Meeting Paris

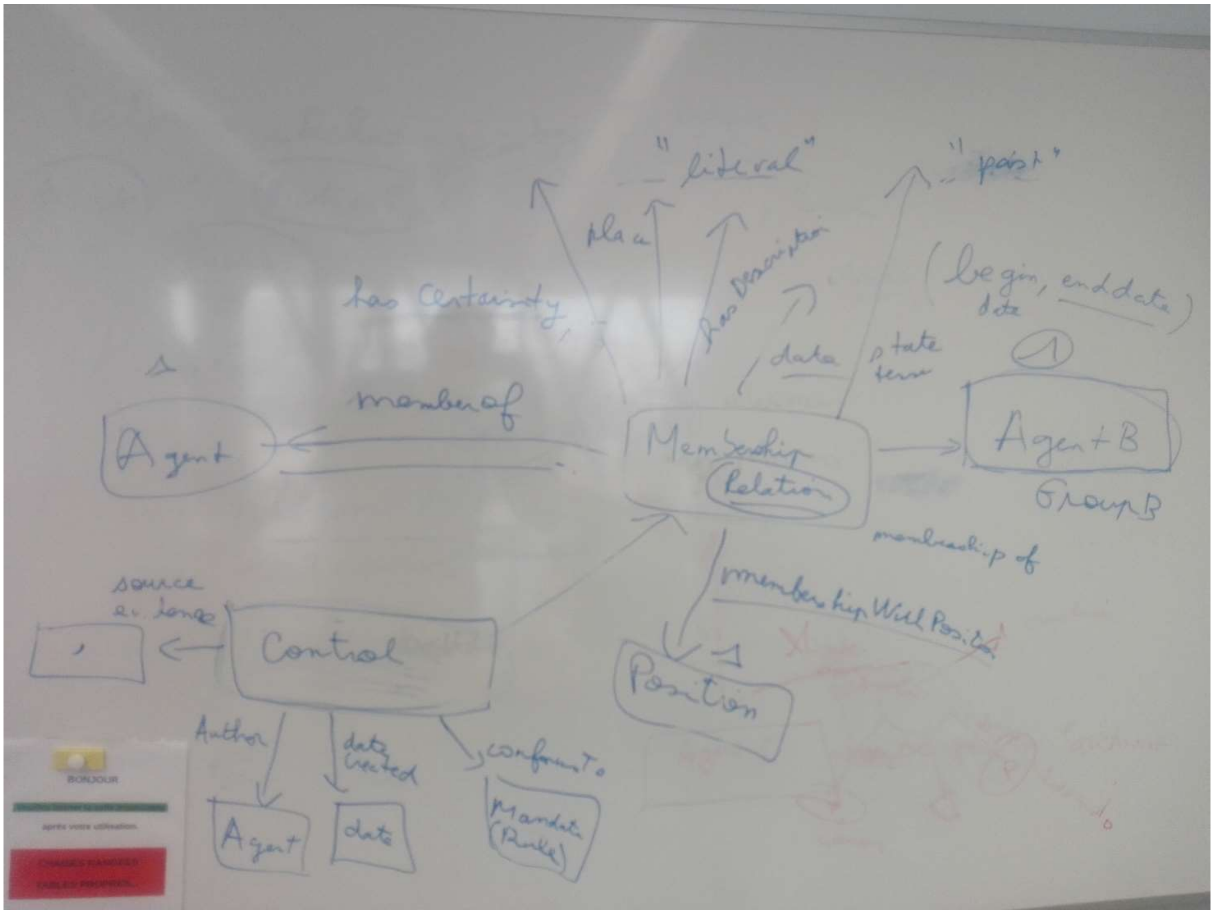
14/02/17

Membership relation

We start looking at the Agent class, with the ontology provided by Florence (ric-o-subset 13/02/2017)

Probably we need a better name than “Trait”-physical etc. But the distinction sounds good.

Florence shows the membership relation and how the model of a relation with associated control information (in the owl file at the moment called “Description” class, probably to be renamed in AssociationDescription). AssociationDescription/Control can be connected to anything.



Relation (superclass) could have date, place, certainly, description (we can discuss if they are data or object properties).

This solution solves the needing of express information about the relation. From the other side, it is extremely verbose.

It’s a sort formalization of the reification process.

Membership relation: 1 agent, 1 group, 0/1 position (if a person has two position, then they are two different relations).

Categorization of relations

Discussion on the “upper” level of relation class. What is a social relation? and how to categorize relations.

Social relation is an Agent2Agent relation but there are other types of relation between agents.

How to call (categorize) Agent2Record relation?

Relation

AgentToAgentRelation

SocialRelation (agent know each other, they have directly contact. The agents know each other, different from “Know of”)

MembershipRelation

AuthorityRelation

educationRelation

familyRelation

influenceRelation (could be social or intellectual, not only agent to agent)

AgentToIdentityRelation (only agent has identity?)

AgentToName

PerformanceRelation (agent to activity, maybe activity)

FulfillmentRelation (Agent to Function)

AuthorityDelegationRelation (Agent to Mandate, maybe Function and activity? N-ary relation or different relations agent to mandate + mandate to function or activity + mandate to record, function + record, function to agent?)

AgentToRecordRelation

RecordToRecord

OrderRelation (generic apply to any entities… arranging, whatever criterium is used. Do we need a class relation for that?)

SuccessionRelation (hasSuccessor/hasPredecessor… apply to any entities? It's a stronger assertion than TemporalRelation, it's not just chronological but also a change of "power")

PartToWhole (distinct to Membership? Membership is a subclass? Or is just the same?)

MembershipRelation
RecordToRecordSet

Probably PartToWhole is different because if removed a part the whole loose is integrity (ie Record to RecordComponent)

Finally, we analyzed RDF example provided by Florence.

Some discussion in the difference between the person and the description of person (what topic maps solved with the distinction between subject identifier and subject locator).

Slight change of names (i.e. Leader of group)

15/02/2017

Categorization of relations (part 2)

InfluenceRelation, two subclass? Direct know “knows”, and undirect “Knows of”? For the first we have SocialRelation for the second we could change the name in KnowledgeRelation. How to manage if someone knows a record?

Rewrote part of the SocialRelation

SocialRelation

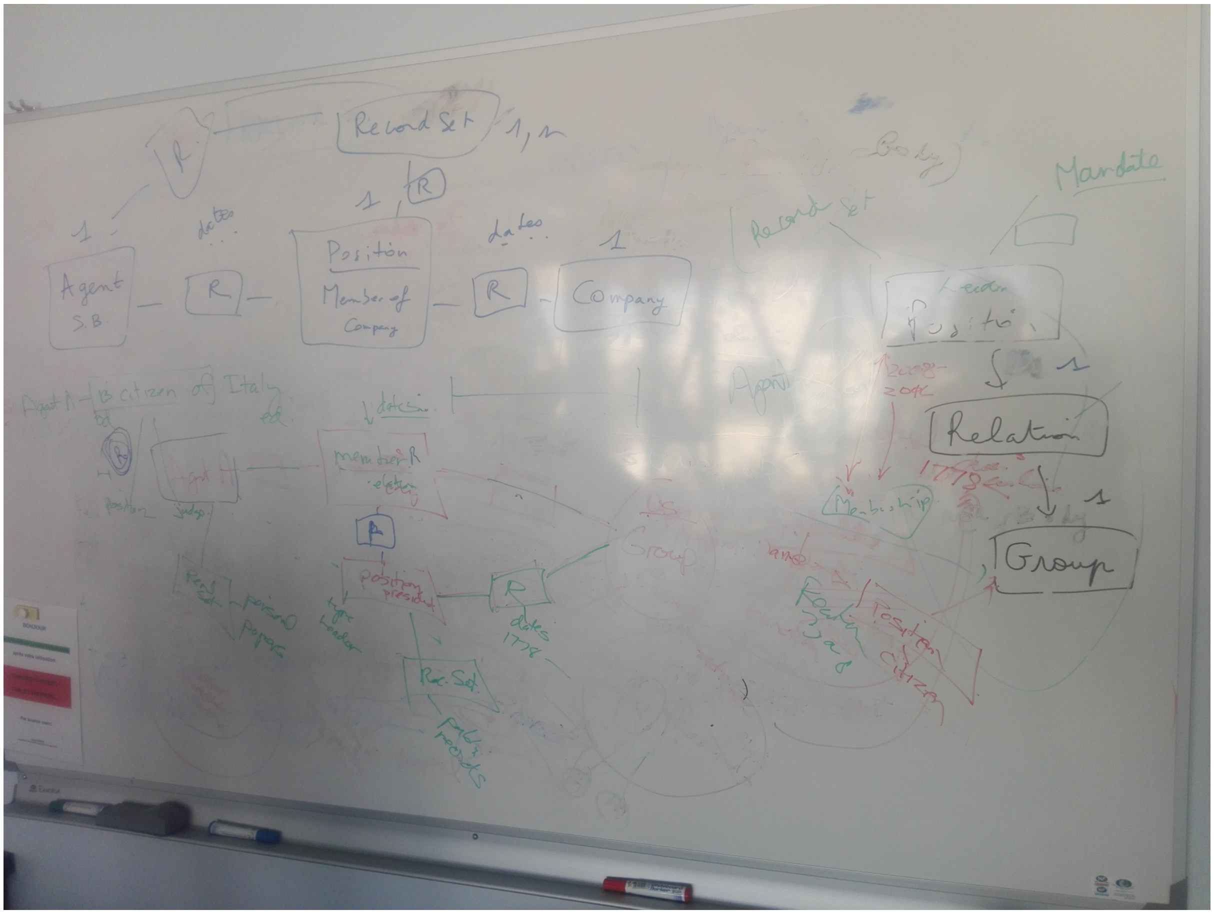
FamilyRelation

AuthorityRelation (controls Agent, owns Agent, person superior to person? Sounds offensive, probably it’s a relation between position or we can used another relation, like ProfessionalRelation)

EducationRelation

InfluenceRelation

Lot of discussion on membership and position, probably we should rewrite all the model for that relation.

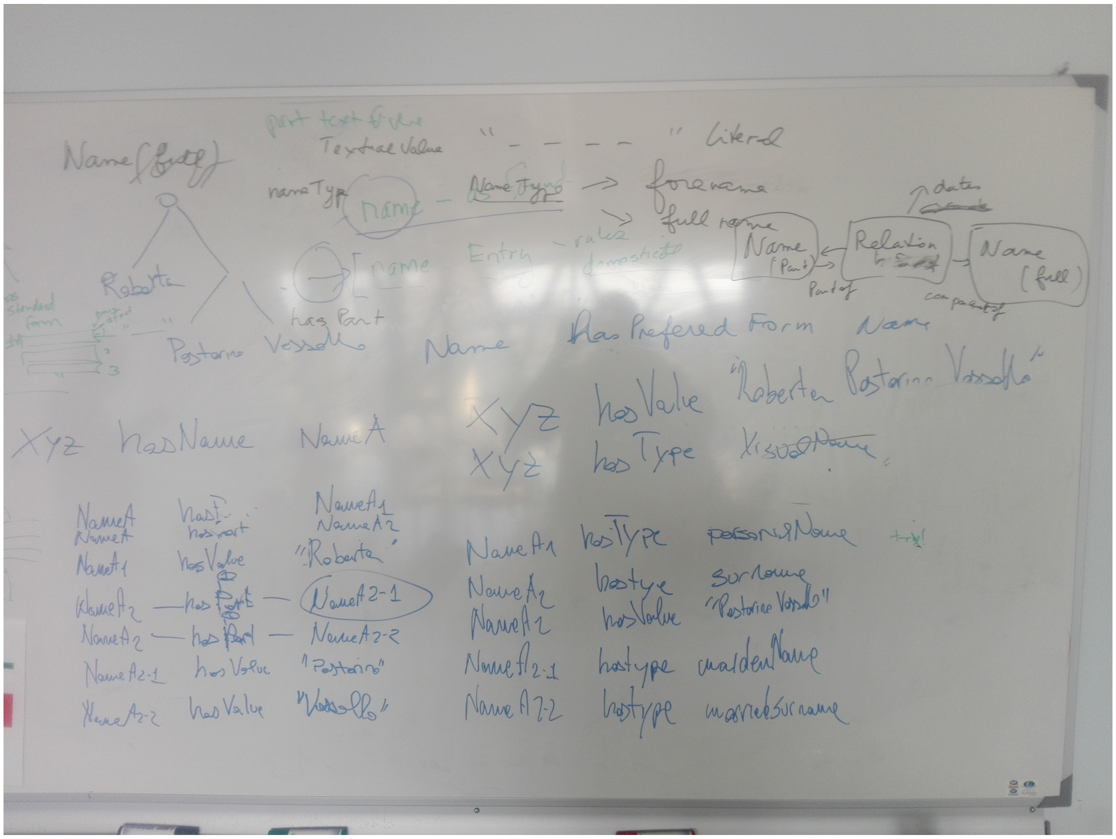


Identifier

PREMIS: use a class Identifier and two data properties hasIdentifierType and hasIdentifierValue

Or the use the same class of name/appellation (as in CIDOC-CRM): YES

Name



Discussion on names, namePart (do we need them?), order of the part, how to express an authorized form of name etc

At the begin we think to have just name but we have to deal with date or “jr” etc that are just component. Rethink the model for that

Record RecordComponent CompoundRecord

Lost track of the discussion ☺

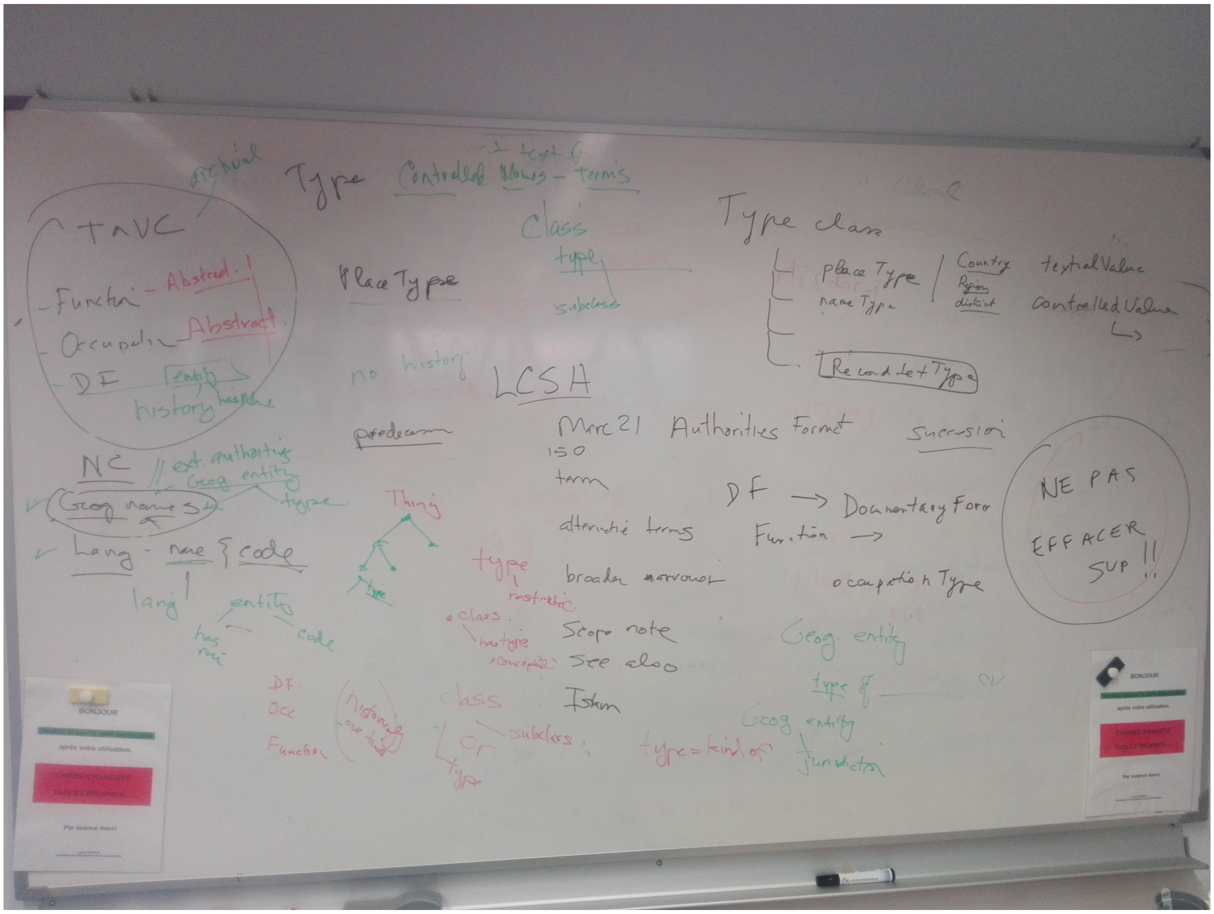
We’ll keep discussing this tomorrow morning.

Type

Used to provide vocabularies. Maybe we can call it Category. We can define individuals for that class or point to an external vocabulary.

Not all “lists” or external vocabularies are under Type. For i.e. Language is not under Type but for the moment under SocialCharacteristic (trait)

Problems with occupations, activities etc because it’s bring the problem of the context (occupation in context vs occupation as abstract). Probably they are not type, we of course need to connect to external vocabularies but it’s unclear if we need to describe the history of occupations with relations and specific classes.


archival
TAVC
Function - Abstract
Occupation - Abstract
DF entity
history has value
Type Controlled Values - terms
Class
type
subclass
PlaceType
no history
predecessor
LCSH
Type class
place Type
name Type
Country
Region
district
textual Value
controlled Value
Record Set Type
Marc 21 Authorities Format
150
term
alternative terms
broader narrower
Scope note
See also
Isbn
DF → Documentary Form
Function →
occupation Type
Succession
NE PAS EFFACER SVP !!
NC
ext. authorities
Geog entity
Geog names
type
Lang - name { code
lang
entity
has name
code
Thing
type
restriction
class
has type
concept
DF
Occ
Function
class
subclass
Or
type
type = kind of
Geog entity
type of
CV
Geog entity
jurisdiction
BONJOUR
après votre utilisation
Par avance merci

16/02/17

Controlled concepts terms

Back to the discussion of yesterday. This is Florence proposal:

Characteristic Class

A superclass for any characteristic of the identity of a thing (we of course have to work on this identifty concept and its domain) e.g. for Name, Function-Abstract, Language, Gender, DocumentaryForm, Activity(Abstract)... like any class :
- hasDescription « a literal » (data property)
- hasName a Name (object property)
- hasHistory « a literal » (at least)
- may have SuccessionRelation and of course if we want so, owl:sameAs is available for linking to, for example, a concept in a SKOS vocabulary.

Name class

Is also a subclass of Characteristic, with generic and specific features
- hasDescription (like any class)
- hasNameType a NameType
- hasPart a NamePart (as said yesterday) (optional)
- hasTextualValue « a literal » (data property)
- hasStandardizedValue

for the hasStandardizedValue property, if we define no range, we allow having any resource as an object, thus, for example, a name in an authority list of names, or a preferred name of a SKOS concept in a SKOS vocabulary, or even a textual value if there is no other solution for a specific instance.

There is still something to discuss (i.e. if we need to express order of NamePart, if hasPart as range NamePart or also Name… see Daniel's example of part sending via email).

Salvatore will take care of that aspect.

Example

Record1 a RiC:Record
Record1 RiC:hasDocumentaryForm Form1
Form1 a RiC:DocumentaryForm
Form1 a RiC:Characteristic
Form1 RiC:hasDescription « blah blah blah »
Form1 RiC:hasName Form1Name
Form1Name a RiC:Name
Form1Name RiC:hasTextualValue 'map'
Form1Name RiC:controlledValue a name defined in an external authority file, or a preferred form of name defined as preferred label of a SKOS concept
Form1 owl:sameAs {a concept defined in an external SKOS vocabulary}

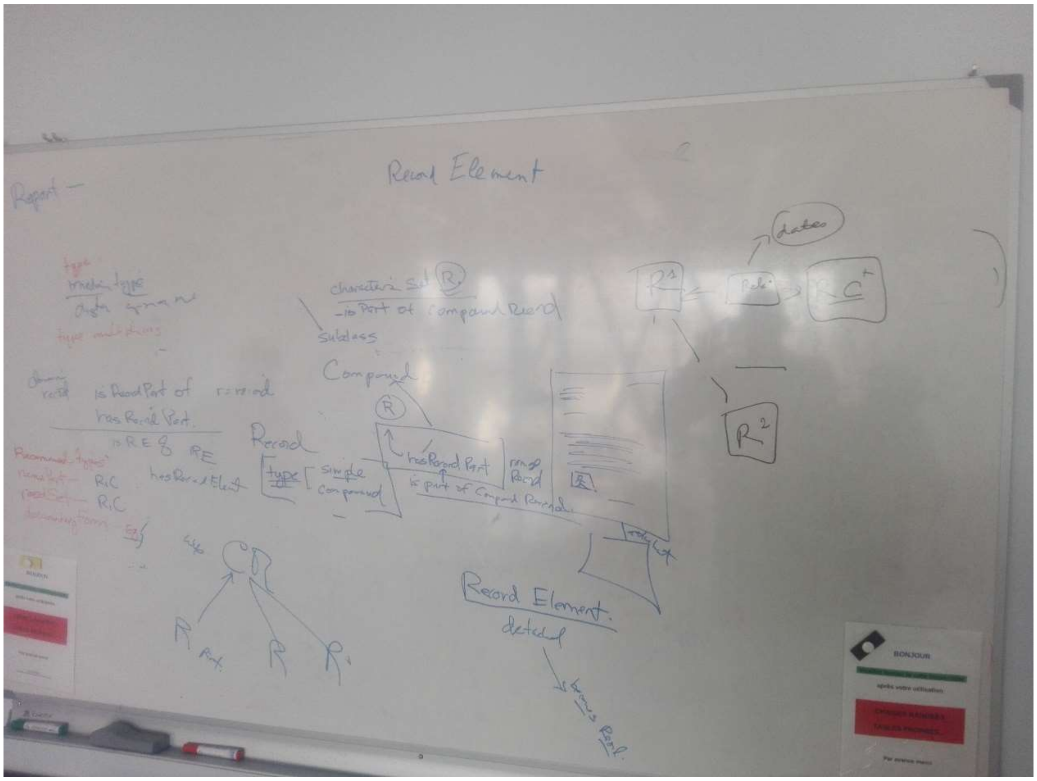
RecordSet, Record, RecordComponent

How to express that an aggregation it's not arbitrary, but it's the expression of provenance etc?

Just using RecordSetType?

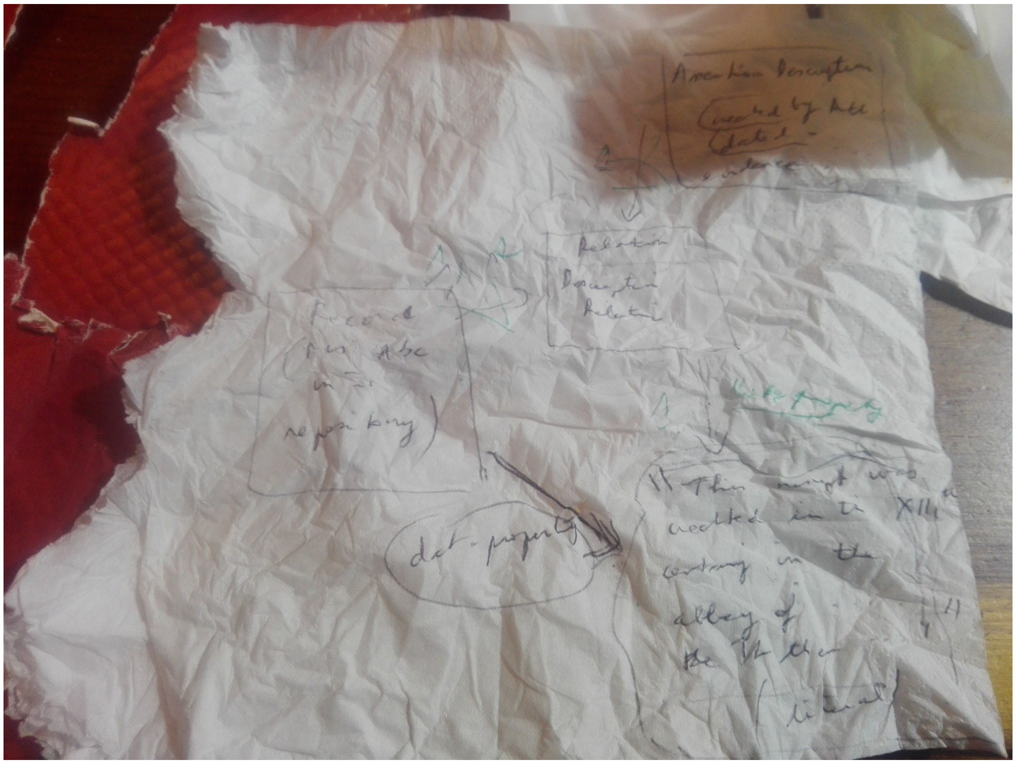
RecordElement instead of RecordComponent with a direct link to the meaning in diplomatic world. Still to check if this solution works in a digital environment.

CompoundRecord do we need a subclass of Record or we can just use Record with a type or just infer that a CompoundRecord is a Record with hasPart relation?



Text Value and element (data properties)

Daniel proposal to rewrite data properties as relations to express control element (AssertionDeclaration) even for data properties. See Florence's note on that.



Test and documentation

Documenting the beta version of the ontology, generating HTML version of the documentation:

- Introduction in prose (Florence will try to write a draft, at the end of march)
- Comments for every entity and property in English (ask Aaron?)
- Diagrams? Automatically generated? See VOWL, Ontograf for i.e. See also http://protegewiki.stanford.edu/wiki/Visualization

Alpha Version, open to a select group of commenters (without introduction)

Beta Version (with introduction)

17/02/17

How to express control elements

We discuss again the question of how to express the control elements even for information such description, history etc (data properties, textual value etc). See also Text Value and element (data properties).

Named Graph?

It seems that Named Graphs are more for big store, in our case you would (could) have a named graph for every assertion (a single triple or a set of triples) and that could be possible but very complex.

In this case NamedGraph would substitute the AssertionDeclaration relation that we created before. Of course, we can create properties for named graph but we can create a set of properties without domain so we can apply to a named graph.

~~In any case either if we use named graph or our assertion declaration if we want to express control elements on every assertion (i.e. description of a record) it should have an URI (so it can't be just a data property).~~ Actually that's not true using named graphs, see mail 20/02/2017

Maybe we can say that AssertionDeclaration is "only" for relation and in the introduction we can say that if we want to say something about the core entities used named graph.

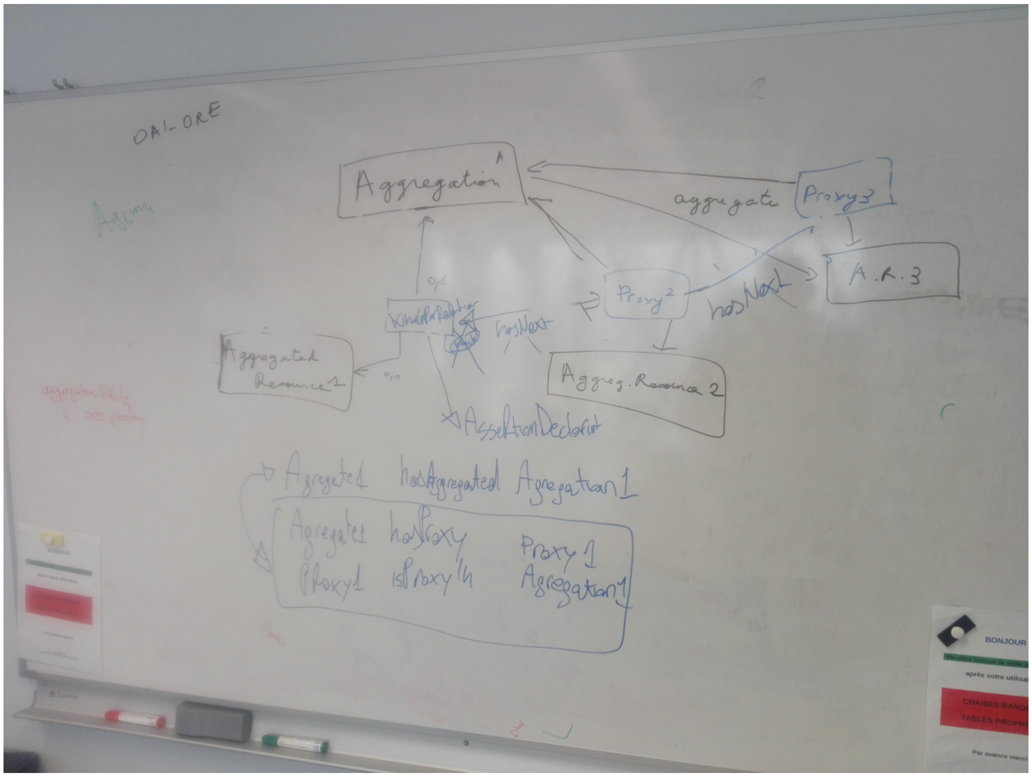
RecordSet and other Sets on the world

We are not going to model all the cultural heritage domain but for any use we create a class Set and define RecordSet as a subclass of Set.

Relation between Record and RecordSet

How to express sequence

OAI-ORE solution:



We can use our WholePartRelation to achieve this result. We don't need a property rank or position (maybe it's useful in local project for some reason but not in the ontology to share with the world). In OAR-ORE you still have the possibility to express direct aggregation without proxy as shortcut. For the moment in our ontology we don't provide such shortcuts (a second way to express the same thing in a concise way).

Record to Record relations

"Genetic" relation

Draft/version and so on. Does it exist a vocabulary (look at VERSIONS project of LSE, unfortunately it seems offline).

Different relations subclass of a generic one or one relation with type?

"Subject" relation

hasPart relation (compound record)

It's an aggregation of 2 or more records that create a new one or there is a main one with some part? (ie email+attachments). We choose the second option.

Record to RecordElement relation

Is it different form hasPart relation between Record and Record? Somehow yes because a RecordElement can't exist without a record. So treat as a subclass

```
hasPart
  hasElementPart
```

NB we want to be able to express sequence of record elements too.

For the moment we'll have different relation between record and record, record and recordSet etc and at the end figure out if there is a common superclass

Seal example

If Seal is on the charter is a recordElement. If it's detached: physically detached is still a RecordElement (because the whole original record still exists even if not physically, as in the digital world), if not treat it as a Record (not in Interpares meaning of curse).

Identities

Example:

Wu Ming, virtual collective association. We can treat as group (even if it's not a group but a collective identity), but sometimes it's treat as a person (in many authority files). It's compound by some Social Identities (Wu Ming 1, Wu Ming 2) of real person ("Roberto Bui" etc).

There are different possible solutions.

1 – SocialIdentity class (Florence OWL) it's not a subclass of Agent

2 – Dolce solution

```
Agent
  PhysicalAgent
    Person
  SocialAgent
    SocialPerson
    Group
```

Etc

Actually the solutions are quite similar because if a SocialIdentities can be connect to records (authors/creators) indeed it "acts" and so it's technically an agent.

Roadmap

First Draft (not public) -> spring (april) or worst scenario at least before the meeting in Rome