A Model for the Collaborative Design of Multi Point-of-View Terminological Knowledge Bases

Gilles Falquet, Claire-Lise Mottaz

Centre universitaire d'informatique, University of Geneva[1]
24, rue du Général-Dufour, CH-1211 Genève 4, Switzerland
Gilles.Falquet@cui.unige.ch Claire-Lise.Mottaz@eti.unige.ch

Abstract

Designing a terminological knowledge base consists in collecting terms and associating them to their definition. Our objective is to define a process model and a formal concept model that supoorts collaborative work. The proposed concept model, which is based on terminological logic and IBIS. The terminological logic part is intended to formally express definitions, while the IBIS part enable users to to express arguments and endorsements on links between terms and definitions. The process model is based on a cyclic conflict resolution process. It includes a formal concept comparison operation, to highlight definition conflicts and their nature, and other operations (derivation, conjunction, disjunction, etc.) to solve the detected conflicts.

1 Introduction

1.1 Background

Terminology is about identifying, describing and naming a field's concepts. Terminology's basic elements are: concepts, terms, definitions and fields. A concept is described by a definition and is named by a term. As a rule, a term can only refer to a single concept within a field. The elaboration of terminological dictionaries and concept bases is generally intended to make translators' job easier or to ensure a better communication between a field's specialists. However, traditional terminological work can also be related to information systems design. In both cases it is about making a list of a field's objects and defining them clearly.

Everyone has his own perception of real world's objects. Thus, when a group of people is building up a concept base or an information system, its members often don't agree on the meaning of the terms, i.e. there are vocabulary conflicts. Surprisingly, although there are many types of concept bases, none of them allows, as far as we know, to store and manage multiple, not necessarily coherent, points of view for a concept's definition without requiring points of view to be coherent. As a result, the choice of a definition or a term must usually be done before it can be inserted into the concept base. So we can say that concept models only allow to store the conceptualization's result but don't directly support the conceptualization process.

1.2 Related Work

Traditional terminology banks, such as Eurodicautom (European Union), Termium (Canada), Lingua-PC (Switzerland, Canton of Bern) or BD-TERM (University of Geneva) [Deb98], [Pul88] represent a first type of concept bases. Concepts are described using textual definitions and other terminological descriptors (synonym, context, source, note). In these terminology banks it could be possible (even if it is not usually done) to store multiple points of view, for instance several definitions for a concept, because the record associated with each term is typically stored as formatted text. But as concept representations are not formalized, it is difficult to apply automatic processing on them.

---

1. This work is a part of a joint project between the CUI and the ETI (School of Translation and Interpretation) at the University of Geneva

In terminological knowledge representation systems, (KL-ONE [Bra91], ALCNR [Buc93], etc.) concepts are characterized by a a set of roles which link them to other concepts in the base. In this case, definitions are not textual but formalized, thus allowing some automatic processing. Nevertheless, in this case we have to face the opposite problem: it is not possible, with this kind of formalism, to handle several definitions for a single concept.

The ConcepTerm model [Ber94], [Sin95] is relatively close to classic terminological knowledge representation systems. Concepts are defined by a set of pairs <characteristic; value>. The goal of ConcepTerm was to enable the search for equivalent terms in different languages by comparing related concepts' definitions. This can give interesting suggestions on how to compare concepts, but this model does not allow to store several definitions for a concept.

The KRL, LOOPS, ROME, VIEWS and TROPES [Mar93] models propose different kinds of solutions for the management of multiple points of view. However, these models all rely on the hypothesis that points of view are partial representations of a unique coherent set of objects. We focus on another situation: when building the concept base, each person (or group of people) has his own incomplete perception of the field; the sum of all individual perception giving an incoherent representation of that field. This difference between basic hypothesis stems from the fact that the model we are presenting in this paper is meant to support group knowledge acquisition and building whereas the others are more adapted to a collective use of already build knowledge.

The Co4 system [Euz96] suggests an interesting approach for the collaborative building of a consensual knowledge base from several individual bases. The bases are organized in a tree in which leaves are the individual bases and each node represents the consensual base of the subtree. The tree's root is the global consensual base. With Co4, the rule is: before inserting a piece of knowledge into a consensual base, one must be sure that all the bases of the subtree agree with it. Co4 is a kind of multi point of view system: knowledge in a consensual base is not the same as knowledge in individual bases. It is however difficult to have a global view, since the different points of view are dispersed in several bases.

Collaboratively designing and building a concept base can also be seen as a decision making process: for each concept it is necessary to choose one definition among those which are suggested by the group members. With this perspective, it is necessary to name three models for decision making support in an argumentative environment: IBIS [Con89], [Con96], [Gro], [Kun72] QOC and DRL [Buc97], [Stu98]. In IBIS for instance, we can see the different "positions" as different points of view. This kind of models will give us a basis for the creation of a multi point of view concept model.

When several points of view are available, it could help to have tools for comparing and manipulating them. So, as we are mainly interested in managing multiple points of view for concepts definitions, we have to quote the works of Shaw and Gaines on conceptual systems comparisons [Sha89]. Since the method of Gaines and Shaw aims at comparing two or more different conceptual systems, it takes into account object names, attributes and values. For instance, it can compare attributes values even if the attributes names do not match.

The following table explains some of the terms that we will use later. It is taken from [Sha89] and indicates the possible situations resulting from the comparison of two ore more conceptual systems.

		Terms	
		Same	Different
Concepts	Same	**Consensus** People use the same terms to name the same concepts	**Correspondence** People use different terms to name the same concepts
	Different	**Conflict** People use the same terms to name different concepts	**Contrast** People use different terms to name different concepts

One can remark that Shaw and Gaines' method is meant to compare two or more different conceptual systems, whereas our main preoccupation is what to do with one incoherent system, build collaboratively. Their method will nevertheless give us suggestions on how to define our concepts comparison operation. These remarks are also applicable to the method presented by Dieng [Die97] for modelling knowledge of multiple experts. (This method is based on the comparison of conceptual graphs.) It is also worth noticing that using differents terminologies doesn't inevitably imply a contrast: maybe people just have a different level of abstraction.

1.3 Our approach

Elaborating definitions for a system's concepts is a difficult task; it often requires long debates, when people don't agree with each other. Thus, when a definition has finally be chosen, it is not always obvious to guess how people made their decision. It could therefore be of real interest to keep the history of the discussion that led to the choice of a definition to the detriment of the others. Allowing the storage of several definitions or "points of view" for a concept could be useful, mainly during the elaboration of the concept base, but also afterwards if a disagreement subsists (particularly if consensus is not strongly required).

In this paper, we chose to present a formal concept model that integrates collaborative work support. In our model, which is based on ConcepTerm and IBIS, it is possible to express arguments on links between terms and definitions. Thus it not only represents knowledge on concepts, but also gives a support for a collaborative building of the concept base.

In a model that enables the management of multiple definitions for a single term, one should be able to compare them. So an operation of comparison has been defined; It highlights conflicts and their nature. (Comparison is done using concepts's attributes and their respective values, as a concept is a pure abstraction and cannot be handled directly) Once a comparison has been carried out, one needs a few other operations to solve the detected conflicts. We have then defined some of these operations and will present them later in this paper.

After the presentation of both the structure and the dynamic part of our model, we will give some suggestions on its use.

1.4 Organization of this paper

The remainder of this paper is organized as follows. Section 2 shows how an argumentative part has been added to a basic concept model. In Section 3, the concept model is presented formally. Section 4 details concepts' comparison as well as other interesting operations. In section 5, we

explain how our model can be used to build a concept base. And finally, section 6 gives a conclusion.

2 IBIS and ConcepTerm integration: ConceptIBIS

When building a terminological concept base, two essential yet reciprocal problems occur: How to define the concept corresponding to a term? What term to use to name a concept with this or that definition? When a group of people is building a terminological concept base, it can lead to several situation corresponding to these two types of problems. Specifically, there can be:

- several different definitions for a single term
- several different terms for a single definition

The main goal of the ConceptIBIS model is to provide a background for 1) highlighting the above-mentioned situations and 2) solving these situations in a multi point of view context. (having a unique point of view is considered as a particular case) In a single point of view context, one roughly tries to come to a single definition for each term and a single term for each definition. (but possibly with synonyms) (reminder: in terminology, the reflection takes place in a specialized field, where terms cannot, in theory, be ambiguous) In a multi point of view context, the different points of view show different ways to describe the same concept by focusing on different characteristics. Thus having more than one definition for a concept doesn't necessarily mean contradiction. For example, it would be easy to accept that a cashier and a mathematician don't define the concept of addition in the same way.

In ConceptIBIS, it is possible to store several definitions for a term only if each of them belongs to a different point of view. The resolution of a definition conflict can lead to the following situations: 1) the two definitions are accepted and each one is linked to a different point of view, 2) one tries to create a single definition from the two conflicting ones, 3) one accept that there are in fact two different concepts (for instance if the definitions are contradictory)

This of course can open philosophical debates. For example, can we admit that two different points of view exist for a single concept? Is a cashier's addition really different from that of a mathematician? Is a fisherman's fish the same as that of a zoologist or restaurant owner? Hopefully, it seems that our model is consistent both with a negative and a positive answer to these questions.

2.1 Structure

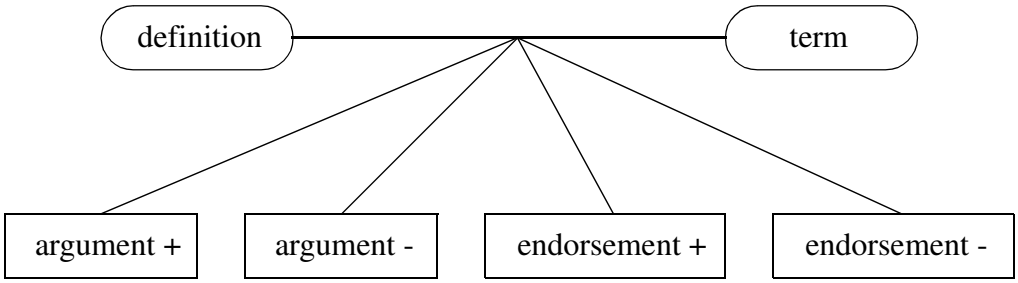
ConceptIBIS is based on ConcepTerm. An argumentative part based on IBIS has been added to enable the management of multiple points of view. A concept is defined by a set of characteristics and by their respective values. Concepts are organized in a hierarchy, using the generalization/ specialization relation.

In IBIS, there are three types of elements: issues (I), positions (P) and arguments (A); these are linked by various kinds of links. When building a terminological concept base, one has to face two types of questions:

a) What is the meaning of term X? (semasiologic process)
b) How to name a concept whose definition is Y? (onomasiologic process)

Then we chose to make the discussion rest on the link between a term and a definition in order to allow to work indifferently with either of this two processes. We have then, simultaneously:

a) I = term, P = definition, A = argument
b) I = definition, P = term, A = argument



In IBIS, it is of course possible to have several positions for an issue. Here, and this is the biggest addition made to a traditional concept model, it is possible to keep several definitions for a single term or several terms for a single definition. It is thus possible to say that we have a multi point of view model.

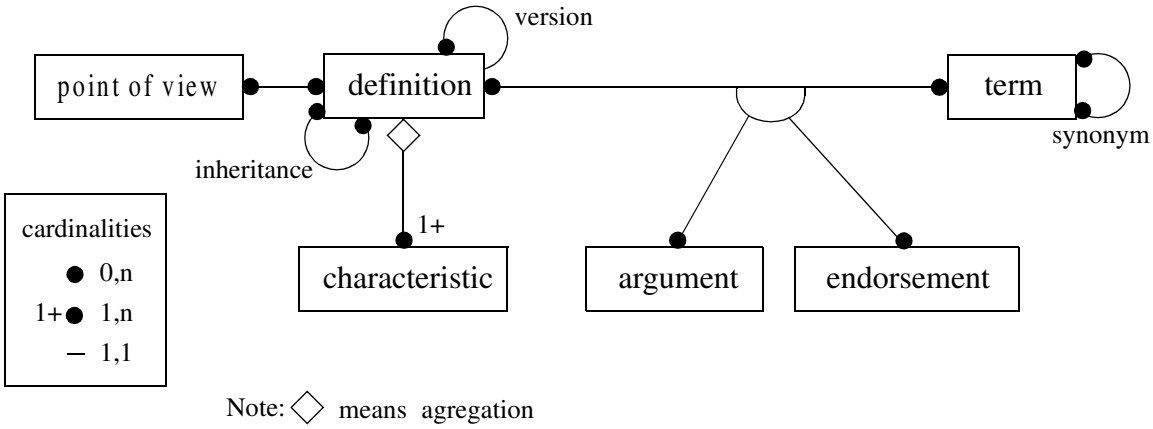
In order to simplify diagrams, arguments and endorsements can only rest directly on the link between a term and a definition. In reality, it could be considered as too restrictive, because a discussion is not only an addition of opinions, but it also has a flow. However this restriction has no effect on definition processing, which is our main interest. Endorsements can be regarded as a kind of link validation. They can also avoid to have too many repetitive arguments. If someone sees that there is already an argument which expresses his opinion, he can simply choose to add an endorsement to show his agreement (or disagreement).

A point of view represents a field's perception of single person or of a group of people. Within a point of view, definitions must be coherent. But points of views don't need to be coherent one with each other. So it is possible to have contradictory definitions for a single concept, provided that they belong to different points of view. For example, at the same time, the concept "dog" can be defined as a domestic animal by an Occidental whereas an Asiatic would tell it is an eatable animal.

Points of view then enable to solve definitions conflicts. As for terms, a possible conflict can be solved using a synonym link. That is if two differents terms are linked to the same definition there is a conflict only if they can't be used equally, i.e. if they are not synonyms.

A concept doesn't have a material existence; it is an abstraction, a mental representation of an real object. It cannot be handled directly. In our model, a concept is represented by a definition.

The ConceptIBIS model can be summed up as follows: (using a OMT-like notation)



There is no direct generalization/specialization relation between terms. This relation could however be deducted from the inheritance link of the underlying definitions, which depends on the point of view.

Characteristics could be decomposed in a set of sub-characteristics. This doesn't appear on the diagram, in order to make it more readable.

3 Concept Model

We use the concept model which was developed for the creation of multilingual concept bases in the ConcepTerm project.

A definition is a specialization of a more general definition; it owns a set of characteristics as well as constraints on these characteristics' domain. A domain constraint is itself a definition; it specifies which object categories are allowed for a given characteristic.

A concept definition is a statement which follows the following syntax:

ConceptDef ::=

definition DefinitionId **generic** DefinitionId **characteristics** CharactersiticConj

CharactersiticConj ::= Characteristic*

Characteristic ::= [**all**] CharacteristicName ":" DomainDef

DomainDef ::= [**not**] Term | Disjunction | Conjunction

Disjunction ::= "{" (Term | Characteristic | Disjunction | Conjunction)* "}"

Conjunction ::= "(" (Term | Characteristic | Disjunction | Conjunction)* ")"

Where * denotes 0, 1 or several occurrences of an element; [] denotes 0 or 1 occurrences and | denotes alternative.

For example (from the "Furniture" concept base of ConcepTerm project)

definition wardrobe
generic storage_furniture
characteristics
Dimension: big
Part: (type: door, number: {1; several})

Part: (type: shelf, number: several)
Part: (type: body, number: {1; several})
all Main_usage: (verb: store, objet: {linen; clothes})

Terms which appear in a definition indicate predefined concepts, i.e. concepts for which there is not explicit definition in the concept base (atomic concepts). The atomicity of a concept is not an absolute notion, it is relative to a field. For instance, *wood* can be regarded as atomic within the furniture field whereas it will be explicitly defined when talking about building materials.

In terms of expression power, this model is a subset of a terminological knowledge representation model such as ALCNR, i.e. every definition statement can be translated in a ALCNR statement. Contrary to ALCNR, it is not possible to express cardinality constraints on roles with ConcepTerm ($\leq$ n R, $\geq$ n R). So we chose to explicitely express cardinality with the "number" characteristic, as shown in the previous example.

A concept base is a set of concept definitions. Like in terminological logics, an interpretation I of a knowledge base (KB) is composed of

- a set D (the interpretation domain)
- for each elementary concept e, an interpretation $I(e) \subseteq D$
- for each characteristic R, a relation $I(R) \subseteq D \times D$

The interpretation of a concept defined as

concept C = generic G characteristics K_1 K_2, … K_n

is the subset I(C) of D, given by

$I(C) = I(G) \cap I(K_2) \cap ... \cap I(K_n)$.

The interpretation of a characteristic K = R: Dom is defined as

$I(R: Dom) = \{ o \in D | \exists p. p \in I(Dom) \text{ et } (o, p) \in I(R)\}$

That is, the set of the objects which have (at least) one characteristic R whose value is an instance of the concept Dom. A characteristic which is universally quantified is interpreted as

$I(\textbf{all } R: Dom) = \{ o \in D | \forall p. (o, p) \in I(R) \Rightarrow p \in I(Dom) \}$

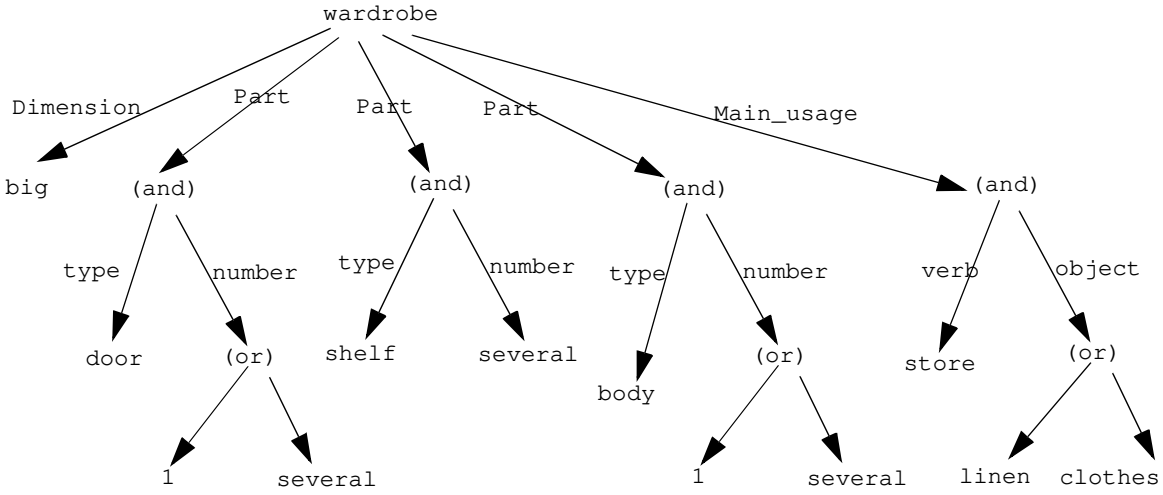
It is the set of the objects whose R characteristics are all instances of Dom.

The interpretation of a domain Dom (concept) is defined by the following rules:

- if Dom = (C_1, C_2, …, C_n), $I(Dom) = I(C_1) \cap I(C_2) \cap ... \cap I(C_n)$ (conjunction type domain),
- if Dom = I{C_1, C_2, …, C_n}, $I(Dom) = I(C_1) \cup I(C_2) \cup ... \cup I(C_n)$ (disjunction type domain),
- if Dom = e, I(Dom) = I(e) (atomic domain),
- if Dom = not e, $I(Dom) = D \setminus I(e)$.

Commutativity and associativity of union and intersection imply that the order in which the elements of a concept definition appear has no importance (interpretation remains unchanged under element permutation).

It is sometimes useful to view a concept definition as a tree with each arc representing a characteristic. The following figure shows the tree representation of the previous example (wardrobe).



4 Operations for collaborative work

4.1 Definitions comparison

Comparison is the basic operation to identify consensus and divergence, identify synonyms, etc. It is central in a process of collaborative building of concept bases. The comparison of two definitions is done by comparing their respective sets of characteristics. For this operation to be useful, it must indicate precisely the differences that exist between two definitions. A boolean comparison is not enough (A is equal to B or A is different from B); neither is a comparison that calculates a distance between two concepts and only gives a positive real number (whatever the sophistication of the calculation). One should also note that n-dimension distance is not applicable since characteristics may be multivaluated.

The following example explains our approach.

We consider two concepts C1 and C2, with their respective definition

C1 =
generic storage_furniture
characteristics
Dimension: big
Part: (type: door, number: {1; several}),
Part: (type: shelf, number: {1; several}),
Part: (type: body, number: {1; several})
Main_Usage:
(verb: store, object: {linen; clothes})

C2 =
generic storage_furniture
characteristics
Part: (type: door, number: {0; 1; several},
material: (type: pane)),
Part: (type: shelf, number: several))
Main_Usagel:
(verb: store, object: books)

One can express the difference between these two definitions by indicating what to add [+] and what to remove in definition C1 to get definition C2:

[+] Dimension: big
Part: (type: door, number: {**[+]** 0; 1; several}, **[+]** material: (type: pane)),
Part: (type: shelf, number: { **[–]** 1; several}),
[–] Part: (type: body, number: {1; several})
Main_Usage: (verb: store, object: {**[–]** linen; **[–]** clothes; **[+]** books})

If definitions are regarded as labeled trees, [+] and [-] labels indicate which subtrees to add or to remove in order to get C2 from C1. A difference between two definitions is a new subtree whose nodes and arcs are labeled with [+] or [-] or remain without label.

Definition. A modification M of definition A is a definition whose unlabeled elements coming either from conjunction or from a disjunction, or basic elements can be labeled with [+], [-], [+all], [-all].

Definition. The evaluation of a modification M consists in carrying out the following operations

- each element (characteristic r: C, single concept, conjunction, disjunction) with a [-] label is removed;
- each element all R: C with a [-all] label is replaced by R: C;
- each element R: C with a [+all] label is replaced by all R: C;
- each [+] label is cancelled.

Remarks

1) The result of an evaluation is a tree without any label ([+], [-], [+all], [-all]) left;

2) The characteristic label remain unchanged, with a possible addition or removal of the **all** prefix

Definition. A path from A to B is a modification M of A, such that the evaluation of M results in B. Notation: $\Delta(A, B)$ is the set of paths from A to B.

The elements of $\Delta(A, B)$ represent all the possible ways to get B from A. Whatever A and B, there is always at least one path from A to B. To see that, we can take A, remove all of its characteristics and then add all the characteristics from B to it. $\Delta(A, B)$ often has more than one element. For example, in addition to the path which was shown in the previous example, $\Delta(C1, C2)$ contains this other path:

Part: ([-] type: door, [+] type: shelf, number: {[-] 0, [-] 1, several}, [-] material: (type: pane))
Part: ([-] type: shelf, [+] type: door, number: { [+] 0, [-]1; several}), [+] material: (type: pane))
[-] Part: (type:body, number: {1; several})

It is evident that this last path is more complicated than the one from the example. To express the difference between two definitions A and B, we will only consider the paths from A to B with minimal complexity. For this purpose, we will define the complexity of a path. There are two main principles: 1) the more a path contains labeled elements ([+], [-], etc.), the more complex it is and 2) the higher the position of a label in the tree, the bigger the importance of the difference.

Definition. The complexity of a path M from A to B is a real number $\chi(M)$ depending on two parameters β and p. It is recursively defined as follows:

Let $M = K_1 \ldots K_n$

where $K_1, \ldots, K_n$ are labeled characteristics.

$\chi(M) = \chi(K_1) + \ldots + \chi(K_n)$

The complexity of a labeled characteristic K is defined as follows:

if K = [+all] R : E or if K = [–all] **all** R : E, $\chi(K) = \beta + (1/p)\chi(E)$ ($\beta > 1/p$),

if K = [+] R : E or if K = [–] R : E, $\chi(K) = 1$

if K = K : E, $\chi(K) = (1/p)\chi(C)$

The complexity of a labeled concept E is

if E = [+] or [–] C, $\chi(E) = 1$,

if $E = \{E_1, \ldots, E_k\}$ ou $(E_1, \ldots, E_k)$, $\chi(E) = (\chi(E_1) + \ldots + \chi(E_k))/k$,

if E = e (concept identifier), $\chi(E) = 0$

Some properties of the complexity measure:

- The p parameter sets the relative importance of a modification level compared with the directly lower level.
- The β parameter sets the relative importance of a [+/-/all] modification compared with other modifications of the same level.
- Every path or section of path which doesn't contain a [+] label has a null complexity.
- The (1/p) factors causes the weight of modification labels to decrease with the depth of their position in the tree.

Definition. The *distance* between A and B, marked d(A, B) is the complexity of the path(s) with minimal complexity from A to B.

Definition. A *difference* between A and B is a path with minimal complexity.

Some properties:

If d(A, B) = 0 then, for every interpretation I, we have I(A) = I(B). On the other hand I(A) = I(B) doesn't involve that d(A, B) = 0, that is there can be several ways to define the same concept. The distance evaluates a difference between definitions but not between interpretations. In other words, the distance is based on the syntax of a definition, not on its semantics. However, this syntactic approach seems more realistic. The computation of a semantics based distance would require either

- to know the interpretation (the extension) of each atomic concept, which is clearly not feasible for real world concepts or
- to use a deductive approach based on a very large set of domain axioms (such as: not small => {medium, large}, style:Renaissance => material:wood, etc.).

To minimize the problem of semantically equivalent definitions having a non-zero distance, one can put all the conjunctions and disjunctions of a definition in conjunctive normal form. This will not completely solve the issue but it can solve the most obvious cases.

Inheritance:

Distance and difference computation must be done on all the characteristics, including those that are inherited from generic concepts. But, if two definitions A and B are based on the same

generic concept, it is sufficient to use only their specific characteristics to calculate the distance. In a general case, it is not required to inherit all the characteristics to be able to carry out comparison; it is enough to go back up to the smallest common ancestor.

Computation:

The complexity of the distance and difference calculation is exponential, because in all cases of (and), (or) and multivaluated characteristics, one needs to try all possible permutations to find which one minimizes complexity. In all the cases that we met, the size of the permutations is limited (maximum 3 or 4 elements' permutations)

Discussion:

In all our examples, comparison is done on characteristics having the same names. It could also be possible to imagine a comparison that takes different names into account, but in this case calculations could become prohibitively complex. When collaboratively building a concept base, such a generality is not necessarily required, since people could agree on a list of characteristics' names that they will all use, thus providing coherence.

It could also be possible to create a list of equivalent terms to enable additional equalities to be detected. In the ConcepTerm project, this kind of method was used for translation: there was a list of equivalent values in different languages for the characteristics. However, this is only possible for a some specific domains, in which all the objects have relatively homogeneous attributes.

4.2 Manipulation operations: derivation

Once comparison has been carried out, one needs a few manipulation operations in order to make further steps towards consensus. Basically, manipulation operations should enable the modification of existing definitions. But as arguments and endorsements rest on links between terms and definitions, modifying a definition could have an unpredictable effect on them. To avoid this situation, we decided to forbid changes on existing definitions, so every operation must be done either on a new version of an existing definition or on a completely new definition (both are basically a copy of the original definition). In other words, one can say that all manipulations operations are grouped under the "derivation" label.

A **derivation** is a new definition which is created from an existing definition by either

- modifying the name and/or the value of one or more of its characteristics, or
- adding one or several new characteristics, or
- removing one or several characteristics.

A derivation can either be considerated as a new version of the original definition or as a completely new definition. (A new version of a definition still refers to the same concept, whereas a new definition corresponds to a new concept.)

Derivation can also be carried out from two (or more) definitions by:

- **conjunction**

  A conjunction of two definitions A and B is a new definition that possesses all their shared characteristics. This operation depends on the difference between A and B that is chosen.

  Let $D \in \Delta(A, B)$ and let D^{-} be the modification that is obtained by removing both the elements with a [+] label and the [-all] labels from D.

  Conjunction of A and B depending on D is the evaluation of D^{-}.

So conjunction corresponds to the removal of all the elements of A that are not in B. If a characteristic is **all** R:K on one side and R:K on the other side, **all** R:K is kept (this explains the removal of [-all] labels)

- **disjunction**

  A disjunction of two definitions A and B is a new definition that possesses all the characteristics of both. This operation depends on the difference between A and B that is chosen.

  Let $D \in \Delta(A, B)$ and let D^+ be the modification that obtained by removing the [+all] and [-] labels from D. ([+] and [-all] labels are kept)

  Disjunction of A and B depending on D is the evaluation of D^+.

  So disjunction corresponds to the addition to A of all the elements of A that are only in B. If a characteristic is **all** R:K on one side and R:K on the other side, R:K is kept (this explains the removal of [+all] labels)

The operations mentioned above can of course be combinated.

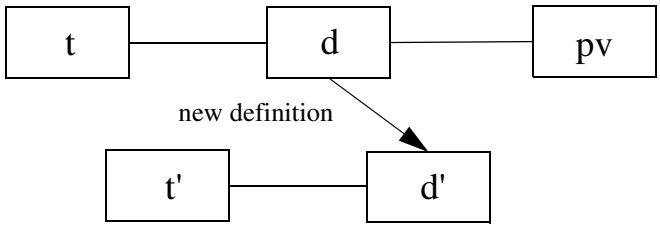
4.2.1 Examples

The following examples illustrate the use of these operations in typical cases.

In the next few diagrams, *t* is a term, *d* is a definition, *pv* is a point of view, *a* is an argument and *e* is an endorsement.

Creation of a generic definition

The creation of a generic definition consist in copying a definition and removing some of its characteristics (characteristics that are left aside must be those that made the specificity of the original concept).

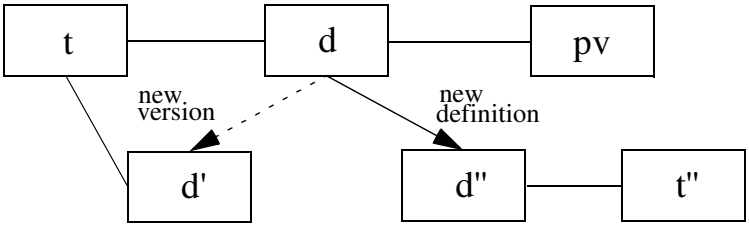


Creation of a specialied definition

Specialization is the opposite to generalization: a new definition is created, but new characteristics are added to the original definition (instead of removed). In both cases (generalization and specialization), a new term (t') will perhaps be necessary to designate the new generic concept (d').

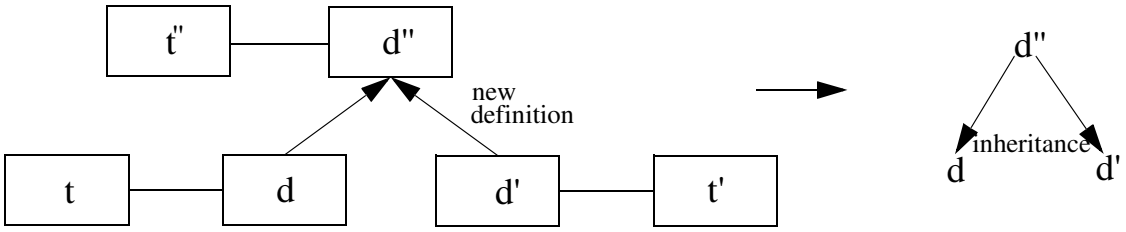
Spliting a definition that represents two concepts

When it appears that a definition doesn't correspond to a single concept (i.e. two concepts have been mixed together), it is necessary to separate them. In this case, one could create a new version (d') in which there would only be the characteristics that correspond to the term (t) and create a new definition (d'') with the other characteristics.



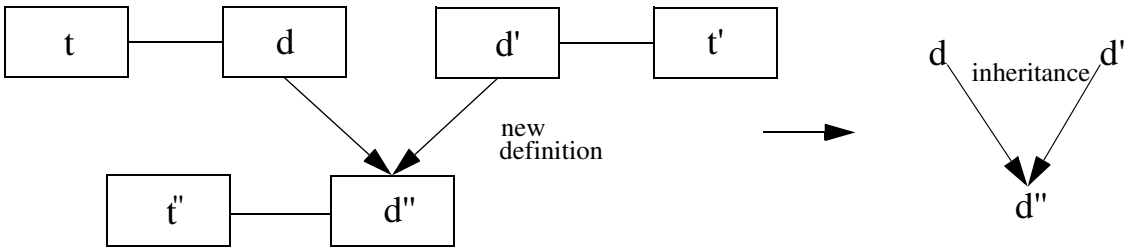
Creation of a shared generic definition

The creation of a shared generic definition for two or more definitions can be done using conjunction. The new definition possesses all the shared characteristics of the basic definitions. A new term (t'') will probably be needed to name the new concept (d'').



Creation of a shared specialized definition

Disjunction can be used to create a shared specific for two or more definitions, that is a concept (d'') that inherits all the characteristics of the basic definitions.



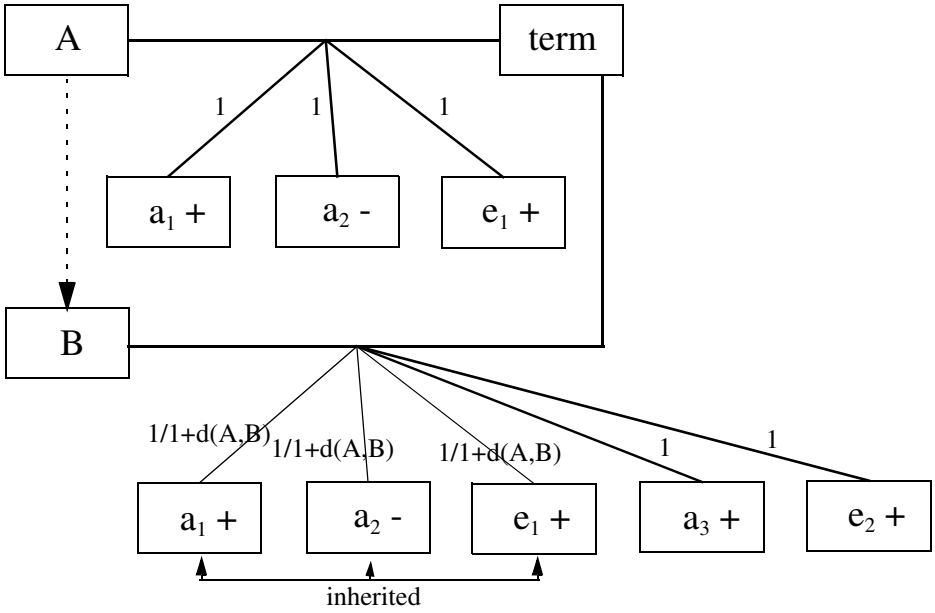
4.3 Arguments and endorsements inheritance

When one of the above-described operations results in the creation of a new definition version, one cannot assume that the arguments and endorsements which are attached to the original link are still valid for the new one.

The greater d(A, B), which represents the distance between the old and the new definition, the less likely arguments and endorsements will be valid. So we chose to weight arguments and endorsement to enable distinction between those which are inherited and those that are specific to a link.

Weighting is:

- 1 for specific arguments and endorsements,
- 1/1+d(A, B) for inherited arguments and endorsements.



Remark: for the following versions (i.e. from the third version), it is always the A from the basic definition that is taken into account for the weighting calculation. Thus the weight decreases progressively.

5 Use of the ConceptIBIS model

5.1 The collective creation process

The ConceptIBIS model is intended to be used as described below for the collaborative building of a terminological concept base.

1) Terms and definition creation, without constraints. In principle, each link between a term and a definition should have at least one argument to enable a constructive discussion.
2) Deliberations: participant can show their agreement or disagreement on the links by creating other arguments or endorsements.
3) Analysis: comparison of the definitions to identify consensus and divergence.
4) Resolution: attempt to reach consensus where there are conflicts, by using the different operations defined previously. Arguments and endorsements can give indications that help making decision. This phase should be carried out under supervision of an administrator. (see 5.2)
5) Removal of the useless or incorrect definitions.

If there are many different definitions for each term, it may not be possible to go forward linearly. One must iterate on the deliberation, analysis and resolution phases (2, 3, 4). Each time a manipulation operation is applied, a new version or a new definition is created. This one must be compared with the remaining ones. Then comparison can lead again to a manipulation oper-

ation, and so on. Moreover, we have to remember that a concept base must be lively to keep in touch with reality. The entire process is then also iterative.

5.2 Role of the resolution administrator

Typically, the administrator could be a terminologist. Terminologists are a kind of knowledge engineers. They can act as more neutral referees than experts because they are not involved in the field. (Terminologists often have a linguistic background) Terminological work usually begins with going through a field's reference texts in order to identify the specific terms. After that, terms are organized in a hierarchy called "domain tree". But this domain tree can also be built progressively. Both kind of situations can be supported with ConceptIBIS: terms and definitions can be created independently and sorted to a hierarchy afterwards; or, terms can be inserted in the domain tree immediately after their creation. It is up to the administrator to decide when to go into the next phase of the process (or when to begin a new iteration).

Storing the discussion (as arguments) together with terminological data not only allows to remember how definitive definition were chosen but it also gives additional information to users. It thus justify the fact that definitions can't be removed (even non consensual definitions are not removed as they represent a part of the discussion). Definition removal should however be allowed to the administrator of the concept base so that he could correct important errors that could disrupt operations. Arguments and endorsements are particularly useful during the concept base building phase. They help users to make a choice between the differents propositions especially when none of them has been validated yet. They also help the administrator to carry out validation. In a traditional terminological concept base, it is not always compulsory to achieve consensus on all definitions. But if the ConceptIBIS model is used to define objects for an information system, it would be really desirable to have, at the end of the process, only one definition for each term, that is even if the other definitions are not removed, the one that is chosen must have a distinction mark (for example a special endorsement by the administrator).

5.3 Points of view and conflicts resolution

Points of view are meant to enable definition conflicts resolution. They can also be used to put together definitions for which a type of users may have interest in.

The concept base is in a *coherent* state when, for each term, we have at most one current definition per point of view. (including the default point of view).

Remark: a current definition is the last version of a definition.

When a "two different definitions for a single term" type conflict occurs, there are three possibilities to solve it:

- Consensus: only one definition is kept. For that purpose, one can either remove one of them or create a new one that combines the two basic definitions.
- Contrast: one accepts that the two definitions correspond to two different concepts (for example when the definition are contradictory). One has then to create a new term for one of the definitions. So, one comes to the "two different terms for two different concepts" situation.
- Different points of view: the two definitions are kept but each one is linked to a different point of view.

In the case of "two different terms for a single definition", one can choose one of the following solutions:

- Correspondence: one admits that the two terms can be used equally. A synonym link is then created between them.
- Consensus: only one of the two terms is kept, the other is removed.
- Contrast: the two terms have different meanings, thus a new definition is created for one of them.

Remark: We use the terms: “conflict”, “consensus”, “correspondence” and “contrast” in the same way as Shaw and Gaines [Sha89] (see table in section 1.2)

For the concept base to be in a coherent state all the conflicts must have been solved using one of the solutions described above, that is the resolution phase must be finished.

6 Conclusion

In this paper, we presented a multi point of view model that is a good base for the creation of a tool aimed at the collective building of a concept base. It enables to resolution of term-definition conflicts by integrating a collaborative part from IBIS in a knowledge model. The most important is that ConceptIBIS allows conflict resolution without imposing to give priority to only one point of view. ConceptIBIS can be seen as a concept base with advanced newsgroups functions.

We are currently testing our comparison function, as well as other operations on the “Furniture” concept base[1] developed with the ConcepTerm project. Simultaneously, we are beginning the implementation of a system based on our ConceptIBIS model. This system will be used by translators, terminologists and field specialists for creating a terminological concept base while exchanging terminological information.

7 Acknowledgements

The authors would like to thank the anonymous referees for their insightful and stimulating comments on the first version of this paper.

1. The “Furniture” concept base contains about 1000 definitions in the furniture field

8 References

[Ber94] Berthet Dominique, Bonjour Michel, De Bessé Bruno, Falquet Gilles, Léonard Michel, Sindayamaze Jeanne, *ConcepTerm "Construction de dictionnaires encyclopédiques multilingues et informatisés"*, CUI technical report, University of Geneva, 1994

[Bra91] Brachman, R., McGuinness, D., Patel-Schneider, P., Borgida, A. and Resnick, L. *Living with CLASSIC: When and How to Use a KL-ONE-Like Language*, in Principles of Semantic Networks. Morgan Kaufman. Pp. 401-456. May, 1991.

[Buc97] Buckingham Shum Simon, *Representing Hard-to-Formalise, Contextualised, Multidisciplinary, Organisational Knowledge*, in AIKM'97 Proceedings

[Buc93] Buchheit Martin, Donini Francesco M., Schaerf Andrea, *Decidable Reasoning in Terminological Knowledge Representation Systems*, Journal of Artificial Intelligence Research, 1993

[Con89] Conklin Jeffrey, Begeman Michael, *gIBIS: A Tool for All Reasons*, Journal of the American Society for Information Science, 40, pp 200-213, 1989

[Con96] Conklin Jeffrey, *Designing Organizational Memory: Preserving Intellectual Assets in a Knowledge Economy*, Group Decision Support Systems, 1996

[Deb89] De Bessé Bruno, Pulitano Donatella, *BD-TERM, un logiciel de gestion terminologique*, ETI, Université de Genève,1989

[Die97] Dieng Rose, *Comparison of Conceptual Graphs for Modelling Knowledge of Multiple Experts: Application to Traffic Accident Analysis*, Rapport de Recherche n°3161, INRIA Sophia Antipolis, Projet Acacia, 1997

[Euz96] Euzenat Jérôme, *Corporate memory through cooperative creation of knowledge bases and hyper-documents*, in Proceedings of the Tenth Knowledge Acquisition for Knowledge-Based Systems Workshop (KAW '96)

[Fal91] Falquet Gilles, Sindayamaze Jeanne, Bonjour Michel, Leonard Michel, *F2Concept, un modèle intégrant la description de la compréhension et l'extension des classes d'objets*, CUI, Université de Genève, 1991

[Gro] Group Decision Support Systems, *The IBIS Manual: A Short Course in IBIS Methodology*

[Kun72] Kuntz Werner, Rittel Horst, *Issues as elements of information systems*, Working Paper No 131, Institute of Urban and Regional Development, University of California at Berkeley, 1972

[Mar93] Mariño Drew Olga, *Raisonnement classificatoire dans une représentation à objets multi-points de vue*, thèse de doctorat, Université Joseph Fourier, Grenoble, 1993

[Pul88] Pulitano Donatella, *Création d'une banque de terminologie à l'Ecole de traduction et d'interprétation de l'Université de Genève : BD-TERM*, ETI, Université de Genève, 1988

[Sha89] Shaw Mildred L. G., Gaines Brian R., *Comparing Conceptual Structures; Consensus, Conflict, Correspondance and Contrast*, Knowledge Science Institute, University of Calgary, 1989

[Sin95] Sindayamaze Jeanne, *Prise en compte de la compréhension dans les bases de données*, thèse de doctorat, Université de Genève

[Stu98] Stumpf Simone, *Argumentation-based Design Rationale - The Sharpest Tools in the box*, Working Paper IN/98/01, University College London