

A Viewpoint Model for Cooperative Building of an Ontology

Myriam Ribière¹ and Rose Dieng-Kuntz²

¹Centre de Recherche Motorola, Espace Technologique
Saint Aubin, 91193 Gif-sur-Yvette Cedex, France

ribiere@crm.mot.com

²INRIA Sophia-Antipolis

2004 route des Lucioles, BP 93, 06902 Sophia-Antipolis Cedex, France

Fax: (33) – 4 92 38 77 83

Rose.Dieng@sophia.inria.fr

Abstract. This paper presents the C-VISTA model enabling viewpoint representation for improving knowledge modelling from several experts: this model can thus be used for cooperative building of an ontology. We describe the C-VISTA model in the framework of conceptual graph formalism.

1 Introduction

Several methods, techniques and protocols were proposed for knowledge acquisition (KA) from multiple experts. They aim at: (a) expressing common parts and differences between experts' models; (b) detecting and solving terminology conflicts between the experts; (c) taking into account the different viewpoints of different experts: several experts according to their specialty or their way to tackle the problem solving, may have divergent analyses or divergent understandings of a same object. So, the notions of multi-expertise and multi-viewpoints are closely related.

An expert can have different viewpoints or perspectives on a domain or on a problem; so, for each expert, the knowledge engineer (KE) can identify several viewpoints (see fig. 1).

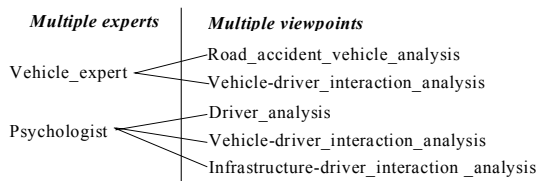


Fig. 1. Example of link multi-expertise/multi-viewpoints in an application of road accident analysis

The viewpoints of each expert are actually the different points of interest of this expert concerning the application. This relation establishes a partition of the knowledge base (KB) where each viewpoint is a coherent and partial view of the KB. But as in the example of fig. 1, some viewpoints may be shared. This leads us to consider the relationship between multi-expertise and multi-viewpoints otherwise. Indeed, on a same identified viewpoint, there may be different “viewpoints” of different experts from the same domain or from different domains. In that case, the KE must either integrate knowledge from the different experts or make those different “viewpoints” on a viewpoint live together.

So even if the link between multi-expertise and management of multiple viewpoints seems to be intuitive, it is not really well defined for different reasons:

- "Viewpoint" is a common sense word and may have different interpretations [2] [11] [1] [12].
- The use of viewpoints in the building of a KB varies according to the different fields considered (knowledge representation (KR), KA, requirements engineering, software design, cooperative design, explanations...) and their definitions of viewpoint.

Our objective is to model the notion of viewpoint and to offer a way of expressing and managing viewpoints. The description of our model C-VISTA is based on conceptual graph (CG) formalism [14] but this model could be relevant for other KR formalisms. First we present our definition of viewpoint, and our model C-VISTA for handling viewpoints. Then we present a method based on this model for cooperative building of an ontology, before a comparison with related work in our conclusion.

2 Viewpoint Definition

Contrary to most work proposed in object-oriented representation, we do not necessarily aim at obtaining consensual and complementary views from viewpoints. Viewpoints may index consensual and non-consensual knowledge:

- They may help in knowledge acquisition process by providing a support to represent non-consensual knowledge from several experts (i.e. express the “viewpoint” of each expert on the same object).
- They enable to keep non-consensual viewpoints on a same object.

According to [11], a viewpoint is "*a perspective of interest from which an expert examines the knowledge base*". Our definition of viewpoint is based on this definition, but we emphasize two dimensions - a contextual dimension where the focus of an expert is described, and a personal dimension where the view angle of the expert is described:

- The *focus* describes the expert's work context (task and objective). Several experts can have the same focus: for example, in fig. 2, the focus is the *security* of a vehicle. According to this focus, we need to express the different viewpoints of different experts involved. Therefore we characterize the viewpoint by a personal dimension: the view angle.

- The *view angle* describes the characteristics of an expert or of a group of experts. It can describe the name of this expert (or of this group of experts), his/her application field (domain), his/her expertise level or skill, his/her experience in other domains interesting for the application, his/her role and place in his/her organization.

Our final definition is the following: "A viewpoint is an interface allowing the indexation and the interpretation of a view composed of knowledge elements. A viewpoint is characterized by a focus and a view angle" (cf. fig. 2).

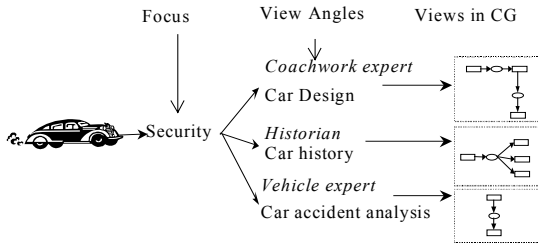


Fig. 2. Example of multi-viewpoint description of a car

In terms of KB building, a viewpoint allows to index knowledge in order to make it accessible, dynamic and reusable. In terms of access to the KB, it plays the role of a filter on the KB and helps the user to avoid to get lost in the whole KB by enabling access only to relevant knowledge according to the user profile. We distinguish two kinds of viewpoints:

- Viewpoints defining *perspectives* that index consensual descriptions of a same object by different experts. Those views are complementary and give a whole vision of the object. The object is supposed to be unique, but may have some characteristics interesting or visible only for some experts. Therefore a given expert will focus only on some perspectives on the object (the ones relevant for him). The models proposed in ROME [2], TROEPS (previously called TROPES) [11], VBOOL [10] for management of multiple viewpoints or in View Retriever [1] for extraction of viewpoints from a frame-based KB rely on the hypothesis that viewpoints are partial representations of a unique, coherent set of objects. So, these models handle perspectives.
- Viewpoints defining *opinions* that index non-consensual descriptions corresponding to the different, specific approaches of the experts. Such views are incomplete descriptions of the studied object and could be collectively inconsistent. For example, such opinions are useful in case of design of an artefact on which several experts will have different (possibly contradictory) propositions.

We call the first type of viewpoints "*perspective viewpoints*" and the second type "*opinion viewpoints*".

3 Model for the Cooperative Building of an Ontology: C-VISTA

Our objectives, according to our viewpoint definition, are the following: (1) Identify and index terminological differences between experts and establish a link between different terminologies; (2) Enable multi-representation of an object according to different experts.

3.1 Problems in Ontology Cooperative Building

Our past experiments of knowledge engineering showed that experts working together with the supervision of a KE tend to integrate their vocabularies and to create a common vocabulary in order to understand one another. But the concepts underlying such terms may be actually used differently by the different experts at different levels of granularity and for different situations, so with divergent interpretations. Besides, the KE does not model the context and the objective of use of each concept or term. This notion of context and objective on a concept must be normally deduced from the concept type hierarchy organization (*kind_of* link). But when in a huge ontology, the whole context of interpretation is not modeled explicitly, it leads to misunderstandings (in particular when a part of the ontology must be reused for another application).

For example, in a hierarchy using only the "*kind-of* link " to provide the interpretation of a concept type, the *Accident_Factor* concept type could be subtyped by the *Lack_of_road_exit_specific_signal* concept type (that is a kind of accident factor) and by the *Indicator_ambiguity* concept (that is also a kind of accident factor). Such accident factors are proposed by different experts. The first concept type is based on the analysis of the infrastructure as accident factor and conforms to the *Infrastructure viewpoint* while the second one is based on the analysis of the driver as accident factor and conforms to the *Driver viewpoint*.

So, a viewpoint enables the explicit expression of a particular subtype relation existing between two concept types. Most of the methods described in related work build a consensual ontology, without such particular subtype links that provide documentation and track of the knowledge integration process. As a concept type hierarchy can often be built using several different criteria, our notion of viewpoint enables to make explicit the criteria underlying the subdivision of a concept type into its subtypes.

3.2 The C-VISTA Model

The C-VISTA model is defined using the CG formalism [14] [3]. In this formalism, we can build a support S and a base of conceptual graphs. A support S is composed of a concept type hierarchy (noted T_c), an ordered set of relation types with their signatures, a set of markers or referents (M) and a conformity relation between concept types and markers. A CG is a bipartite graph built according to the support S and composed of two types of nodes: 1) concept-nodes, each labelled by a concept type and a referent, 2) relation-nodes, each labelled by a relation type. The support S corresponds to terminological knowledge (or ontology) while the base of CG corresponds to assertional knowledge.

In this section, we first present how to express viewpoints in the concept type hierarchy, then we show how to allow the multi-representation of an object and how to link different terminologies in a same concept type hierarchy, thanks to the organization of the ontology in several viewpoints.

Expression of Multiple Viewpoints

Definition 1: Basic and v-oriented concept types

Let t_c and $t_{c'}$ be two concept types. If $t_{c'}$ is a subtype of t_c , then there may exist a viewpoint p such that $t_{c'}$ is a subtype of t_c according to the viewpoint p . In that case, t_c is called “*basic concept type*” and $t_{c'}$ “*viewpoint-oriented concept type*” (noted “*v-oriented concept type*”).

For example, the v-oriented concept type *Highway* is a subtype of the basic concept type *Infrastructure* according to the *Administrative* viewpoint.

A given concept type may have several immediate supertypes: a concept type t' may be both a subtype of t_1 w.r.t. the viewpoint p_1 and a subtype of t_2 according to the viewpoint p_2 . So, the concept type hierarchy corresponds to a partial order but not to a rooted tree.

A given concept type can be both basic (i.e. have v-oriented subtypes) and v-oriented (i.e. be itself a subtype according to a viewpoint). So the presence of basic concept types is not necessarily restricted to the higher levels of the concept type hierarchy.

Definition 2: Criterion

A viewpoint is characterized by the explicit *criteria* according to which a v-oriented type will be considered as a subtype of its basic type. A criterion is called *generic* if it is not instantiated (e.g. “*Context*” is a generic criterion). A criterion is called *specific* if it is instantiated (e.g. “*Context: Accident_analysis*”).

Definition 3: Viewpoint template

A *viewpoint template* is composed of two sets of generic criteria, the first set characterizing the focus and the second one the view angle.

We will suppose that a given ontology relies on one viewpoint template: before developing the ontology, the KE and the experts must agree on a viewpoint template on which they will rely for organizing the ontology. The viewpoint template proposed in fig. 3 is composed of:

- The *focus*, characterized by the generic criteria *Context* and *Objective*,
- The *view angle*, characterized by the generic criteria *Person*, *Field of skill*, *Expertise level*, *Other expertise fields* and associated levels of expertise.

Definition 4: Viewpoint

Let P_{template} a viewpoint template made of n generic criteria. A viewpoint p is defined on this template P_{template} , by instantiating m generic criteria of this template ($1 \leq m \leq n$). So p is defined by a set of m specific criteria (C_i having a value v_i). We note $\text{Criteria}(p) = \{(C_i, v_i) ; i=1..m\}$.

	Example of viewpoint template
Focus	Context Objective
View angle	Person Field of skill Expertise level Other expertise field/level of expertise

Fig. 3. Example of viewpoint template

Fig. 4 gives an example of a particular viewpoint built according to the viewpoint template shown in fig. 3.

	Example of viewpoint
Focus	Context: Accident analysis Objective: Security of the crossroad
View angle	Person: Manuel Field of skill: Infrastructure Expertise level: 9 (Expert) Other expertise field / level of expertise: Vehicle dynamics / 5

Fig. 4. Example of viewpoint

Definition 5: Viewpoint link

Let t_c and $t_{c'}$ be two concept types. Let p be a viewpoint defined by $Criteria(p) = \{(C_i, v_i) ; i=1..m \}$. If $t_{c'} < t_c$ then there may exist a link (VPT: p) such that $t_{c'}$ is a subtype of t_c according to this viewpoint p .

Definition 6: Inclusion of two viewpoints

Let p_1 and p_2 be two viewpoints respectively defined by $Criteria(p_1) = \{(C_{1i}, v_{1i}) ; i=1..m \}$ and $Criteria(p_2) = \{(C_{2j}, v_{2j}) ; j=1..k \}$. p_1 is included in p_2 iff $Criteria(p_2) \subset Criteria(p_1)$.

For instance, the viewpoint in Figure 4 is included in the viewpoint defined by $\{(Context, Accident analysis) (Person, Manuel) (Field of skill, Infrastructure)\}$.

C-VISTA model, summarized in fig. 5, allows to express for one given concept type, a set of more specialized concept types provided by the experts. For the process of terminology integration, we need to express different kinds of viewpoint links in order to distinguish consensual and non-consensual knowledge in the ontology.

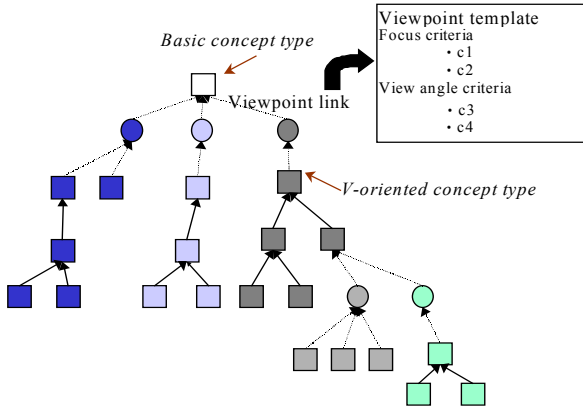


Fig. 5. C-VISTA Model

Therefore we introduce the following links:

- The *perspective link* noted (PERSP: p) can index, according to the viewpoint p, a concept type having a consensual definition, i.e. shared by all the experts. The subtypes of such a concept type are considered as having also consensual definitions.
- The *opinion link* noted (OPINION: p) can index, according to the viewpoint p, a concept type having a non-consensual definition. This concept type stems from the opinion of an expert, not yet shared by the other experts. The subtypes of a such concept type are considered as also having non-consensual definitions.

Fig. 6 shows an example of application of C-VISTA model. It describes three viewpoint links and shows the different v-oriented subtypes of *Accident_factor* according to those three viewpoints. Each of those viewpoints is characterized by a particular instantiation of the same viewpoint template. So, the C-VISTA viewpoint model enables to organize the concept type hierarchy into explicit viewpoints, making the resulting ontology more accurate and readable.

Multi-representation of a Concept

Having multiple representations of an object allows, for a given instance, to obtain different perspectives describing this instance according to different viewpoints. For example, an *infrastructure* could be seen as a *straight_road* or a *curved_road* according to the *curve viewpoint*; it could also be seen as a *highway*, a *national_road* or a *departemental_road* according to the *administrative viewpoint*; and last, according to the *nb_lanes viewpoint*, it could be a *3_lane_road* or a *2_lane_road* or a *One_way*. The different possible viewpoints for an instance are represented in the concept type hierarchy with the C-VISTA model.

In CG formalism, the creation of an instance establishes a link between the original concept type of the hierarchy and the name of the instance. This link is called *instantiation link* (it corresponds to the *is_a* link of object-oriented representations). The instantiation of a basic concept type is called a *basic concept* and the instantiation

of a v-oriented concept type is called a *v-oriented concept*. We also introduce another link, called *representation link* (see model in fig. 7 and example in fig. 8), and inspired by ROME [2].

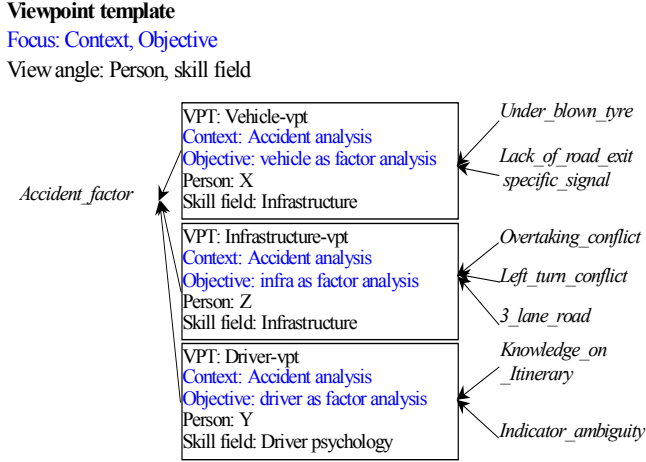


Fig. 6. Example using C-VISTA model

Definition 7: Representation link

Let C, noted [t_c:ref], a concept characterized by a basic concept type t_c and by a referent ref. Let T=∪_{i=1..n}{t_{ci}} the set of v-oriented concept types subtypes of t_c. If C has as representations m v-oriented concepts noted [t_{cj}:ref] (with 1≤j≤m≤n and t_{cj} ∈ T), then each [t_{cj}:ref] has a representation link with C.

As such v-oriented concepts are different perspectives on a same object, they are defined from v-oriented concept types, which are indexed by *perspective viewpoints*.

Definition 8: Coreference set

Let C, noted [t_c:ref], a concept characterized by a basic concept type t_c and by a referent ref. If E_{rep}(C) = ∪_{j=1..m}{C_j} is the set of the concepts having a representation link with C, then the coreference set of ref is:

$$Coref(ref) = \{t_c\} \cup \{type(C_j) \text{ such that } C_j \in E_{rep}(C)\}.$$

Definition 9: Extension of conformity relation

The conformity relation, conf, defines the set of authorized markers, for each concept type. This relation is defined on T_c x (M ∪ {*,0}) and satisfies:

$$\forall m \in M, \forall t, t' \in T_c :$$

- conf(T, m) and ¬ conf(⊥, m) where T is the universal type and ⊥ the absurd type,
- if t' ≤ t and conf(t', m), then conf(t, m)
- if conf(t,m) and conf(t', m) then conf(t ∧ t', m) and t ∧ t' > ⊥
- if t is a basic concept type and if conf(t,m), then ∃ t_i ∈ Coref(m), conf(t_i,m).

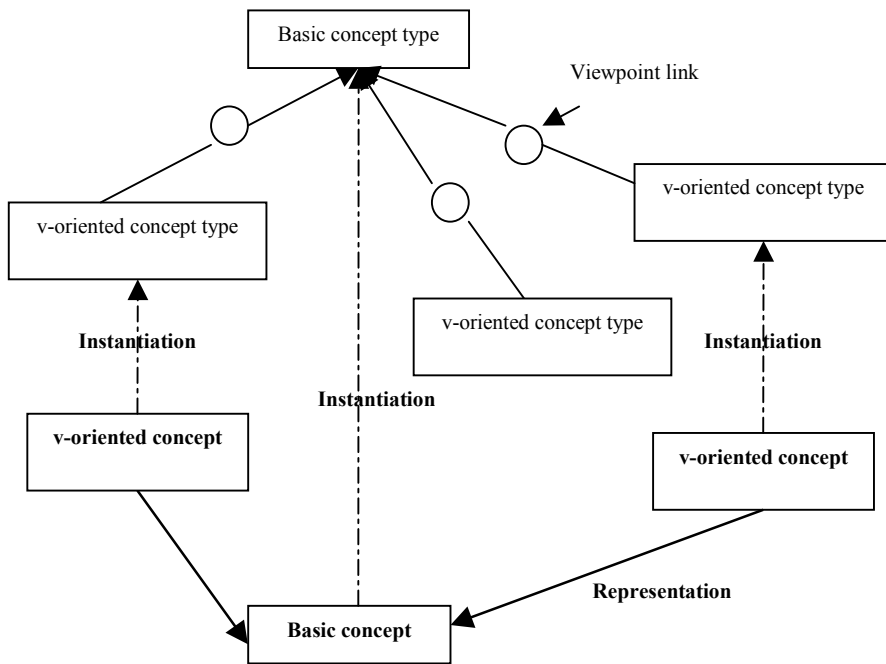


Fig. 7. Model of multi-viewpoint representation

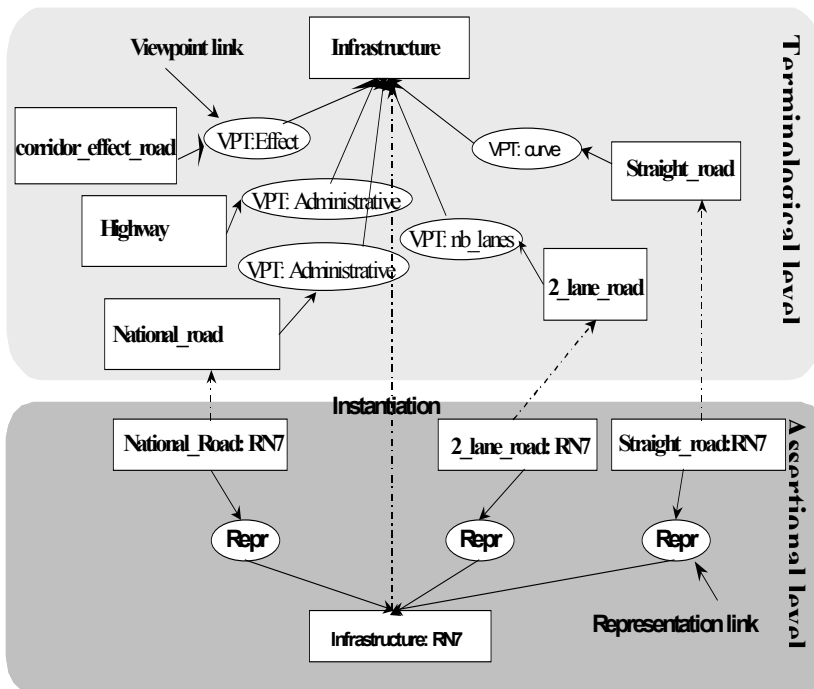


Fig. 8. Example of multi-representation with C-VISTA

For example, in figure 8, the marker RN7 conforms to the basic concept type *Infrastructure* and to the v-oriented concept types *National_Road*, *2_lane_road* and *Straight_road* – that must have a maximum common subtype.

This model has two advantages:

- It allows to gather information about an object in a same model. So it takes advantage of object-oriented formalism but keeps the advantage of a relational-based formalism.
- It is a dynamic structure allowing to modify an object by adding or removing a perspective on it without having to destroy and recreate the object.

Links between Terminologies

If we try to integrate terminologies in the same concept type hierarchy, it is important to interconnect v-oriented concept types. In the example of fig. 6, we notice that some subtypes of *Accident_factor* are equivalent such as *Lack_of_road_exit_specific_signal* and *Indicator_ambiguity*. They are defined in different viewpoints (having the same focus but different view angles) but if they are used in an assertion about the accident, they could be used to mean the same thing.

Therefore, we defined three types of links to handle the different possible relations existing between terminologies.

Definition 7: Equivalence link

Let t_c be a basic concept type. Let t_{c1} and t_{c2} be two v-oriented concept types, subtypes of t_c according to two distinct viewpoints p_1 and p_2 . If there exists an equivalence link, noted *Equiv*, between t_{c1} and t_{c2} , it means that, if $C_1=[t_{c1}:ref]$ is a representation of $C=[t_c:ref]$, then $C_2=[t_{c2}:ref]$ is also a representation of C and vice-versa.

This equivalence link between two v-oriented concept types stemming from two different viewpoints enables to identify two concept types having the same meaning but used in different contexts (and perhaps named differently) by two experts. It corresponds to the bridge among classes of different perspectives, offered by TROEPS [11].

Definition 8: Inclusion link

Let t_c be a basic concept type. Let t_{c1} and t_{c2} be two v-oriented concept types, subtypes of t_c according to two distinct viewpoints p_1 and p_2 . If there exists an inclusion link, noted *Incl*, between t_{c1} and t_{c2} (we say t_{c1} includes t_{c2}), it means that, if $C_1=[t_{c1}:ref]$ is a representation of $C=[t_c:ref]$, then $C_2=[t_{c2}:ref]$ is a representation of C .

This inclusion link enables to express that the meaning of the first concept type implies that of the second one. It can be useful if two experts express their concept types with different grain levels. For example, in fig. 9, *Overtaking_conflict* and *Left_turn_conflict* are types of accident factors identified by the expert in infrastructure. But in fact, a discussion among the experts reveals that their definitions are included in the concept type defined by the psychologist and called *Indicator_ambiguity*. So, the KE can add inclusion links between *Overtaking_conflict* and *Indicator_ambiguity* on the one hand and between *Left_turn_conflict* and *Indicator_ambiguity* on the other one.

Definition 9: Exclusion link

Let t_{c1} and t_{c2} be two v-oriented concept types, subtypes of t_c according to the same viewpoint p . If there exists an exclusion link, noted *Excl*, between t_{c1} and t_{c2} , it means that, if $C_1=[t_{c1}.ref]$ is a representation of $C=[t_c.ref]$, then $C_2=[t_{c2}.ref]$ cannot exist (and vice-versa). It means that t_{c1} and t_{c2} cannot both belong to the coreference set of the same referent.

This exclusion link enables to identify the concept types that cannot be at the same time representations of the same instance. For example, the concept types *Straight_road* and *Curved_road* stemming from the *Curve* viewpoint cannot be both used for representations of a same infrastructure.

In C-VISTA model, we propose those three links that seem the most useful for our intended applications, but other links between v-oriented concept types could be defined: e.g. the composition of viewpoints proposed in [1].

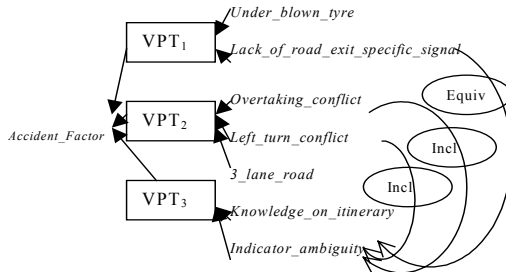


Fig. 9. Example of links

3.3 User’s Viewpoint for Accessing the Ontology

A user’s viewpoint can be defined using the same viewpoint template as the one used for the ontology building. Thanks to the viewpoint representation, the set of the ontology viewpoints which are included in this user’s viewpoint (cf. definition 6) can be determined automatically, so as to restrict to them when visualizing the ontology for this user.

4 Method for Ontology Cooperative Building with C-VISTA

The KE can exploit C-VISTA model as follows:

1. Elicit knowledge from the experts or analyse their documents (using a knowledge engineering method or possibly a KA method from texts).
2. For each expert, determine the concept types used by this expert.
3. Agree on the viewpoint template to be used for the ontology.
4. Establish the “common hat” of ontology, composed of the *common concept types* (i.e. denoted by the same terms by all the experts and having the same definition for all of the experts).

5. For each expert, index by *opinion viewpoints* the concept types that are both specific to this expert and subtypes of a common concept type. Such concept types are *v-oriented*. Make explicit the focus and view angle of these opinion viewpoints, as well as the initial set of definitions, without yet working on their interpretation: some definitions may be redundant (resp. conflictual).
6. Work with the experts upon this first representation (by analysing the so far obtained concept type hierarchy), in order to confront the concept type definitions so as to detect the consensual ones.
7. Index the set of consensual definitions by *perspective viewpoints* according to the following rules:
 - In the *view angle* of the perspective viewpoint, indicate the experts from which the considered concept type stems.
 - Name differently the concept types denoted by the same term by the different experts, if there is no agreement on a common definition: e.g. concatenate the initial name and a feature of the expert.
 - If two concept types from two different viewpoints have similar definitions, they can be linked by an *equivalence link*.
 - If two concept types from two different viewpoints have compatible definitions, one including the other, an *inclusion link* can be set between them.
 - If two concept types describe different properties of a basic concept type and if, according to the experts, these properties cannot occur simultaneously on a same instance, an *exclusion link* can be set between both concept types.
8. Compare the definitions of the concept types indexed according to *opinion viewpoints* having the same focus. Index the definitions that can be integrated, by *perspective viewpoints* as described in step 6. Track of this integration relies on opinion viewpoints keeping the intermediate definitions elaborated during the construction of the ontology.

This method, obtained by abstraction of our own experiments, offers a methodological guide for helping the humans involved (i.e. the knowledge engineer and the experts) to create the ontology collaboratively but it is not yet supported by a tool.

5 Conclusions

5.1 Applications

C-VISTA model was completely implemented in C++ above the conceptual graph platform COGITO [9] that we extended by second-order concept and relation types, as required by C-VISTA: we implemented an environment for multiple viewpoint management, and in particular, methods enabling to create a viewpoint template, create the corresponding generic viewpoint, create a viewpoint from the list of specific criteria, add a viewpoint to a list of viewpoints managed in the environment,

establish a viewpoint between two concept types, establish a bridge (i.e. a link) between two concept types, establish a representation link between two concepts, identify the coreference set of a referent, extract a subset of the concept type lattice according to a user's viewpoint.

C-VISTA was tested in road accident analysis [13]: we built an ontology on traffic accident analysis, based on the different viewpoints of seven experts (two specialists in psychology three infrastructure engineers and two vehicle engineers). All the examples in the paper were based on this application.

C-VISTA was also tested in the framework of a memory of a concurrent engineering project in aeronautics [12] [13], with the objective of representing the artefact to be designed by several participants. For this application, we needed to introduce a different viewpoint template, shown in Figure 10.

<i>Focus</i>	DesignView: Material view Task: Building of the Brake component Step: 4
<i>View angle</i>	Participant: Mr X Skill field / Level: Mechanics / Expert Objective: Description Satisfied requirements: Cost reduction

Fig. 10. Example of viewpoint for a concurrent engineering project memory in aeronautics

5.2 Related Work

As noticed above, related work on viewpoints in object-oriented representation [2] [11] [10] [1] generally handles perspectives, while C-VISTA model enables to represent not only *perspectives* for consensual knowledge but also *opinions* for non consensual knowledge. C-VISTA also characterizes a viewpoint by its *focus* and its *view angle*. Moreover, it exploits the relational features of CG formalism, by expression of *equivalence*, *inclusion links*, and *exclusion links* between the concept types. C-VISTA could be extended by other links (e.g. inter-ontology relations [18] or composition of viewpoints [1]). Integration of viewpoints in CG was offered in [12] [17].

Our method for using C-VISTA model can be compared to methods for building cooperatively ontologies [6] [8] [16] or terminological concept bases [7]. Our approach is different from the methods for integration of ontologies proposed in [4] or [15] or for comparison of conceptual graphs from several experts [5]. Our method enables to build a multi-viewpoints ontology, with cohabitation of several possibly contradictory viewpoints, but each viewpoint itself being coherent. We must notice that, though it was presented in the framework of conceptual graph formalism, the C-VISTA model can be exploited in the framework of other formalisms. As a conclusion, using viewpoints enables a more accurate knowledge modelling from several experts and a user-oriented access to the ontology thus organized through viewpoints.

References

1. Acker, L. and Porter, B. Extracting Viewpoints from Knowledge Bases. Proc. of AAAI'94. (1994).
2. Carré, B. and Dekker, L. The point of view notion for multiple inheritance. Proc. of ECOOP/OOPSLA'90, Ottawa, Canada, (1990).
3. Chein, M. and Mugnier, M.-L. Michel Chein and Marie-Laure Mugnier. Conceptual graphs: fundamental notions. RIA, 6(4): 365-406. (1992).
4. Dieng, R. and Hug, S. Comparison of "personal ontologies" represented through conceptual graphs. In H. Prade ed, Proc. of the 13th European Conference on Artificial Intelligence (ECAI'98), Wiley & Sons, p. 341-345, Brighton, UK, (1998).
5. Dieng, R. and Hug, S. MULTIKAT, a Tool for Comparing Knowledge from Multiple Experts. In M. L. Mugnier, M. Chein eds, Conceptual Structures: Theory, Tools and Applications, Proc. of the 6th Int. Conference on Conceptual Structures (ICCS'98), Montpellier, (1998), Springer-Verlag, LNAI 1453
6. Euzenat, J. Corporate memory through Cooperative Creation of Knowledge-Based Systems and Hyper-Documents. Proceedings of the 10th Banff Knowledge Acquisition for Knowledge-Based Systems Workshop (KAW'96), Banff, Canada, November, (1996).
7. Falquet, G. and Mottaz Jiang, C.-L. Conflict Resolution in the Collaborative Design of Terminological Knowledge Bases. In R. Dieng & O. Corby eds, Knowledge Engineering and Knowledge Management: Methods, Models and Tools, Proc. of the 12th International Conference, EKAW'2000, Springer-Verlag, LNAI 1937, Juan-les-Pins, (2000), p. 156-171.
8. Garcia, C. Cooperative building of an ontology within multi-expertise framework. Proc. of the 2nd International Conference on Cooperative Systems (COOP'96), Juan-les-Pins, June, (1996), p. 435-454.
9. Haemmerlé, O. CoGITO: une plate-forme de développement de logiciel sur les graphes conceptuels. PhD Thesis, Université de Montpellier II, (1995).
10. Marcaillou, S. , Coulette, B., and Vo, D. An approach to viewpoint modelling. Proc. of TOOLS'93, (1993).
11. Marino, O., Rechenmann, F. and Uvietta, P. Multiple Perspectives and Classification Mechanism in Object-Oriented Representation. Proc. of ECAI'90, Stockholm, (1990), p. 425-430.
12. Ribière, M. Using viewpoints and CG for the representation and management of a corporate memory in concurrent engineering. In M. L. Mugnier, M. Chein eds, Conceptual Structures: Theory, Tools and Applications, Proc. of the 6th Int. Conference on Conceptual Structures (ICCS'98), Montpellier, August 10-12, (1998), p. 94-108.
13. Ribière, M. Représentation et gestion de multiples points de vue dans le formalisme des graphes conceptuels, PhD Thesis, Université de Nice - Sophia Antipolis, 19 April 1999.
14. Sowa, J. Conceptual Graphs: Information Processing in Mind and Machine. Reading, Addison Wesley, (1984).

15. Tamma, V. A. M. and Bench-Capon, T. J. M. Supporting Inheritance Mechanisms in Ontology Representation. In R. Dieng & O. Corby eds, Knowledge Engineering and Knowledge Management: Methods, Models and Tools, Proc. of the 12th International Conference, EKAW'2000, Springer-Verlag, LNAI 1937, Juan-les-Pins, (2000) p. 140-155.
16. Tennison, J. and Shadboldt, N. APECKS, a Tool to Support Living Ontologies. Proceedings of the 11th Banff Knowledge Acquisition for Knowledge-Based Systems Workshop (KAW'98), Banff, Canada (1998).
17. Thanitsukkarn, T. and Finkelstein, A. A Conceptual Graph Approach to Support Multiperspective Development Environments. Proceedings of the 11th Banff Knowledge Acquisition for Knowledge-Based Systems Workshop (KAW'98), (1998).
18. Visser, P. R. S., Jones, D. M. , Bench-Capon, T. J. M. and Shave, M. J .R. Assessing Heterogeneity by Classifying Ontology Mismatches. Proc. of FOIS'98, Trento, Italy, (1998).