

DIG35 v0.1 to v0.2

Gaëtan Martens, Chris Poppe, Erik Mannens

30-9-2007

Multimedia Lab – UGent

Latest revision of this document: November 8, 2007

Legend

[?]: question

[i]: info

+: change

[!]: problem of expressing st in RDF/OWL

[D]: discarded

OWL DL

For this version of the DIG35 vocabulary, we decided **not** to use Foaf Dublin Core, Skoss in order to obtain an OWL DL ontology.

[i] Foaf is *OWL Full*: Output Wonderweb validator:

- Data Properties cannot be inverse functional:
<http://xmlns.com/foaf/0.1/yahooChatID>
- Data Properties cannot be inverse functional:
<http://xmlns.com/foaf/0.1/icqChatID>
- Data Properties cannot be inverse functional:
<http://xmlns.com/foaf/0.1/aimChatID>
- Data Properties cannot be inverse functional:
http://xmlns.com/foaf/0.1/mbox_sha1sum
- Data Properties cannot be inverse functional:
<http://xmlns.com/foaf/0.1/msnChatID>
- Data Properties cannot be inverse functional:
<http://xmlns.com/foaf/0.1/jabberID>
- <http://xmlns.com/foaf/0.1/name> `rdfs:subPropertyOf`
<http://www.w3.org/2000/01/rdf-schema#label>. Only allowed between two DatatypeProperties or ObjectProperties.
- Untyped Class: <http://www.w3.org/2000/01/rdf-schema#Literal>
- Redefinition of: <http://www.w3.org/2000/01/rdf-schema#Literal>
- Even more, many properties and classes of Foaf have status "testing" or "unstable"...

[i] Dublin Core is *OWL Full*: Untyped Object Properties, untyped individuals, `rdf-properties` instead of `owl-properties`.

This has some consequences: eg, when a ObjectProperty or DatatypeProperty is created as a sub-property of dc property (e.g. `dc:identifier`) the WonderWeb OWL validator complains about the fact that `rdfs:subPropertyOf` is "only allowed between two DatatypeProperties or ObjectProperties" as we create a subProperty of a Dublin Core property.

[i] Skos is *OWL Full*: Untyped Object Properties, untyped classes, `rdf-properties`...

See also: Sean Bechhofer, Raphael Volz, *Patching Syntax in OWL ontologies*, International Semantic Web Conference 2004: 668-682

OWL 1.0 shortcomings

[!] unable to express custom datatypes, e.g. a non-negative double

[i] expressing the concept of a primary key (i.e. identifier) can be accomplished using an owl:InverseFunctionalProperty. There is however a restriction using the latter type: an owl:InverseFunctionalProperty is only applicable for object properties. So if one want to use a string or Literal as ID, an extra ID class needs to be created to so. The relation to the original class is of the latter type. The ID class has a data type property, *uid*, which is an owl:FunctionalProperty since the ID class has one unique identifier (the cardinality of *uid* has to be set to 1 because the cardinality of a FunctionalProperty is 0..1)

[i] According to Wonderweb validator a union of an owl class and an xsd:string or rdfs:Literal is not possible, so we need to build another class with a data type property to refer to the string/literal.

0. General changes

- [D] class LangString: xsd:string used instead
- [D] class NonNegativeDouble: xsd:double used instead
- [D] class PositionLocation: use Position and Location
 - + language can be indicated using the xml:lang attribute.
 - + use rdfs:comment instead of a comment property

1. Basic Image Parameter

<BasicParam.owl> (A) OWL DL

+ renamed class BasicImageInformation into BasicParam

[?] there is at most one (and not just one) file name => does it make sense to have metadata and no related image file?

+ the version property has now FormatTypes as domain since it is related to the format and not to the image.

[i] ternary relation: image file, uid & idType -> ImageID class created

[D] property formatVersion

+ new property: version: domain: FormatTypes, range: &xsd:string

+ the property imageID is now an owl:InverseFunctionalProperty since an ImageID must uniquely identify the image

+ the property uid is also an owl:FunctionalProperty since the class ImageID has one unique ID (cardinality of uid is set to 1 because the cardinality of a FunctionalProperty is 0..1)

+ two properties hold the width and height (in pixels) of an image: imageHeight and imageWidth

[D] class ImageSize

[?] is it Ok to have e.g. a width but no height property? Yes!

+ Colorspace renamed into ColorSpace

+ range of property compressionMethod is CompressionTypes

+ preferred presentation parameters are now hold by 2 properties: preferredPresentationHeight & preferredPresentationWidth: their range is xsd:double

[!] preferred presentation parameters should have a range of xsd:double and > 0

[D] class PreferredPresentationSize

+ new class: ColorComponent with instances Red, Green,...

+ new class: ColorComponents a collection of some ColorComponent instances and are related to ComponentDescription class, e.g. "Cyan" is related to CMYK

- + new class: ComponentItem to hold the properties which relate a ColorComponent to its size
- + new property component: domain ComponentDescription, range ColorComponent, componentItem: ,
- + property size has domain ComponentItem (maxCardinality 1)
- + new property colorComponent: domain ComponentItem, range ColorComponents
- + renamed ProfileName instances into CS_Grayscale, CS_sRGB, CS_YCbCr
- + new ComponentDescription instance: YCbCr

2. Image Creation metadata

<Address.owl> (F2.9) OWL DL

- + renamed addressName: addressDescription
- + renamed post: postcode, zip: zipCode
- + properties postcode & zipCode : Cardinality = 1
- + addressCountry: maxCardinality = 1
- + renamed class Country: CountryCode
- + range addressCountry: new class CountrySpecification which is a union of a CountryName and a CountryCode since this property can have a value which is a country code or to the country its name.
- + addressComponentValue: cardinality = 1

<PhoneNumber.owl> (F2.10) OWL DL

- + renamed class Phone: PhoneNumber
- + renamed Phone.owl: PhoneNumber.owl

<EmailWWW.owl> (F2.11 & F2.12) OWL Lite

- + renamed Email: EmailAddress
- + renamed Web: WebAddress
- + properties: cardinality webPageAddress & emailAddress set to 1; and maxCardinality of webPageType & emailAddressType set to 1.

<Organization.owl> (F2.14) OWL

- + new class OrganizationID: since an organization id should uniquely identify a person, the corresponding property should be an InverseFunctionalProperty which is only applicable for ObjectProperties. The property organizationID is now an ObjectProperty which has range OrganizationID

+ new property uid is a FunctionalProperty (domain: OrganizationID, range: string)

+ entity &mail; is renamed into &web;

[D] organizationComment: use rdfs:comment

<Person.owl> (F2.13) OWL DL

+ new class PersonID: since a person id should uniquely identify a person, the corresponding property should be an InverseFunctionalProperty which is only applicable for ObjectProperties
+ PersonID has a FunctionalProperty with cardinality 1: uid.

[i] properties Person:given, Person:nickName , Person:family can be defined as sub-properties or equivalent properties of foaf:firstName, foaf:surname and foaf:family_name or foaf:name. But the latter (foaf) properties have still status "testing".

[D] property personName and class NameComponents. The properties of the latter class now directly relate to class Person.

+ renamed nick: nickname; prefix: prefixName; given: givenName, family: familyName; suffix: suffixName; maide: maidenName;
personOrganizationMember: organizationMember
+ entity &mail; is renamed into &web;
+ birthDate is a FunctionalProperty

<ProductDetails.owl> (F.2.18) OWL DL

+ renamed hasManufacturer into manufacturer; hasModel int model; hasSerial into serial; hasVersion into version
+ minCardinality of manufacturer, model, serial & version changed into maxCardinality

<DateTime.owl> (F.2.8) OWL DL

[D] Comment: use rdfs:comment, class Century: property century range: xsd:integer

[!] century value 0 is not valid (no century 0): can't be expressed in RDF/OWL

+ exactDate is a functional property

<Address.owl> (F.2.9) OWL DL

+ cardinality of addressComponentType = 1

<Location.owl> (F.2.15) OWL DL

[i] range of gpsSpeed, gpsDOP, gpsImageTrack, gpsImageDir, gpsDestBearing, gpsDestDistance, gpsSeconds should be non-negative double

[D] gpsTime: use dig35:timestamp instead; locationComment: use rdfs:comment

<Direction.owl> (F.2.16) OWL Lite

+ GeneralComment replaced by xsd:string

[D] hasDirectionComment: use rdfs:comment

[D] hasDirectionTime: use dig35:timeStamp

<Position.owl> (F.2.17) OWL DL

[!] properties x, y, x1, y1, x2, y2, xEnd, yEnd should be non-negative double.

<ImageCreation.owl> (B) OWL DL

+ new classes: ImageCapture, which will be the superclass for CameraCapture, ScannerCapture, SoftwareCreation.

+ property softwareInfo has domain ImageCapture

+ new property: imageCapture (domain ImageCreation, range ImageCapture)

+ maxCardinality of imageCapture = 1 since a digital image is (originally) created 1) with a camera, or 2) with a scanner or 3) by software

+ property sensorTechnology has domain CameraCapture

+ maxCardinality of cameraLocation = 1. domain of the latter property is CameraCapture

+ focalPlaneResolution, spectralSensitivity, isoSaturation, isoNoise, & spatialFrequencyResponse, cfaPattern, oecf, minFNumber, exposureTime, fNumber, expProgram, brightness, exposureBias, subjectDistance, meteringMode, sceneIlluminant, colorTemp, focalLength, flash, flashEnergy, flashReturn, backlight, subjectPosition, exposureIndex, autoFocus, specialEffect, par: domain = CameraCapture

+ the ordering of the special effects is of importance: new class SpecialEffectItem. The (data type) property effectOrder indicates the ordering and is a FunctionalProperty. The (object) property relates to the type of special effect.

+ range of par & exposureTime: rdfs:Literal since the value is a rational (string notation) or a number (double)

[!] range of isoSaturation, isoNoise, spatialFrequency, horizontalSFR, verticalSFR, outputRedLevel, outputGreenLevel, outputBlueLevel, minFNumber, fNumber, subjectDistance, colorTemp, focalLength, flashEnergy, should be a non-negative double

+ exposureTime range is rdfs:Literal

[!] ExposureTimeValue should be a string OR non-negative double: unionOf is not applicable since doubles and strings are no owl:classes

[i] about property specialEffect: the semantics state a meaning to the ordering of the special effects namely the stacking order of the special effect filters... How to describe this?

[D] scannerSoftwareInfo -> see softwareInfo property of ImageCapture

+ scannerInfo, scannerPixelSize, physicalScanResolution: domain ScannerCapture

[!] range of scannerPixelSize should be non-negative double

+ maxCardinality of scannerInfo, scannerPixelSize, physicalScanResolution = 1

+ maxCardinality of captureItem = 1

+ creationTime is a FunctionalProperty

+ renamed: hasRegionChoice: regionType; RegionChoice: RegionType, hasPositionChoice: positionType; PositionChoice: PositionType

[D] hasPositionComment, hasPositionTime use rdfs:comment and dig35:timestamp

<RectangularSize.owl> (/) OWL Lite

[!] doubleWidth, doubleHeight, integerWidth, integerHeight: should be non-negative

3. Content Description Metadata

<Content.owl> (C) OWL DL

+ renamed: contentThingDescription into thingDescription

<SomeProperty.owl> (C.3.10) OWL DL

+ renamed hasSomePropertyName into name; hasSomePropertyValue into value

[i] if there is a sub-property, there is no value! => So we need a union of a property and a ... string (literal) but this is not possible in owl since a string (literal) is not an owl-class, so we need to build another class, i.e. SomePropertyString, with a corresponding datatype property to hold the string

+ new classes: SomePropertyString & SomePropertyValues (union of SomeProperty and SomePropertyString)

+ range of value is SomePropertyValues

+ datatype property somePropertyString: domain= SomePropertyString, range= xsd:string

+ cardinality of hasSomePropertyString = 1

+ contentCaptureTime is a FunctionalProperty

+ property dictRef is a FunctionalProperty

[D] contentComment & comment: use rdfs:comment

[?] is it a good idea to model a property as a class? To be consistent with the RDF/OWL specification, one should create an owl data type or object property with as rdf:ID the name, as rdfs:range the type of the value and as rdfs:domain the type of the instance to which this property relates.

<PersonDescription> (C.3.5) OWL DL

+ renamed: personAtPositionLocation into person

[D] personHasPositionLocation: new properties position & location, both maxCardinality 1

+ renamed: personHasProperty into personProperty

<TangibleThing.owl> (C.3.6) OWL DL

+ renamed: thingName into name

[D] thingPositionLocation: new properties position & location, thingComment: use rdfs:comment

+ new class ThingID

+ range idTangibleThing = ThingID

- + idTangibleThing is an InverseFunctionalProperty since it is an identifier of TangibleThing
- + uid: domain=ThingID, range=xsd:string and is a FunctionalProperty
- [?] does sub-thing relate to "part of" or to "sub-class of"

<OrganizationDescription.owl> (C.3.7) OWL

- + renamed: organizationAtPositionLocation into organization
- [D] organizationHasPositionLocation: new properties position & location both have maxCardinality 1
- + cardinality of organization is 1.

<Event.owl> (C.3.8) OWL DL

- + new class EventID: an identifier of class Event
- + eventID: maxCardinality = 1; InverseFunctionalProperty, range: EventID
- + uid domain= EventID, range=xsd:string ; FunctionalProperty
- + renamed eventParticipantRole into role
- + renamed hasEventRelation: eventRelation
- [i] EventParticipantRoleType is a union of PersonDescription, OrganizationDescription and TangibleThing. The classes Person and Organization are not needed in the union since PersonDescription & OrganizationDescription have a [0-1] relation (maxCardinality=1) relation to Person and Organization respectively.
- [i] eventRef refers to a related Event and not to its "Description" as in the DIG35 specification
- [D] eventComment: use rdfs:comment

<Dictionary.owl> (C.3.11) OWL Lite

- + renamed hasDictID into dictID
- + new class DictionaryID
- + dictID range = DictionaryID
- + uid domain = DictionaryID, range xsd:string, FunctionalProperty,
- [D] hasDictTime: use dig35:timeStamp instead; dictComment: use rdfs:comment

<Audio.owl> (C.3.9) OWL Lite

- [D] audioComment: use rdfs:comment

+ removed minCardinality of audioFormat, audioMimeType and audioDescription

4. History Metadata

<History.owl> (D) OWL DL

+ new classes ProcessingHintItem, ProcessingHint and ProcessingHints
(=oneOf ProcessingHint...)

+ property processingHintItem: domain=History, range=ProcessingHintItem

+ property processingOrder: domain=ProcessingHintItem, range=xsd:positiveInteger

+ property processingHint: domain=ProcessingHintItem, range=ProcessingHints

[D] comment: use rdfs:comment

[!] ProcessingHints has same properties as processingSummary but the ordering is important + multiple occurrences are possible...
The ordering is expressed using the processingOrder property however no constraints can be imposed to have an unique number within the scope of a History class

5. Intellectual Property Rights Metadata

<IPR.owl> (E) OWL DL

- + IPRPERSON renamed into IPRPerson
- + NameDescription renamed into IPRNameDescription
- [D] class IPRNameChoice & hasIPRNameChoice: use IPRName instead
- [D] class IPRDescription: the corresponding properties now have domain=IPR
- [D] class IPRDates & property hasIPRDates
- [D] class IPRNameReference: use rdf:resource
- [D] hasIPRDescriptionLanguage (use attribute instead)
- + renamed hasIPRName: iprName; hasIPRDescriptionTitle: descriptionTitle; hasIPRDescriptionLegend: descriptionLegend;
- hasIPRCopyright into copyright; hasIPRExploitation: iprExploitation;
- hasIPRIdentification: iprIPRIdentification; hasIPRHistory: iprHistory;
- hasIPRCaption: caption
- hasIPRNameDescription: iprNameDescription;
- hasIPROrganizationDescription: iprOrganizationDescription;
- hasIPRPersonDescription: iprPersonDescription;
- hasIPRDate : iprDate;
- hasIPRExploitationProtection: iprExploitationProtection;
- hasIPRExploitationUseRestriction: iprExploitationUseRestriction;
- hasIPRExploitationObligation: iprExploitationObligation;
- hasIPRManagementSystem: iprManagementSystem;
- hasIPRManagementSystemType : iprManagementSystemType;
- hasIPRManagementSystemID: iprManagementSystemID;
- hasIPRManagementSystemLocation: iprManagementSystemLocation;
- hasIPRIdentificationIdentifier: iprIdentificationIdentifier;
- hasIPRIdentificationLicencePlate: iprIdentificationLicencePlate;
- hasIPRIdentificationIdentifierMode: iprIdentificationIdentifierMode;
- hasIPRIdentificationIdentifierID: iprIdentificationIdentifierID;
- hasIPRIdentificationLPCountry: iprIdentificationLPCountry;
- hasIPRIdentificationLPRegAut: iprIdentificationLPRegAut;
- hasIPRIdentificationLPRegnum: iprIdentificationLPRegnum;
- hasIPRIdentificationLPDate: iprIdentificationLPDate;

- [D] hasIPRTimeStamp: use dig35:timeStamp
- [D] hasIPRDescriptionTime: dig35:timeStamp
- [D] hasIPRDateTime: IPRDate is a subclass of DateTime so there's no need for this property!
- + renamed hasIPRDateTimeDescription into iprDateDescription

[D] hasIPRLanguage, hasIPRIdentificationLanguage,
hasIPRManagementSystemLanguage, hasIPRHistoryLanguage,
hasIPRDateLanguage: use xml:lang attribute
[D] hasIPRManagementSystemTime, hasIPRIdentificationTime,
hasIPRHistoryDate

+ hasIPRNameChoice minCardinality = 1

6. DIG 35

<DIG35.owl> *OWL DL*

+ renamed:

hasBasicImPar: basicParam;

hasImageCreation: imageCreation;

hasContent: imageContent;

hasHistory: imageHistory;

hasIPR: imageIPR;

+ basicParam: cardinality = 1.

+ maxCardinality of imageCreation, imageContent, imageHistory,
imageIPR = 1

+ new datatype property: timeStamp: no domain,
range: xsd:dateTime, FunctionalProperty