma part d'un centre comercial. només pot formar-ne part d'un	Instance	ComercialCentreComercial	0,1	hasCentreComercialPuntContingut

Classe: ComercialSupermercat

Nom de la classe	ComercialSupermercat	**Label**	Supermercat
Superclasses	PuntComercial	**Subclasses**	-
Disjunt amb	ComercialCentreComercial, ComercialHipermercat, ComercialMercat, ComercialComerç		

Descripció	Representa un supermercat				
Slots					
Nom	**Descripció**	**Tipus**	**Val/class**	**Car**	**Invers**
isPartCentreComercial	Indica que un punt d'interès forma part d'un centre comercial. només pot formar-ne part d'un	Instance	ComercialCentreComercial	0,1	hasCentreComercialPuntContingut

Classe: ComercialHipermercat

Nom de la classe	ComercialHipermercat	Label	Hipermercat
Superclasses	PuntComercial	Subclasses	-
Disjunt amb	ComercialCentreComercial, ComercialComerç, ComercialMercat, ComercialSupermercat		
Descripció	Representa un hipermercat		
Slots			

Nom	Descripció	Tipus	Val/class	Car	Invers
isPartCentreComercial	Indica que un punt d'interès forma part d'un centre comercial. només pot formar-ne part d'un	Instance	ComercialCentreComercial	0,1	hasCentreComercialPuntContingut

Classe: ComercialMercat

Nom de la classe	ComercialMercat	Label	Mercat
Superclasses	PuntComercial	Subclasses	-
Disjunt amb	ComercialCentreComercial, ComercialHipermercat, ComercialComerç, ComercialSupermercat		
Descripció	Representa un mercat		
Slots			

Nom	Descripció	Tipus	Val/class	Car	Invers
isPartCentreComercial	Indica que un punt d'interès forma part d'un centre comercial. només pot formar-ne part d'un	Instance	ComercialCentreComercial	0,1	hasCentreComercialPuntContingut

Classe:PuntHostalaria

Nom de la classe	PuntHostalaria	Label	Punt hostalaria
Superclasses	PuntInteres	Subclasses	HostalariaAllotjament, HostalariaRestauracio
Disjunt amb	PuntComercial, PuntOci, PuntSanitari, PuntServei, PuntTransport		
Descripció	Representa un punt d'interès des del punt de vista de l'hostaleria		
Slots			

Nom	Descripció	Tipus	Val/class	Car	Invers Func
hasPuntInteresCadena	Indica que el punt d'interès pertany a una cadena o companyia	String	-	0,1	-
hasPuntInteresCapacitat	Capacitat en persones del punt d'interès	Integer	-	0,1	-

Classe:HostalariaAllotjament

Nom de la classe	HostalariaAllotjament	Label	Allotjament
Superclasses	PuntInteres	Subclasses	AllotjamentBedBreakfast, AllotjamentCamping, AllotjamentHostal, AllotjamentHotel, AllotjamentPensio, Allotjament Residencia
Disjunt amb	-		
Descripció	Representa un tipus de punt d' allotjament		
Slots			

Nom	Descripció	Tipus	Val/class	Car	Invers Func
hasPuntInteresNombreHabitacions	Indica el nombre d'habitacions d'un allotjament	Integer	-	0,1	-
hasAllotjamentQualificacio	I Indica la qualificació de qualitat d'un punt d'allotjament(estrelles, claus, etc...)	String	-	0,1	-

Classe:HostalariaRestauracio

Nom de la classe	HostalariaRestauracio	Label	Punt de restauració
Superclasses	PuntInteres	Subclasses	RestauracioBar,RestauracioRestaurant,RestauracioBodega
Disjunt amb	-		
Descripció	Representa un tipus de punt de restauració		
Slots			

Nom	Descripció	Tipus	Val/class	Car	Invers Func
hasRestauracioQualificacio	I Indica la qualificació de qualitat d'un punt de restauració (estrelles michelin, forquetes, etc...)	String	-	0,1	-

Classe:PuntOci

Nom de la classe	PuntOci	Label	Punt d'oci
Superclasses	PuntInteres	Subclasses	OciCultural, OciEsportiu, OciNocturn, OciParcAtraccions
Disjunt amb	PuntComercial, PuntHostalaria, PuntSanitari, PuntServei, PuntTransport		

Descripció	Representa un punt d'interès des del punt de vista del oci				
Slots					
Nom	**Descripció**	**Tipus**	**Val/class**	**Car**	**Invers Func**
hasPuntOciEdatMinima	Indica l'edat mínima per a poder accedir al punt d'oci	Integer	-	0,1	-
hasPuntInteresCadena	Indica que el punt d'interès pertany a una cadena o companyia	String	-	0,1	-
hasPuntInteresCapacitat	Capacitat en persones del punt d'interès	Integer	-	0,1	-
hasPuntInteresEntrada	Indica una entrada associada al punt d'interès que s'ha de pagar per accedir-hi	Instance	Entrada	0,n	isEntradaAssociadaPuntInteres
isPartCentreComercial	Indica que un punt d'interès forma part d'un centre comercial. només pot formar-ne part d'un	Instance	ComercialCentreComercial	0,1	hasCentreComercialPuntContingut

Classe: OciCultural

Nom de la classe	OciCultural	Label	Punt d'oci cultural
Superclasses	PuntOci	Subclasses	CulturalCinema,CulturaTeatre,CulturalMuseu
Disjunt amb	OciEsportiu, OciNocturn, OciParcatraccions		
Descripció	Representa un punt d'interès des del punt de vista cultural		
Slots	No en té de pròpies definides		

Classe: OciEsportiu

Nom de la classe	OciEsportiu	Label	Punt d'oci esportiu
Superclasses	PuntOci	Subclasses	EsportiuEstadi,EsportiuPoliesportiu
Disjunt amb	OciCultural, OciNocturn, OciParcatraccions		
Descripció	Representa un punt d'oci esportiu		
Slots			
Nom	**Descripció**	**Tipus**	**Val/class**	**Car**	**Invers Func**
hasOciEsportiuEsport	Indica un esport que es practica al punt esportiu. No s'ha plasmat com una classe per tal de no expandir massa l'ontologia però seria convenient fer-ho.	String	-	0,n	-

Classe: OciNocturn

Nom de la classe	PuntOciNocturn	Label	Punt d'oci nocturn

Superclasses	PuntOci	**Subclasses**	NocturnPub, NocturnDiscoteca
Disjunt amb	OciEsportiu, OciNocturn, OciParcAtraccions		
Descripció	Representa un punt d'oci nocturn		
Slots	No en té de pròpies definides		

Classe: OciParcAtraccions

Nom de la classe	PuntParcAtraccions	**Label**	Parc d'atraccions
Superclasses	PuntOci	**Subclasses**	-
Disjunt amb	OciEsportiu, OciNocturn, OciCultural		
Descripció	Representa un par d'atraccions		
Slots			

Nom	Descripció	Tipus	Val/class	Car	Invers Func
hasParcAtraccionsTema	Indica la temàtica de un parc d'atraccions	String	-	0,1	-

Classe:PuntSanitari

Nom de la classe	PuntSanitari	**Label**	Punt d'interès sanitari
Superclasses	PuntInteres	**Subclasses**	SanitariHospital, SanitariFarmacia, SanitariPAC
Disjunt amb	PuntComercial, PuntHostalaria, PuntOci, PuntServei, PuntTransport		
Descripció	Representa un punt d'interès des del punt de vista sanitari		
Slots			

Nom	Descripció	Tipus	Val/class	Car	Invers Func
hasPuntInteresCadena	Indica que el punt d'interès pertany a una cadena o companyia	String	-	0,1	-

Classe: SanitariHospital

Nom de la classe	SanitariHospital	**Label**	Hospital
Superclasses	PuntSanitari	**Subclasses**	-
Disjunt amb	SanitariHospital, SanitariPAC		
Descripció	Representa un hospital		
Slots			

Nom	Descripció	Tipus	Val/class	Car	Invers Func
hasPuntInteresNombreHa	Indica el nombre d'habitacions de un	Integer	-	0,1	-

bitacions	punt d'interès				

Classe: SanitariFarmacia

Nom de la classe	SanitariFarmacia	Label	Farmacia
Superclasses	PuntSanitari	Subclasses	-
Disjunt amb	-		
Descripció	Representa una farmàcia		
Slots			

Nom	Descripció	Tipus	Val/class	Car	Invers Func
isPartCentreComercial	Indica que un punt d'interès forma part d'un centre comercial. només pot formar-ne part d'un	Instance	ComercialCentreComercial	0,1	hasCentreComercialPuntContingut

Classe: PuntServeiPublic

Nom de la classe	PuntServeiPublic	Label	Punt interès servei públic
Superclasses	PuntInteres	Subclasses	ServeiAjuntament, ServeiEstacioPolicia, ServeiOficinaPostal
Disjunt amb	TuristicEspaiNatural, PuntTransport		
Descripció	Representa un punt d'interès des del punt de vista de servei públic		
Slots	No en té de pròpies definides		

Classe: PuntTransport

Nom de la classe	PuntTransport	Label	Punt transport
Superclasses	PuntInteres	Subclasses	TransportAeroport, TransportMaritim, TransportEstacio
Disjunt amb	TuristicEspaiNatural		
Descripció	Representa un punt d'interès des del punt de vista del transport		
Slots	No en té de pròpies definides		

Classe: TransportEstacio

Nom de la classe	TransportEstacio	Label	Estació
Superclasses	PuntTransport	Subclasses	-
Disjunt amb	-		
Descripció	Representa una Estació de Metro, Bus, Autocar, Tramvia, Tren, etc...		
Slots			

Nom	Descripció	Tipus	Val/class	Car	Invers Func
hasEstacioMediTransport	Indica que un punt d'interès forma part d'un centre comercial. Només pot formar-ne part d'un. No s'ha plasmat com una classe per tal de no expandir massa l'ontologia però seria convenient fer-ho.	String	-	1,n	-

Classe:PuntTuristic

Nom de la classe	PuntTuristic	Label	Punt turístic
Superclasses	PuntInteres	Subclasses	TuristicMonument, TuristicEspaiNatural
Disjunt amb	-		
Descripció	Representa un punt d'interès turístic		
Slots			

Nom	Descripció	Tipus	Val/class	Car	Invers Func
hasPuntInteresCadena	Indica que el punt d'interès pertany a una cadena o companyia	String	-	0,1	-
hasPuntTuristicHistoria	Indica una història pel punt turístic	String	-	0,1	-
hasPuntInteresEntrada	Indica una entrada associada al punt d'interès que s'ha de pagar per accedir-hi	Instance	Entrada	0,n	isEntradaAssociadaPuntInteres

Classe: TuristicEspaiNatural

Nom de la classe	TuristicEspaiNatural	Label	Espai Natural
Superclasses	PuntTuristic	Subclasses	EspaiNaturalHidrografic, EspaiNaturalPlatja, EspaiNaturalPaissatge, EspaiNaturalZonaVerda
Disjunt amb	-		
Descripció	Representa un espai natural		
Slots			

Nom	Descripció	Tipus	Val/class	Car	Invers Func
hasEspaiNaturalExtensio	Indica l'extensió d'un espai natural en metres quadrats	Integer	-	0,1	-

Classe: EspaiNaturalHidrografic

Nom de la classe	EspaiNaturalHidrografic	Label	Espai Natural hidrogràfic
Superclasses	TuristicEspaiNatural	Subclasses	HidrograficRiu, HidrograficLlac
Disjunt amb	EspaiNaturalPlatja, EspaiNaturalPaissatge, EspaiNaturalZonaVerda		

Descripció	Representa un espai natural hidrògrafic
Slots	No en té de pròpies definides

Classe: EspaiNaturalZonaVerda

Nom de la classe	EspaiNaturalZonaVerda	**Label**	Zona Verda
Superclasses	TuristicEspaiNatural	**Subclasses**	ZonaVerdaParc, ZonaVerdaParcNatural
Disjunt amb	EspaiNaturalPlatja, EspaiNaturalPaissatge, EspaiNaturalHidrografic		
Descripció	Representa una zona verda		
Slots	No en té de pròpies definides		

Classe: TuristicMonument

Nom de la classe	TuristicMonument	**Label**	Monument
Superclasses	PuntTuristic	**Subclasses**	MonumentHistoric, MonumentReligios
Disjunt amb	-		
Descripció	Representa un monument		
Slots			

Nom	Descripció	Tipus	Val/class	Car	Invers Func
hasPuntInteresCapacitat	Capacitat en persones del punt d'interès	Integer	-	0,1	-

Classe: MonumentHistoric

Nom de la classe	MonumentHistoric	**Label**	Monument Històric
Superclasses	TuristicMonument	**Subclasses**	HistoricCastell, HistoricMoli, HistoricFar, HistoricTorre, HistoricPlaça, HistoricFont, HistoricMemorial
Disjunt amb	-		
Descripció	Representa un monument de tipus històric		
Slots			

Nom	Descripció	Tipus	Val/class	Car	Invers Func
hasPuntTuristicHistoria	Indica una història pel punt turístic. En el cas de un monument històric és obligatori	String	-	0,1	-

Classe: MonumentReligios

Nom de la classe	MonumentHistoric	**Label**	Monument Religiòs

Superclasses	TuristicMonument	Subclasses	ReligiosEsglesia, ReligiosCatedral, ReligiosBasilica, ReligiosTemple, ReligiosCementiri, ReligiosMesquita, Religiossinagoga, ReligiosAbadia, ReligiosConvent
Disjunt amb	-		
Descripció	Representa un monument de tipus religiòs		

Slots					
Nom	Descripció	Tipus	Val/class	Car	Invers Func
hasReligiosReligio	Indica una religió associada a un monument per qualsevol raó	String	-	0,n	-

Instanciació de les classes

Un pic s'ha acabat de crear l'ontologia s'ha de començar a instanciar-la, és a dir crear instàncies de les diferents classes assignant valors als seus slots per tal d'aconseguir una mostra suficientment àmplia per a poder realitzar les proves i poder comprovar la utilitat de l'ontologia.

Els individus inserits seran una mostra suficient per a demostrar els casos més habituals, intentant incloure totes les casuístiques en quan a relacions entre classes. Les classes creades duran el procés d'instanciació quedaran emmagatzemades en un arxiu .repository amb el mateix nom que l'arxiu del projecte.

Per classes, s'ha creat el següent nombre d'individus:

Classe	Instancies	Classe	Instancies
DadesContacte	DadesContacteEstacioCentral DadesContactePachaPalma DadesContacteRiu DadesContacteRiuPortoPi DadesContacteRiuSantaMaria DadesContacteSantFrancesc	**DocumentAudio**	DocumentAudioFontTortugues DocumentAudioPromoHogans DocumentAudioPromoReialMallorca DocumentAudioSantFrancesc DocumentAudioVideoPlatjaPalma
DocumentVideo	DocumentVideoPlatjaPalma DocumentVideoPromoPacha	**DocumentFotografic**	FotografiaBasilicaSantFrancesc FotografiaCastellBellver FotografiaFiraRam FotografiaFiraRam2 FotografiaHotelRiuPortoPi FotografiaONOEstadi FotografiaPortoPiCentre
DocumentText	DocumentTextOfertesCarrefourMaig2011	**DocumentWeb**	DocumentWebPortoPiCentre
PeriodeHores	PeriodeHores08001800 PeriodeHores08002100 PeriodeHores23000600 PeriodeHores08001800 PeriodeHores23590000	**PeriodeDates**	PeriodeDatesFiraRam2011 PeriodeDatesTemporadaAlta PeriodeDatesTemporadaBaixa1 PeriodeDatesTemporadaBaixa2
Temporada	TemporadaAlta TemporadaBaixa TemporadaFiraRam	**Idioma**	IdiomaCastella IdiomaCatala IdiomaEnglish
HorariObertura	HorariOberturaCarrefour HorariOberturaCastellBellver HorariOberturaFiraRam HorariOberturaHotelRiuPortoPi HorariOberturaPachaPalma	**LocalitzacioCoordenadesGPS**	CoordenadesGPSfiraRam CoordenadesGPSFontTortugues CoordenadesGPSONOEstadi CoordenadesGPSPacha CoordenadesGPSPlatjaPalma

			CoordenadesGPSPortoPiCentre CoordenadesGPSCastellBellver CoordenadesGPSAjuntamentPalma
LocalitzacioDireccioPostal	DireccioPostalCarrefourPortoPi DireccioPostalBasilicaSantFrancesc DireccioPostalHotelRiuPortoPi DireccioPostalAjuntamentPalma	**Entrada**	EntradaCastellBellver EntradaCinemaPortoPiNormal EntradaCinemaPortoPiReduida EntradaPachaPalma
ComercialCentreComercial	ComercialCentreComercialPortoPi	**AllotjamentHotel**	HotelRiuPortoPi HotelRiuSantaMaria
ComercialHipermercat	ComercialHipermercatCarrefourPortoPi	**TransportEstacio**	EstacioCentral
ServeiAjuntament	AjuntamentPalma	**EspaiNaturalPaltja**	PlatjaDePalma
RestauracioRestaurant	McDonaldsPortoPi	**HistoricCastell**	CastellBellver
OciAtraccions	ParcAtraccionsFiraRam	**HistoricFont**	FontSesTortugues
ReligiosBasilica	ReligiosBasilicaSantFrancesc		

Tests realitzats

El primer test que s'ha realitzat s'ha fet abans de la instanciació i inclusió d'individus amb una opció inclosa dins el Protégé que permet verificar la integritat de l'ontologia fent diverses comprovacions respecte l'estàndard del subllenguatge. Aquesta opció es troba en el menú "*OWL Run Ontology Tests*". No es detecta cap inconsistència.

Proves amb el raonador

Per als següents tests a realitzar en el que es durà a terme el raonament de l'ontologia és necessari adquirir una eina que actuï com a raonador sobre per a Protégé i, com s'ha explicat a l'apartat de eines a utilitzar, hi ha una eina anomenada RacerPro que es pot connectar amb Protégé per a l'anàlisi i raonament de les ontologies creades.

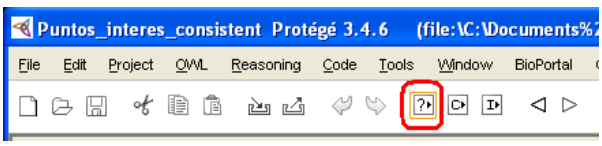
RacerPro s'utilitzà per a tres activitats principals que seran:

- Verificació de consistència: Permet controlar que hi hagi consistència entre les propietats d'objectes i restriccions assignades a les classes.
- Classificació taxonòmica: Troba la jerarquia d'inferència de l'ontologia basada en la jerarquia de asserció, la qual és la classificació que s'obté en la construcció de l'ontologia.
- Computació de tipus inferits: Permet computar els tipus d'objectes dins l'ontologia.

Aquestes funcions s'executen mitjançant icones a la barra d'eines de Protégé un pic instal·lat el raonador.

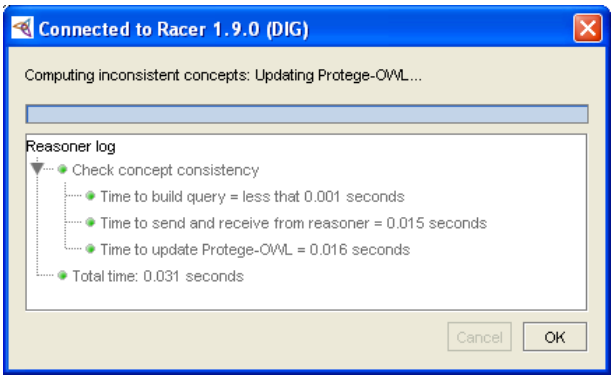
Verificació de consistència

Per llançar el test de consistència de l'ontologia s'utilitzarà el icona situat a la barra d'eines de Protégé amb un interrogant:



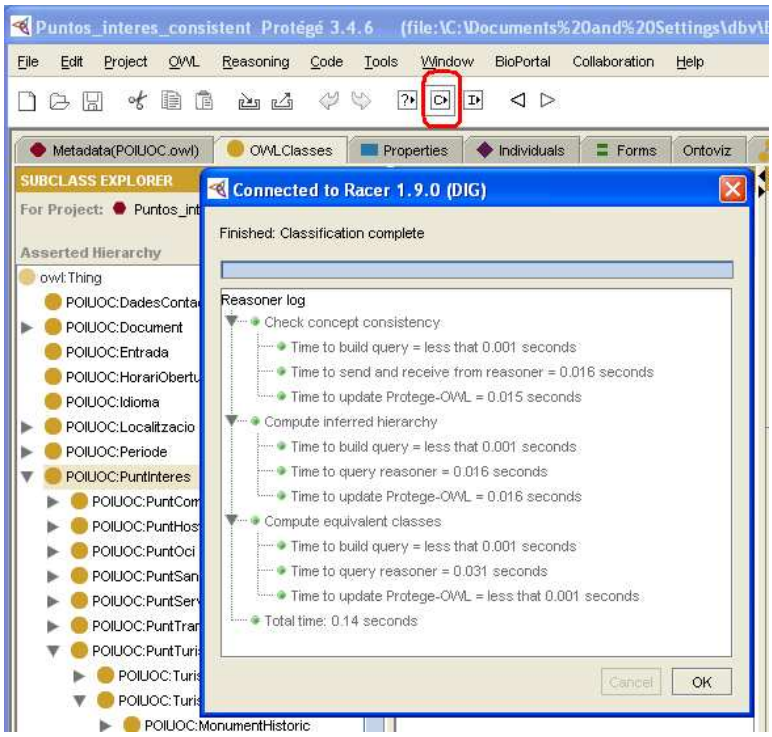
Amb el RacerPro s'analitzarà l'ontologia creada i es definirà la jerarquia de classes i propietats inferides. L'anàlisi de les jerarquies inferides determina la coherència del model i en el cas de l'ontologia creada en aquest treball, el model inferit pel RacerPro és exactament igual al model definit.

Un pic realitzat el test es pot veure que el resultat també és correcte:



Classificació taxonòmica

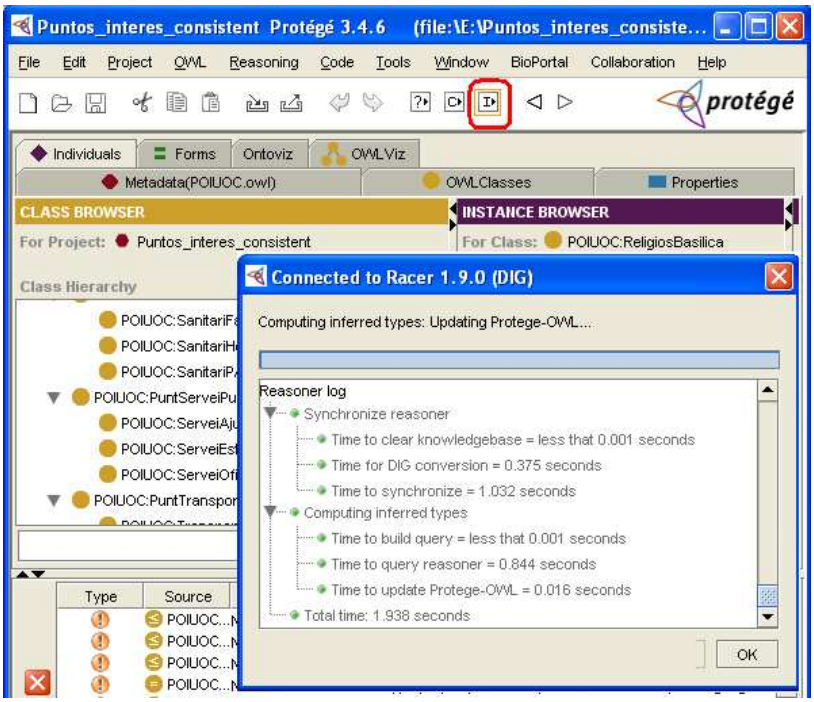
La icona central amb una "C" és el que permet fer la classificació taxonòmica que és l'operació més important de RacerPro. En realitzar aquesta operació, Protégé genera una jerarquia de inferència en l'ontologia.



Com es pot veure els anàlisi que fa són incrementals i abans de realitzar la classificació taxonòmica torna a comprovar la consistència de l'ontologia.

Computació de tipus inferits

La darrera operació que queda per fer és la de realitzar la computació dels tipus inferits que es farà a través del icona "I"



Dir que existeixen certes restriccions que no han pogut esser inferides per RacerPro com són les cardinalitats(maxCardinality, minCardinality, cardinality) o el tipus de propietat float utilitzat per la propietat de la classe Entrada per culpa de que són restriccions pròpies de OWL Full i degut a que RacerPro només pot inferir fins OWL DL.

Interrogació de l'ontologia amb SPARQL

Un altre test que es pot realitzar sobre l'ontologia per a veure si és funcional i s'ha compleix els objectius que s'havien plantejat és la d'interrogar-la utilitzant SPARQL.

Aquestes consultes es poden realitzar a través del mateix programa Protégé, ja que té una consola per a poder executar-les directament.

S'han utilitzat algunes de les preguntes que plantejades per crear l'ontologia ajustant-les a les instàncies creades per intentar obtenir-ne resultats interrogant-la. Si és possible, en lloc de demanar totes les dades, es mirarà d'acotar el resultat seleccionant només un grup de slots:

Hi ha qualque arxiu de vídeo de platges?

Equivalent SPARQL:

```
Query
SELECT ?Iloc ?Nom ?NomDoc ?NomExt ?NomDocA ?NomExtA
WHERE {
    ?Iloc rdf:type POIUOC:EspaiNaturalPlatja.
    ?Iloc POIUOC:hasPuntInteresNom ?Nom.
    ?Iloc POIUOC:hasPuntInteresDocumentAssociat ?document.
    ?document rdf:type POIUOC:DocumentVideo.
    ?document POIUOC:hasDocumentNom ?NomDoc.
    ?document POIUOC:hasDocumentExtensio ?NomExt.
   OPTIONAL {
    ?document POIUOC:hasDocumentVideoAudioAssociat ?audio.
    ?audio rdf:type POIUOC:DocumentAudio.
    ?audio POIUOC:hasDocumentNom ?NomDocA.
    ?audio POIUOC:hasDocumentExtensio ?NomExtA.
    }
 }
Execute Query
```

Aquesta pregunta retornarà el nom i l'extensió de tots els arxius de vídeo associats amb qualque platges, si té el vídeo àudios associats també els mostrarà.El resultat és el següent:

Results

Iloc	Nom	NomDoc	NomExt	NomDocA	NomExtA
POIUOC:PlatjaDePalma	Platja de Palma	platjadepalma	avi	platjadepalma	wav

A on està l'Ajuntament de Palma?

Equivalent SPARQL:

```
Query
SELECT ?NomIloc ?carrer ?pis ?municipi ?provincia ?pais ?latitud ?longitud
 WHERE {
      ?subject rdfs:subClassOf POIUOC:Localitzacio.
      ?loc rdf:type ?subject.
 OPTIONAL{
   ?loc rdf:type POIUOC:LocalitzacioDireccioPostal.
    OPTIONAL {?loc POIUOC:hasDireccioPostalCarrer ?carrer}
    OPTIONAL {?loc POIUOC:hasDireccioPostalPis ?pis}
    OPTIONAL {?loc POIUOC:hasDireccioPostalMunicipi ?municipi}
    OPTIONAL {?loc POIUOC:hasDireccioPostalProvincia ?provincia}
    OPTIONAL {?loc POIUOC:hasDireccioPostalPais ?pais}
    ?loc POIUOC:isDireccioPostalPuntInteres ?punt.
 }
OPTIONAL{
    ?loc rdf:type POIUOC:LocalitzacioCoordenadesGPS.
    OPTIONAL{?loc POIUOC:hasCoordenadesGPSLongitud ?longitud.}
    OPTIONAL{?loc POIUOC:hasCoordenadesGPSLatitud ?latitud.}
    ?loc POIUOC:isCoordenadesGPSPuntInteres ?punt.
 }
  ?punt POIUOC:hasPuntInteresNom  ?NomIloc.
  ?punt POIUOC:hasPuntInteresNom  "Ajuntament de Palma"^^xsd:string.
}
Execute Query
```

Aquesta pregunta retornarà les dades de les possibles localitzacions del Ajuntament de Palma. Una Localització pot esser una direcció postal o unes coordenades GPS. El resultat és el següent:

Results

NomIloc	carrer	pis	municipi	provincia	pais	latitud	longitud
Ajuntament de Palma	Plaça de Cort 1	2°	Palma	Illes Balears	Espanya		
Ajuntament de Palma						4°34"	5°12"11'

Quines són les dades de contacte del Hotel Riu Porto Pi?

Equivalent SPARQL:

```
Query
SELECT  ?nompunt ?tel ?fax ?email ?web
WHERE {
        ?dc rdf:type POIUOC:DadesContacte.
      OPTIONAL{?dc POIUOC:hasDadesContacteFax ?fax.     }
      OPTIONAL{ ?dc POIUOC:hasDadesContacteTelefon ?tel.}
      OPTIONAL{ ?dc POIUOC:hasDadesContacteCorreuElectronic ?email.}
      OPTIONAL{ ?dc POIUOC:hasDadesContactePaginaWeb ?web.}
         ?dc  POIUOC:isDadesContactePuntInteres ?punt.
         ?punt POIUOC:hasPuntInteresNom  "Hotel Riu Porto Pi"^^xsd:string.
         ?punt POIUOC:hasPuntInteresNom  ?nompunt
}
Execute Query
```

Aquesta pregunta retornarà totes les dades de contacte que trobi del punt d'interès amb nom "Hotel Riu Porto Pi". El resultat és el següent:

Results

nompunt	tel	fax	email	web
Hotel Riu Porto Pi	902335566		atencioclienti@riu.com	http://www.riu.com
Hotel Riu Porto Pi	971345522	971205566	riuportopi@riu.com	http://www.riu.com/portopi

Què puc trobar a prop del Ajuntament de Palma?

Equivalent SPARQL:

```
Query
SELECT REDUCED ?Nomlloc  ?Nomlloc2
 WHERE {
       ?subject rdfs:subClassOf POIUOC:Localitzacio.
       ?loc rdf:type ?subject.
       ?loc POIUOC:isLocalitzacioAprop ?loc2
 OPTIONAL{
   ?loc2 rdf:type POIUOC:LocalitzacioDireccioPostal.
    ?loc2 POIUOC:isDireccioPostalPuntInteres ?punt2.
 }
OPTIONAL{
    ?loc2 rdf:type POIUOC:LocalitzacioCoordenadesGPS.
    ?loc2 POIUOC:isCoordenadesGPSPuntInteres ?punt2.
 }
 OPTIONAL{
   ?loc rdf:type POIUOC:LocalitzacioDireccioPostal.
   ?loc POIUOC:isDireccioPostalPuntInteres ?punt.
 }
OPTIONAL{
    ?loc rdf:type POIUOC:LocalitzacioCoordenadesGPS.
    ?loc POIUOC:isCoordenadesGPSPuntInteres ?punt.
 }
  ?punt POIUOC:hasPuntInteresNom  ?Nomlloc.
  ?punt POIUOC:hasPuntInteresNom  "Ajuntament de Palma"^^xsd:string.
  ?punt2 POIUOC:hasPuntInteresNom  ?Nomlloc2.
}
Execute Query
```

Aquesta pregunta retornarà el nom dels punts d'interès que estiguin a prop del Ajuntament de Palma. El resultat és el següent:

Results

Nomlloc	Nomlloc2
Ajuntament de Palma	Font de Ses Tortugues
Ajuntament de Palma	Basílica de Sant Francesc
Ajuntament de Palma	Fira del Ram

Quins comerços hi ha a Porto Pi Centre?

Equivalent SPARQL:

```
Query
SELECT REDUCED ?Nom
WHERE {
      ?subject rdfs:subClassOf ?object.
      ?loc rdf:type ?subject.
      ?loc POIUOC:isPartCentreComercial ?centre.
      ?loc POIUOC:hasPuntInteresNom ?Nom.
      ?centre POIUOC:hasPuntInteresNom  "Porto Pi Centre"^^xsd:string.
}
Execute Query
```

Aquesta pregunta retornarà el nom dels punts d'interès que formin part del Centre Comercial "Porto Pi Centre". El resultat és el següent:

Results

Nom
McDonalds Porto Pi
Carrefour Porto Pi
Cinesa Porto Pi

Quan obre la Fira del Ram?

Equivalent SPARQL:

```
Query
SELECT ?dia ?hini ?hfi  ?tempnom ?tempini ?tempfi
WHERE {
      ?hor POIUOC:isHorariOberturaPuntInteres ?punt.
      ?hor POIUOC:hasHorariOberturaNom ?nom.
      ?hor POIUOC:hasHorariOberturaDiaSetmana  ?dia.
      ?hor POIUOC:hasHorariOberturaPeriodeHores ?per.
      ?per POIUOC:hasPeriodeHoresHoraInici  ?hini.
      ?per POIUOC:hasPeriodeHoresHoraFi     ?hfi.
      OPTIONAL {
                ?hor POIUOC:hasHorariOberturaTemporada ?temp.
                 ?temp POIUOC:hasTemporadaNom  ?tempnom.
                ?temp POIUOC:hasTemporadaPeriodeDates ?pdat.
                 ?pdat POIUOC:hasPeriodeDatesDataInici  ?tempini.
                 ?pdat POIUOC:hasPeriodeDatesDataFi     ?tempfi.

      }
      ?punt POIUOC:hasPuntInteresNom "Fira del Ram"^^xsd:string.
}
Execute Query
```

Aquesta pregunta retornarà els horaris d'obertura de la "Fira del Ram" incloent les temporades de dates en que estan vigents. El resultat és el següent:

Results

dia	hini	hfi	tempnom	tempini	tempfi
dimarts	08:00:00	22:00:00	Temporada d'obertura de la Fira del Ram	2011-05-15	2011-03-14
dissabte	08:00:00	22:00:00	Temporada d'obertura de la Fira del Ram	2011-05-15	2011-03-14
divendres	08:00:00	22:00:00	Temporada d'obertura de la Fira del Ram	2011-05-15	2011-03-14
diumenge	08:00:00	22:00:00	Temporada d'obertura de la Fira del Ram	2011-05-15	2011-03-14
dilluns	08:00:00	22:00:00	Temporada d'obertura de la Fira del Ram	2011-05-15	2011-03-14
dijous	08:00:00	22:00:00	Temporada d'obertura de la Fira del Ram	2011-05-15	2011-03-14
dimecres	08:00:00	22:00:00	Temporada d'obertura de la Fira del Ram	2011-05-15	2011-03-14

Quant val entrar a la discoteca Pacha?

Equivalent SPARQL:

```
Query
SELECT ?Nom ?preu
WHERE {
    ?ent POIUOC:isEntradaAssociadaPuntInteres ?punt.
    ?ent POIUOC:hasEntradaNom ?Nom.
    ?ent POIUOC:hasEntradaPreu ?preu.
    ?punt POIUOC:hasPuntInteresNom "Pacha Palma"^^xsd:string.
}
Execute Query
```

Aquesta pregunta retornarà les possibles entrades amb el seu preu per a "Pacha Palma". El resultat és el següent:

Results

Nom	preu
Entrada Pachà estàndard	15.0

Hi ha qualque àudio en català?

Equivalent SPARQL:

```
Query
SELECT ?Nom ?Ext
WHERE {
    ?id rdf:type POIUOC:Idioma .
    ?id POIUOC:hasIdiomaDocument ?doc.
    ?doc rdf:type POIUOC:DocumentAudio.
    ?doc POIUOC:hasDocumentNom  ?Nom.
    ?doc POIUOC:hasDocumentExtensio  ?Ext.
    ?id POIUOC:hasIdiomaNom "Català"^^xsd:string.
}
Execute Query
```

Aquesta pregunta retornarà tots el fitxers d'àudio en català. El resultat és el següent:

Results

Nom	Ext
explicaciosantfrancesc	wav
PromoHogans	wav

Previsions de futur

En quant als futurs treballs amb l'ontologia creada, seria convenient comprovar la seva utilitat a nivell pràctic un pic acabada la fase teòrica presentada amb aquest projecte. Un opció seria a integrar-la a un projecte global de aglutini totes les branques de les ontologies geogràfiques turístiques, per aportar la integració dels punts d'interès amb documents relacionats per demostrar la seva utilitat real ajudant a turista o viatger a en les seves recerques turístiques.

Aquesta utilització pràctica podria realitzar-se mitjançant la seva implantació en un model real, podent formar part be d'un web o portal turístic, o d'un prototipus creat per a tal efecte, de tal forma que pugui servir i proporcionar als usuaris documents o arxius sobre els punts d'interès que sobre els que s'hagin realitzat recerques, formant part en el modelatge de la informació relacionada amb el tema i facilitant el procés de recerca de punts i de documents relacionats amb ells, que es pugin presentar com a resultats a l'usuari. Aquest portal actuaria sobre les instàncies relacionades amb les categories de l'ontologia i n'extrauria resultats segons les cerques demandades per l'usuari.

Un altre opció de futur seria que, en lloc d'integrar-se en una ontologia més global, es podria plantejar la extensió de la creada agrupant altres punts que no es contemplen en ella com a podrien esser les rutes entre punts d'interès, esdeveniments, etc... Per tal de poder crear-ne una completa que cobreixi totes les necessitats i els àmbits que cobreixen i s'han explicat als apartats d'estat de les ontologies.

Conclusions

En aquest projecte s'ha presentat el desenvolupament d'una ontologia dins del domini específic dels punts d'interès turístic que tenen una sèrie de documents associats, abastant des de la fase d'identificació de la informació a representar, fins una prova de concepte realitzada amb les instàncies creades dintre de l'ontologia creada.

Durant aquest projecte s'han pogut treure els conclusions següents

- La web semàntica es pot considerar una extensió del web de l'actualitat que permet utilitzar noves tecnologies per millorar l'ús i accés dels recursos del web fent ús de conceptes com OWL, que suporten un llenguatge ontològic per definir venir i classificació car conceptes en un domini, en aquest cas particular en un domini geogràfic turístic. Aquests es poden implementar mitjançant serveis web.

- En base a les definicions d'ontologia trobades i aplicades al àmbit que ens ocupa, es pot concloure que una ontologia és una jerarquia de conceptes amb atributs i relacions que crea una classificació per definir la semàntica d'unitats d'informació interrelacionades. Les ontologies proveeixen una comprensió d'un domini que permetrà comunicar persones i sistemes heterogenis cosa que en l'àmbit turístic és d'una necessitat imperiosa pel volum de negoci que genera actualment .

- Al capítol referent al l'estat de l'art de els ontologies geogràfiques s'aconsegueix veure que és un àmbit bastant madur i que existeix també una gran quantitat d'informació al respecte, informació que queda plasmada en un gran nombre d'ontologies, algunes d'elles descrites ell. Es demostra amb això la importància i el bon nivell existent en el desenvolupament dintre d'aquest camp, impulsat en molts casos des d'organismes de internacionals rellevants com la Unió Europea o governs regionals. Durant el desenvolupament d'aquest projecte s'ha pogut apreciar que també que les tecnologies relacionades amb la web Semàntica, no es queden enrere en quant al seu desenvolupament en els darrers anys, sobretot amb l'entrada de llenguatge OWL , han rebut un gran impuls que els hi permet dotar a les dades d'un major contingut semàntic a través de diferents tipus de propietats com la simetria, la transitivitat o la reflexivitat, difícilment representables amb eines anteriors.

- En quant a les eines utilitzades, es pot concloure que Protégé és una bona eina de desenvolupament d'ontologies que permet la construcció d'ontologies OWL tenint en compte totes les característiques de un domini. Per mitjà d'aquest programa es poden elaborar les propietats i restriccions necessàries per a permetre generar inferència sobre el domini, generant una conceptualització i coneixement del mateix. Tenim també els

programes OWLViz i OntoViz com plugins de Protégé m'han permès realitzar representacions gràfiques de l'ontologia ja siguin del model de asserció o del d'inferència. A més ens proporciona un entorn integrat per a poder realitzar interrogatoris a ontologies a través del llenguatge de consultes SPARQL que he pogut fer servir a durant les proves..

- Instanciant l'ontologia, es pot veure la utilitat de l'ontologia per si mateixa en el domini facilitant la possibilitat de crear un web o portal per a que els usuaris puguin realitzar recerques d'informacions sobre punts d'interès a l'hora de realitzar un viatge i presentar-se-les completada amb documents escrits, audiovisuals i referències webs per tal augmentar la qualitat de la informació sobre els punt trobats. A més existeix la possibilitat d'integrar aquesta ontologia amb altres d'àmbit més genèric com les presentades al apartat d'estat de l'art de les ontologies per ampliar-ne la cobertura de les mateixes.

- Gràcies a RacerPro i les consultes amb SPARQL s'ha pogut realitzar de forma ràpida i fàcil la verificació de l'ontologia tenint en compte elements de consistència, a més de realitzar la classificació taxonòmica anomenada jerarquia d'inferència, derivada del model construït a l'ontologia anomenat jerarquia d'asserció, acabant amb una comprovació empírica de que les consultes plantejades inicialment al hora de fer l'ontologia poden esser, efectivament, resoltes amb ella. No seria difícil a través d'aquestes consultes en SPARQL construir un prototip real que les fes servir.

Com a conclusió final dir al llarg del treball realitzat, es pot apreciar que l'ús de ontologies permet millorar la recerca dins de la ingent quantitat d'informació disponible a Internet, ajudant a la seva categorització dins del desordre existent en el món tecnològic i global actual, i també, que aplicant l'ontologia presentada es facilitaria la tasca de desenvolupar aplicacions de recerca en l'àmbit dels punt d'interès mostrant-li al usuari tota la informació relacionada que es trobi disponible, en tots els formats que possibles.

BIBLIOGRAFIA

Material de l'assignatura

- **Material d'exemple proporcionat pel tutor:**
    - **"PFC GIS - PLA DE TREBALL"**
    - **"Creació d'una Ontologia Universal - Pla de treball"**
- **Modúl1 "Gestió i desenvolupament de projectes Conceptes i suggeriments" de l'assignatura e Treball de fi de Carrera de la UOC:** http://materials.cv.uoc.edu/cdocent/LF95ZUJ32Z5BCOQYYULJ.pdf

Bibliografia recomanada

- **OWL, XML:**
    - http://www.w3.org/tr/2008/wd-owl2-primer-20080411/

- http://www.w3.org/TR/owl2-quick-reference/
- http://www.w3.org/TR/xmlschema-2/#datatype
- http://cesarob.blogspot.com/2008/12/protg-owl-tutorial.html

- **Ontologies**
  - http://www.daml.org/ontologies/
  - http://protege.stanford.edu/publications/ontology_development/ontology101-es.pdf
  - http://www.tucamon.es/contenido/creacion-de-ontologias-con-protege
  - http://dguaman04.wordpress.com/2009/07/24/creacion-de-una-ontologia-sobre-una-orquesta-sinfonica/
  - http://www.gsic.uva.es/wikis/yannis/images/7/75/G3_Revision_Davidzayas.pdf
  - http://www.tonahtiu.com/SIS032.pdf

- **Protege Ontology Library:**
  - http://protege.cim3.net/
  - http://protegewiki.stanford.edu/wiki/Main_Page
  - http://owl.cs.manchester.ac.uk/tutorials/protegeowltutorial/resources/ProtegeOWLTutorialP4_v1_3.pdf
  - http://scalab.uc3m.es/~dborrajo/papers/icaps05-ontologies-workshop.pdf
  - http://www.tucamon.es/contenido/creacion-de-ontologias-con-protege

- **Swoogle - Cercador de ontologies:** http://swoogle.umbc.edu/

- **Estat de l'art d'ontologies turístiques**
  - "TOURISM ONTOLOGY AND SEMANTIC MANAGEMENT SYSTEM: STATE-OF-THE-ARTS ANALYSIS: http://www.iadis.net/dl/final_uploads/200712C070.pdf
  - http://idi.fundacionctic.org/cruzar/turismo.html
  - http://qallme.fbk.eu/qallme-tourism4.0.owl
  - http://qallme.fbk.eu/index.php?location=ontology
  - http://www.uni-koblenz.de/~fb4reports/2010/2010_01_Arbeitsberichte.pdf
  - http://www.perey.com/ARStandards/%5BDERI%5DPOIs_in_Semantic_Web.pdf
  - http://www.opentravel.org/
  - http://www.harmonet.org/
  - http://www.w3.org/2005/04/FSWS/Submissions/16/paper.html
  - http://e-tourism.deri.at/ont/docu2004/OnTour%20-%20System%20Design.pdf
  - http://www.fipa.org/specs/fipa00091/PC00091A.html#_Toc511707116
  - http://anota.ibit.org/bloc/index.php?cat=4

- http://www.zaragoza.es/ciudad/turismo/es/visitar/ontologia.htm

- http://www.w3.org/TR/owl-time/#calclock

- **Graphviz i OWLViz**:
  - http://www.graphviz.org/About.php
  - http://www.co-ode.org/downloads/owlviz/OWLVizGuide.pdf

- **RACERPRO**:
  - https://www.racer-systems.com/products/download/dlform-edu.phtml

- **SPARQL:**
  - http://en.wikipedia.org/wiki/SPARQL
  - http://www.w3.org/TR/rdf-sparql-query/
  - http://www.w3.org/TR/2007/CR-rdf-sparql-query-20070614/#modReduced
  - http://librdf.org/query
  - http://www.slideshare.net/Hicham_Qaissi/sparql-1637952
  - http://ontoguate.wordpress.com/2009/01/25/utilizacion-de-sparql-para-consultas-a-ontologias/
  - http://serqlsparql.50webs.com/ejsparql.html

Tesis i Projectes fi de carrera

- **"Creación de una ontologia universal": h**ttp://openaccess.uoc.edu/webapps/o2/bitstream/10609/955/1/36607tfc.pdf

- **"Aplicació d'Ontologies per a aplicacions web del treball en grup":** http://hdl.handle.net/10609/452

- **"Ontologías Turísticas Geográficas : creación de una ontología sobre rutas turísticas (a Pie o en Bicicleta) por espacios naturals":** http://hdl.handle.net/10609/2284

- **"DESARROLLO DE ONTOLOGÍAS SOBRE BBDD RELACIONALES: ONTOLOGÍA QUE SOPORTA LA BÚSQUEDA AVANZADA DE INFORMACIÓN. PROYECTO SIBO Sistema de Información Basado en Ontologías":**http://hdl.handle.net/10016/5493

- **"Creació d'un plug-in de Protégé per a podar ontologies":** http://hdl.handle.net/10609/5881

- **"Ontología del Mapa de la Planta de un Edificio con Protégé-OWL":** http://www.dicc-cid.uji.es/InfTec/reports/ICC%202006-10-01.pdf

- "**Técnicas de depuración e integración de ontologías en el ámbito empresarial":** http://personal.us.es/aparedes/TesisAPM.pdf