

Question and comments to Yangfan's proposal:

Q1: What's the major difference between ontology evolution and traditional schema evolution?

A: Reference Paper: Ontology Evolution: Not the same as schema evolution. See reference.

Q2: Whether partial change (update in place, overwrite) on ontology can be possible or not?

A:

- ✍ Partial Change means only applying the changes requested by the Client and don't make any change to the remaining part of the ontology.
- ✍ For some ontology **languages** (Examples??), if there is no statement about the type of an individual, the individual will be treated as an instance of "Thing" automatically. Then partial change will be acceptable for an ontology described with these languages.
- ✍ In some **applications**, a concept can be deleted only after all its individuals have been defined as instance of another concept, be it ancestor or not.
- ✍ In some other **applications**, no deletion operation is supported. We can only get rid of a concept by merging it to an ancestor, its individuals are turned instance of the ancestor automatically.
- ✍ If no automatic or mandatory type assignment mechanism as above is embedded in the language or applications, we have to consider some additional changes which ensure that only well defined elements are included in the new ontology.

✍ Now the problem is whether constraint between changes is a general phenomenon for ontology languages and applications. Maybe some languages have strong constraint while the others have weak ones or do not have any. (My guess, some further study is needed)

✍ If there is no constraint between changes for all the ontology languages, then we should not consider such a problem in MFI-3. Otherwise, we should define "change constraint" and "change closure".

✍ If there exists a language which has no constraint about changes at all, then we can put the cons cardinality for the relationship between "requested change" and "change closure"

Q3: If the client wants to keep the whole ontology unchanged, what will the consistency constraint be like according to Constraint Model?

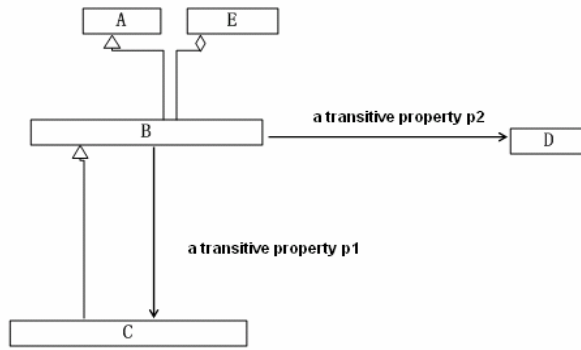
A:

- ✍ If we will keep the whole ontology unchanged, the consistency requirement can be specified as a set which has only one member—the URI of the ontology.
- ✍ Since all the elements are kept, the consistency constraint will be a set consists of the URI of all the component and atomic constructs in the ontology.

Q4: In some cases, the consistency closure should be calculated by adding the concept's children rather than its ancestors. How to get such closure?

A:

- ✍ If we want to get in the ancestors of the concept, the super-sub relationship should be in user's requested consistency.
- ✍ If we need add in the children instead, there must be a transitive property which takes the superclass as domain and the sub class as range.
- ✍ For each class in the user requested consistency, we can get all the related class by means of its properties which are transitive.



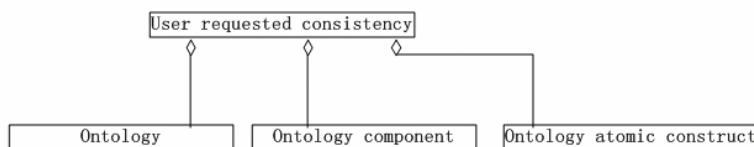
Case of consistency	Consistency closure
B	B
B, superclass (B)	B,A
B,p1	B,C
B,p2	B,D
E, consistsOf	E,B
C, Superclass(C)	A,B,C

- ⌘ For super-sub relationship, it is only possible to navigate from the subclass to super class. So the children will be included only if a transitive property such as p1 is also requested by the user.

Q5: What kinds of consistency can you envision?

A:

- ⌘ Consistency is a concept which should be defined with context. In some cases, we need keep certain features during evolution. But in the other cases, it is possible that these features are not deemed so important to be included in consistency constraint.
- ⌘ As a new concept to MFI-3, consistency should be defined with the elements in Kernel Model. Kernel Model provides a registration facility for the basic content of the ontology by means of ontology atomic construct and ontology component. So we suggested that consistency should be defined as a set of ontology atomic constructs or ontology components, which should be kept in the new ontology.
- ⌘ According to Q3, sometimes we need keep the whole ontology unchanged. We changed the definition of user requested consistency as following.



Q6: How to elicit constraints from the clients?

A:

- ✍ Domain experts and ontology engineers (administrators?) will participate in the process to fix the requirement for ontology evolution.
- ✍ Domain experts may express their requirement by what is needed and what can not be changed in natural language.
- ✍ For “What is needed”, there may be several ways to change the ontology for this kind of requirement. As long as the domain experts think that these changes correctly reflect their requirements, these changes will be treated as an instance of User Requested Change. After that, the ontology engineer will consider how to get change closure with change constraints.
- ✍ The constraint should be adjusted continuingly as the ontology changes.

Q7: Is Q6 in the scope of MFI-3?

A:

In the specification, we need some statement like “The process from user’s coarse requirement to the well defined requirement is beyond the scope of this specification.”

Q8: How to keep Delete Class (Base quantity) with Change Model? If the ontology is described with FOL, the operation should be Delete Predicate (Base quantity)

A:

- ✍ Class (Base quantity) /Predicate (Base quantity) will be registered as an ontology atomic construct.
- ✍ According to all the registration information of the ontology, we can not find out whether the concrete type of the atomic construct.
- ✍ We just know that there are several non logical symbols in the sentence.
- ✍ Details about the type of the atomic construct are omitted.

- ✍ We can registrate evolution related information in the same manner.
- ✍ Both Delete class (Base quantity) and Delete Predicate(Base quantity) will be registered as instance of “Delete “ operation.
- ✍ The arguments about change operation only include change type and ID of the Atomic Construct.
- ✍ The other details can be omitted.

- ✍ For an ontology, whether there is any conflict in it is beyond the scope of MFI-3.
- ✍ In the same sense, whether applying the evolution strategy actually ensures the desired result is beyond the scope of MFI-3 Evolution Management Metamodel.
- ✍ MFI-3 just provides an abstract registration framework for the information which will be used to evolve ontologies.
- ✍ For any evolution, we need requirements about change and consistency. And usually we need some additional changes to meet the two kinds of requirements. So the basic ideas of MFI-3 Ontology Evolution Metamodel can be applied for almost all the ontology evolution processes.
- ✍ But if we would like registration support for ontology specified with a particular language, we have to make some extension to MFI-3 Ontology Evolution Management Metamodel.

Q9: If Base Quantity is deleted, can we keep the individuals of it? Because in some logic language or framework, it is reasonable that individuals stay while corresponding predicate is removed.

A: See Q2.

Q10: Is there any reference papers for this document? If there is, we can share them.

A: See reference

Q11: The proposal is on operational level, so it's not easy for people to understand it. It's also important to express the ideas in the proposal on metamodel level.

A:

We have four models, namely Change Model, Constraint Model, Change Propagation Model and Evolution Information Model for ontology evolution. These models are on metamodel level.

Operations can be registered as instance of "Change".

Q12: Do we need some administrative task or action for evolution?

A:

Administered Item for Ontology Evolution Management Metamodel include

- ✍ Change
- ✍ Change Closure
- ✍ Consistency Closure
- ✍ Evolution Strategy
- ✍ Change Propagation Strategy
- ✍ Evolution Information

Whether the evolution information is trustable is a task beyond the scope of MFI-3. But MFI-3 provides a general framework for the registration of crucial information for this task.

Q13: What kinds of information do the metaclasses you proposed hold?

A: In the future, we will add some attributes to the metaclasses in the metamodel.

Q14: How to get the information about User requested consistency or any other metaclasses you proposed?

A:

- ✍ User requested change and User requested consistency
 - ✍ Domain experts and Ontology engineers (Ontology administrator) cooperate to produce User's requirement for change and consistency.
 - ✍ Domain experts may express their requirements by means of natural languages.
 - ✍ Ontology engineer will provide a set of operations.
 - ✍ Then ontology engineer should show domain experts that these operations can get the result requested by him.
 - ✍ Otherwise, ontology engineer should have some further communication with domain experts until the experts are satisfied.

- ✍ Change closure
 - ✍ We can make the constraint table like that in [Stoj] .
 - ✍ Change closure could be produced by adding all the possible changes caused by user requested change.

- ✍ Consistency closure
 - ✍ For each transitive property in user requested consistency, we can produce a set of classes aggregated with it. See Q4.

- ✍ Evolution strategy
 - ✍ Relationship between change closure, consistency closure and additional changes.
 - ✍ What additional changes are needed is decided by ontology engineer.

- ✍ Propagation strategy
 - ✍ Relationship between Evolution Information, user requested consistency and additional changes.
 - ✍ What additional changes are needed is decided by ontology engineer.

- ✍ Evolution Information
 - ✍ Using information about User requested change, User requested consistency and Evolution Strategy.

Q15: Slide 36, "Rename is a modify change". But in MFI name will be identifier for an atomic construct. When the name changes, it will become totally different thing.

A:

No idea yet.

Q16: Who is the "Client" in Q6?

A: Domain experts.

Reference:

Natalya F. Noy and Michel Klein, Ontology Evolution: Not the Same as Schema Evolution

Michel Klein, PhD Thesis "Change Management for Distributed Ontologies"

M.Sc. Ljiljana Stojanovic, PhD Thesis "Methods and Tools for Ontology Evolution"