MULTIKAT, a Tool for Comparing Knowledge of Multiple Experts

Rose Dieng[1], Stefan Hug[2]
[1] INRIA, ACACIA Project, 2004 Route des Lucioles, BP 93,
06902 Sophia-Antipolis Cedex, France
Rose.Dieng@sophia.inria.fr
[2] PTV Planungsbüro Transport und Verkehr GmbH, Stumpfstraße 1,
76131 Karlsruhe, Germany
Stefan.Hug@ptv.ptv.de

Abstract. This paper presents MULTIKAT, a tool aimed at conflict management during knowledge modeling from multiple experts: this tools allows to compare knowledge of several experts both automatically and cooperatively, when such knowledge is represented through Sowa's conceptual graph formalism. MULTIKAT implements an algorithm of comparison and integration of several supports, and an algorithm of comparison and integration of multiple conceptual graphs corresponding to different viewpoints, the integration being guided by different integration strategies. This paper details this last algorithm.

1 Introduction

The development of a corporate memory or of a knowledge-based system in a company may involve knowledge acquisition from several experts, that can stem from the same domain or from different ones. Therefore, the knowledge engineer (KE) must detect and solve several kinds of conflicts: (a) differences of terminology, (b) incompatibility between terminologies, (c) differences between compatible reasonings (i.e. the experts use different problem solving methods but obtain non contradictory results), (d) incompatibility of reasonings (i.e. the different problem solving methods used by the experts lead to contradictory results). Few knowledge acquisition methods or tools take into account expertise conflict management: study of terminology conflicts and comparison of repertory grids in [7] [17]; management of several viewpoints in [6]; conflict detection in the framework of KADS-I methodology in [2]; cooperative building of a common ontology from several experts [8] [19].

After the KE elicited rough data from the different experts, she must analyze the elicited data in order to build: a) a *common model* corresponding to the kernel of knowledge common to all experts and perhaps models common only to sub-groups of experts, b) *specific models* corresponding to knowledge specific to an expert and not shared by other experts. Two approaches are possible: (1) either the KE tries to build such models (the common one and the specific ones) directly from the rough data, or (2) she builds separately each model of expertise corresponding to each expert (independently of the others) and then tries to compare the obtained models of expertise in order to find their common parts and their specific parts. In this second case, the common and specific models are obtained not directly from the rough data but from the separate expertise mo-

M.-L. Mugnier and M. Chein (Eds.): ICCS'98, LNAI 1453, pp. 139-153, 1998

dels.

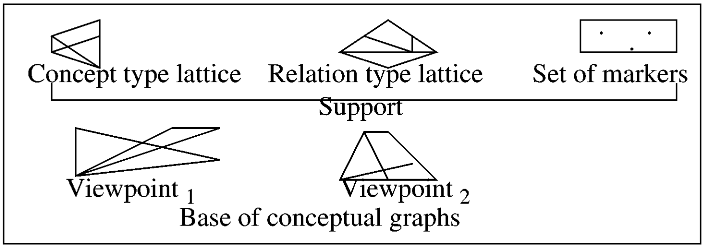

Fig. 1. Expertise model of an agent.

In [2], we had proposed techniques for comparing knowledge graphs representing knowledge of multiple experts. In [3] [4], we adapted such techniques to Sowa's conceptual graph (CG) formalism [18]. As shown in Fig. 1, to each expert we associate an artificial agent having a *support* (made of a concept type lattice, a relation type lattice, and a set of markers satisfying a conformity relation w.r.t. the concept type lattice) and a *base of canonical conceptual graphs*, built on this support and partitioned in several viewpoints. The purpose of this paper is to detail the tool MULTIKAT [12] implementing such techniques.

After a brief summary of the model of CG formalism, we will first present our algorithm of comparison of CGs implemented in MULTIKAT: this algorithm relies on the *local and global relations established between the graphs to be compared.* Then we will present the *integration strategies* offered by MULTIKAT for guiding the construction of the integrated graph. Throughout the paper, we will illustrate the tool through examples in road accident analysis.

2 Conceptual Graph Model

We rely on the model of simple CGs, as defined in [18] [1] [20]. A support S is a tuple = ($\mathcal{T}$c, $\mathcal{T}$r, $\mathcal{B}$, $\mathcal{M}$, conf) where : ($\mathcal{T}$c, ≤, T, ⊥) is a lattice of concept types, $\mathcal{T}$r is a partially ordered set of relation types (we will suppose that ($\mathcal{T}$r , ≤, T_{Tr}, $\bot_{Tr}$) has also a structure of lattice), $\mathcal{B}$ is a set of "star graphs" in bijection with $\mathcal{T}$r, and indicating the signature of each relation type, M is a set of individual markers (M ∪ {0, *} has a structure of lattice), conf is a conformity relation, that relates type labels to individual markers.

A CG defined with respect to a support S is a connected, bipartite, labelled graph, ($\mathcal{C}$, $\mathcal{R}$, $\mathcal{E}$, *label*) with labeled vertices (the labelling respecting some constraints): $\mathcal{C}$ is the set of concept nodes (or C-vertices), $\mathcal{R}$ is the set of relation nodes (or R-vertices), $\mathcal{E}$ is the set of edges, *label* is a function of $\mathcal{C} \cup \mathcal{R} \cup \mathcal{E}$ that associates to a C-vertex or R-vertex or an edge its label.

Moreover, we will exploit the following characteristics:

– We define the function adj : N * $\mathcal{R} \rightarrow \mathcal{C}$: rel being a R-vertex of arity n and i being an integer ∈ N, adj (i, rel) is the ith C-vertex adjacent to the R-vertex rel if i ≤ n and () otherwise.

– To a given CG, we associate $\mathcal{A}$ the set of its "elementary links" denoted rel (C_1, ..., C_n), with rel $\in \mathcal{R}$, arity(rel) = n, and $\forall$ i $\in$ [1..n], C_i = adj (i, rel) $\in \mathcal{C}$. Therefore, we will rather consider a CG as a tuple ($\mathcal{C}$, $\mathcal{R}$, $\mathcal{A}$, $\mathcal{E}$, *label*) and we will rather use the simplified notation G = ($\mathcal{C}$, $\mathcal{R}$, $\mathcal{A}$) .
– We make some simplifying hypotheses:
 $\forall$ rel_1 and $rel_2 \in \mathcal{T}r \setminus \{T_{Tr}, \bot_{Tr}\}$ such that $rel_1 < rel_2$, we have:
 arity (rel_1) = arity (rel_2) and type (adj (i, B (rel_1))) $\leq$ type (adj (i, B (rel_2))).
– To each concept type or relation type, a set of synonyms of its main name can be associated.
– To each relation type, a list of incompatible relation types is associated.

In this paper, we will use the following vocabulary:

– *«concept specialization»* or restriction of a concept type: on a concept-node [type:ref], type is replaced by one of its subtypes,
– *«instantiation»* or restriction of a referent : on a concept-node [type:*ref], the generic marker *ref is replaced by an individual marker conform with type,
– *«relation specialization»* or restriction of a relation type : on a R-vertex denoted (type-rel) , type-rel is replaced by one of its subtypes,
– *«concept generalization»*: on a concept-node [type:ref], type is replaced by a supertype,
– *«conceptualization»*: on a concept-node [type:ind-ref], the individual marker ind-ref is replaced by a generic marker conform with type,
– *«relation generalization»*: on a R-vertex (type-rel), type-rel is replaced by a supertype of type-rel.

3 Algorithm of Integration of two Expertise Models

During the knowledge modelling phase, for each expert, the KE describes the concept and relation types handled by this expert through the support associated to the agent representing this expert. For each expert, the KE can also indicate which concept types correspond to the competence domain of this expert (e.g. all concept types subtypes of Infrastructure will be marked as specialties of the infrastructure engineer)[1].The base of canonical CGs associated to an agent can be partitioned according to viewpoints. For example, in an application of traffic accident analysis, in addition to a *task* viewpoint, the experts handle CGs focusing on the *drivers*, the *vehicles*, the *infrastructure*, the *interaction driver-vehicle*, the *interaction driver-infrastructure* and the *interaction vehicle-infrastructure*.

The algorithm of comparison of the expertise models of two agents is based on the following steps:

1.This information will be exploited later if a strategy of the greatest competence is chosen for building the integrated graph.

1. Comparison of the two supports and building of the common support. It can be decomposed in:
   – Comparison and fusion of the concept type lattices of the two experts.
   – Comparison and fusion of the relation type lattices used by the two experts.
   – Comparison and fusion of the two sets of markers.
2. Comparison of the two bases of CGs : for each viewpoint for which both agents have associated CGs, compare the two corresponding CGs, G_1 and G_2. This comparison of CGs of the same viewpoint can be decomposed as follows:
   – Preprocessing: In each CG of an agent, replace the expert's terms by the agreed terms adopted in the common lattices of concept types and of relation types.
   – Set up local relations between comparable "elementary links" of both graphs.
   – Set up global relations between G_1 and G_2.
3. Construction of the base of integrated graphs, according to the chosen integration strategy: by exploiting the local and global relations previously established, build the integrated CGs for each viewpoint (unless G_1 and G_2 were recognized as incompatible).

The comparison and fusion of supports is detailed elsewhere [12] [5]. In this paper we will focus only on the comparison of CGs, i.e. the steps 2 and 3 of the algorithm. Therefore, from now on, we will suppose the common support $S_{com} = (\mathcal{T}_{c\text{-}com}, \mathcal{T}_{r\text{-}com}, \mathcal{B}_{com}, \mathcal{M}_{com}, conf_{com})$ has been built.

4 Local Relations among C-vertices of Different Conceptual Graphs

During the elicitation sessions, the experts may express themselves with different levels of precision or of abstraction, they may exploit (or not) abstract knowledge, examples, particular cases... Therefore strategies for comparing two experts may be based on criteria of generalization vs specialization, conceptualization vs instantiation. In order to exploit such strategies, the KE must be able to detect global relations among the CGs describing the respective knowledge of the different experts. These global relations among CGs rely on the local relations among their elementary links. Such local relations rely themselves on the local relations among the concepts appearing in both graphs. The purpose of all such relations is to help to determine what is the most specialized (resp. generalized) or the most instantiated (resp. conceptualized) among two knowledge elements to be compared. This section will describe the local relations among concepts, section 5 the local relations among elementary links and section 6 the global relations among CGs. For lack of room, we don't detail all the relation definitions that can be found in [4] [12].

4.1 Definitions

Let $G_1 = (\mathcal{C}_1, \mathcal{R}_1, \mathcal{A}_1)$ and $G_2 = (\mathcal{C}_2, \mathcal{R}_2, \mathcal{A}_2)$ the CGs of $Expert_1$ and of $Expert_2$, corresponding to the same viewpoint v and based on the common support S_{com}.

Definitions of Binary Relations on $\mathcal{C}_1 * \mathcal{C}_2$

Let C_1 be a C-vertex in G_1 and C_2 a C-vertex in G_2.

- *is-same-concept* (C_1, C_2) iff type (C_1) = type (C_2) $\wedge$ referent (C_1) = referent (C_2),
- *is-generalization* (C_1, C_2) iff type (C_2) < type (C_1) $\wedge$ referent (C_1) = referent (C_2),
  is-specialization (C_1, C_2) iff is-generalization (C_2, C_1),
- *is-conceptualization* (C_1,C_2) iff type(C_1) = type(C_2) $\wedge$ referent(C_2) < referent(C_1),
  is-instantiation (C_1, C_2) iff is-conceptualization (C_2, C_1),
- *is-generalization&conceptualization* (C_1, C_2) iff type (C_2) < type (C_1) $\wedge$ referent (C_2) < referent (C_1),
  is-specialization&instantiation (C_1, C_2) iff is-generalization&conceptualization (C_2, C_1),
- *is-more-generalized* (C_1, C_2) iff is-generalization (C_1, C_2) $\vee$ is-generalization&conceptualization (C_1, C_2),
  is-more-specialized (C_1, C_2) iff is-more-generalized (C_2, C_1),
- *are-comparable-concepts* (C_1, C_2) iff type (C_1) $\cup$ type (C_2) $\neq$ T ,
- *possible-common-specialization* (C_1, C_2) iff type (C_1) $\cap$ type (C_2) $\neq$ $\perp$.

Definitions of Functions: $\mathcal{C}_1 * \mathcal{C}_2 \rightarrow \mathcal{C}_{com}$

- *most-generalized* (C_1, C_2) =
  - C_1 iff is-same-concept (C_1, C_2) $\vee$ is-more-generalized (C_1, C_2),
  - C_2 iff is-more-generalized (C_2, C_1),
  - () otherwise.
- *common-generalization* (C_1, C_2) =
  - [$type_{com}$: ref] iff $type_{com} \neq$ T
  - (with $type_{com}$ = type (C_1) $\cup$ type (C_2) and ref = referent (C_1) $\cup$ referent (C_2))
  - () otherwise.
- *most-specialized* (C_1, C_2) =
  - C_1 iff is-same-concept (C_1, C_2) $\vee$ is-more-specialized (C_1, C_2),
  - C_2 iff is-more-specialized (C_2, C_1),
  - () otherwise.
- *common-specialization* (C_1, C_2) =
  - [$type_{com}$: ref] iff $type_{com} \neq$ T $\wedge$ ref $\neq 0$
  - (with $type_{com}$ = type (C_1) $\cap$ type (C_2) and ref = referent (C_1) $\cap$ referent (C_2)),
  - () otherwise.
- *most-conceptualized* (C_1, C_2) =
  - C_1 iff is-same-concept (C_1, C_2) $\vee$ is-conceptualization (C_1, C_2),
  - C_2 iff is-conceptualization (C_2, C_1),
  - () otherwise.

– *common-conceptualization* (C_1, C_2) =

[$type_{com}$: ref] iff type (C_1) = type (C_2)

(with $type_{com}$ = type (C_1) = type (C_2) and ref= referent $(C_1) \cup$ referent (C_2))

() otherwise.

– *most-instantiated* (C_1, C_2) =

C_1 if is-same-concept $(C_1, C_2) \vee$ is-instantiation (C_1, C_2),

C_2 if is-instantiation (C_2, C_1),

() otherwise.

– *common-instantiation* (C_1, C_2) =

[$type_{com}$: ref] iff type (C_1) = type $(C_2) \wedge$ ref $\neq 0$

(with $type_{com}$ = type (C_1) = type (C_2) and ref =referent $(C_1) \cap$ referent (C_2))

() otherwise.

4.2 Examples

is-generalization ([Driver's-Error : err-Paul], [Wrong-Distance-Evaluation : err-Paul])
is-conceptualization ([Vehicle-Control-Loss : *x], [Vehicle-Control-Loss : err-Jean])
is-specialization&instantiation ([Wrong-Manoeuvre : err-Fred], [Driver's-Error : *y])
is-more-specialized ([Vehicle-Control-Loss : err-Jean], [Driver's-Error : *y])
are-comparable-concepts ([Obstacle-Perception-Lack : *x], [Indicator-not-put : *y])
most-generalized ([Driver's-Error : *y], [Vehicle-Control-Loss : err-Jean]) =
 [Driver's-Error : *y]
common-generalization ([Vehicle-Control-Loss : err-Jean], [Wrong-Manoeuvre : err-Fred]) = [Action-Error : *]
common-specialization ([Driver's-Error : *y], [Vehicle-Control-Loss : err-Jean]) =
 [Vehicle-Control-Loss : err-Jean]

5 Local Relations among Elementary Links of Conceptual Graphs

5.1 Definitions of Functions on $\mathcal{A}_1 * \mathcal{A}_2$

Let $l_1 = r_1(C_{11} \ldots C_{1n})$ and $l_2 = r_2(C_{21} \ldots C_{2n})$ be elementary links of G_1 and G_2. MULTIKAT implements several possible relations between them:

– *is-same-link* (l_2, l_1) iff

type(r_1) = type $(r_2) \wedge \forall$ i $\in$ [1, n], is-same-concept (adj (i, r_2), adj (i, r_1))

– *are-incompatible-links* (l_2, l_1) iff

incompatible(type(r_1), type(r_2)) $\wedge \forall$ i $\in$ [1, n], is-same-concept(adj(i, r_2),adj (i, r_1))

– *are-comparable-links* (l_2, l_1) iff

type$(r_1) \cup$ type$(r_2) \neq T \wedge \forall$ i $\in$ [1, n], are-comparable-concepts (adj(i, r_2), adj(i, r_1))

Specialization

– *is-relation-specialization* (l_2, l_1) iff

type(r_2) < type $(r_1) \wedge \forall$ i $\in$ [1, n], is-same-concept (adj (i, r_2), adj (i, r_1))

– *is-concept-total-specialization* (l_2, l_1) iff

$\text{type}(r_1) = \text{type}(r_2) \wedge \forall i \in [1, n]$, is-specialization (adj (i, r_2), adj (i, r_1))

– *is-concept-partial-specialization* (l_2, l_1) iff

$\text{type}(r_1) = \text{type}(r_2)$

$\wedge \forall i \in [1, n]$, (is-specialization (adj (i, r_2), adj (i, r_1))

$\vee$ is-same-concept (adj (i, r_1), adj (i, r_2)))

$\wedge \exists i \in [1, n]$, is-specialization (adj (i, r_2), adj (i, r_1))

The relations *is-relation&concept-total-specialization* and *is-relation&concept-partial-specialization* can be defined by the same way.

Generalization. Generalization relations are defined thanks to specialization relations.

Instantiation

– *is-total-instantiation* (l_2, l_1) iff

$\text{type}(r_1) = \text{type}(r_2)$

$\wedge \forall i \in [1, n]$, type (adj (i, r_1)) = type (adj (i, r_2))

$\wedge$ referent (adj (i, r_2) < referent (adj (i, r_1)))

We also define the relation *is-partial-instantiation.*

Conceptualization. Conceptualization relations are defined w.r.t. instantiation ones.

Specialization and Instantiation

– *is-relation-specialization&total-instantiation* (l_2, l_1) iff

$\text{type}(r_2) < \text{type}(r_1) \wedge \forall i \in [1, n]$, is-instantiation (adj (i, r_2), adj (i, r_1))

Other relations such as *is-relation-specialization&partial-instantiation*, *is-concept-total-specialization&total-instantiation* or *is-concept-total-specialization&partial-instantiation* can be defined by the same way (see [4] [12]).

Generalization and Conceptualization.«Generalization and Conceptualization» relations are defined thanks to «Specialization and Instantiation» relations

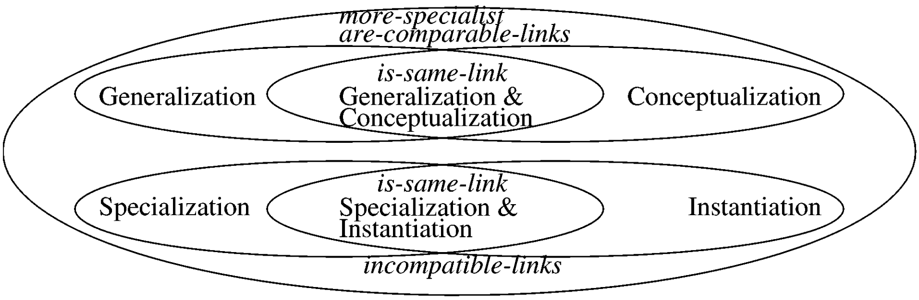

Fig. 2. Clustering of possible relations between elementary links

Competence Domain. If nb-specialties (Expert$_i$, l_j) is, in the link l_i, the number of concept types marked by the KE as corresponding to Expert$_i$'s domain of competence:

– *More-specialist* (l_1, l_2) =

l_1 iff nb-specialties $(Expert_1, l_1)$ > nb-specialties $(Expert_2, l_2)$

l_2 iff nb-specialties $(Expert_2, l_2)$ > nb-specialties $(Expert_1, l_1)$

() otherwise.

Fig. 2 shows the clustering of possible relations among elementary links.

Examples. In the example shown Fig. 3, in G_1, let us denote:

l_{11} = [Driver] → (Moving-with) → [Vehicle],

l_{12} = [Vehicle] → (Possible-Influence) → [Driver's-Error]

l_{13} = [Age] → (Possible-Influence) → [Driver's-Error],

l_{14} = [Driver] → (Characteristic) → [Age]

In G_2, let us denote :

l_{21} = [Driver] → (Moving-with) → [GTI],

l_{22} = [GTI] → (Incites-to) → [Excessive-Speed]

l_{23} = [GTI] → (Incites-to) → [Vehicle-Control-Loss],

l_{24} = [Driver] → (Characteristic) → [Age:young]

l_{25} = [Age:young] → (Possible-Influence) → [Excessive-Speed]

Between these graphs, the following local relations among their elementary links exist:

is-concept-partial-generalization (l_{11}, l_{21}),

is-relation&concept-total-generalization (l_{12}, l_{22})

is-relation&concept-total-generalization (l_{12}, l_{23}),

is-concept-partial-conceptualization (l_{14}, l_{24})

is-concept-partial-generalization&partial-conceptualization (l_{13}, l_{25})

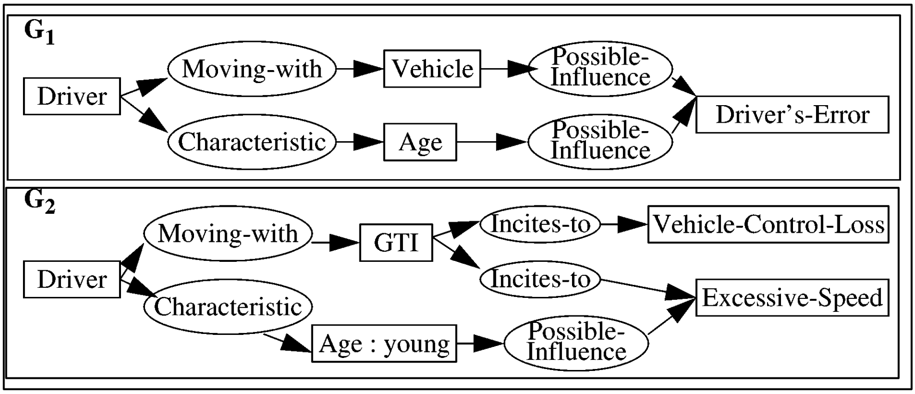

Fig. 3. Conceptual graphs of two experts, on the viewpoint interaction-vehicle-driver

5.2 Definitions of Functions: $\mathcal{A}_1 * \mathcal{A}_2 \rightarrow \mathcal{A}_{com}$

– *most-specialized*(l_1, l_2) = l_1 iff ∃ a specialization relation between l_1 and l_2

l_2 iff $\exists$ a specialization relation between l_2 and l_1
() otherwise.

– *common-specialization* (l_1, l_2) =
most-specialized(l_1, l_2) iff most-specialized$(l_1, l_2) \neq ()$
$r(c_1, ..., c_n)$ iff type $(r) \neq T \wedge \forall i \in [1, n], c_i \neq ()$
(with $r = r_2 \cap r_1$ and $\forall i \in [1, n], c_i$ = common-specialization (adj(i, r_2), adj(i, r_1)))
() otherwise.

– *most-generalized* (l_1, l_2) = l_2 iff $\exists$ a specialization relation between l_1 and l_2
l_1 iff $\exists$ a specialization relation between l_2 and l_1,
() otherwise.

– *common-generalization* (l_1, l_2) =
most-generalized (l_1, l_2) iff most-generalized $(l_1, l_2) \neq ()$
$r(c_1, ..., c_n)$ iff type $(r) \neq T_{tr} \wedge \forall i \in [1, n], c_i \neq ()$
(with $r = r_1 \cup r_2$ and $\forall i \in [1, n], c_i$ = common-generalization (adj(i, r_2), adj(i, r_1)))
() otherwise.

– *most-instantiated* (l_1, l_2) = l_1 iff $\exists$ an instantiation relation between l_1 and l_2
l_2 iff $\exists$ an instantiation relation between l_2 and l_1
() otherwise.

– *common-instantiation* (l_1, l_2) = most-instantiated (l_1, l_2).

– *most-conceptualized* (l_1, l_2) = l_1 iff $\exists$ an instantiation relation between l_2 and l_1
l_2 iff $\exists$ an instantiation relation between l_1 and l_2
() otherwise.

– *common-conceptualization* (l_1, l_2) =
most-conceptualized (l_1, l_2) iff most-conceptualized $(l_1, l_2) \neq ()$
$r(c_1, ..., c_n)$ iff $r_1 = r_2 \neq T_{tr} \wedge \forall i \in [1, n], c_i \neq ()$
(with $r = r_1 = r_2$ and $\forall i \in [1, n], c_i$=common-conceptualization (adj(i, r_2), adj(i, r_1)))
() otherwise.

With the example of Fig. 3,
common-specialization $(l_{11}, l_{21}) = l_{21}$, common-generalization $(l_{12}, l_{22}) = l_{12}$
common-instantiation $(l_{13}, l_{25}) = l_{25}$, common-conceptualization $(l_{14}, l_{24}) = l_{14}$
common-generalization (l_{22}, l_{23}) = [GTI] $\rightarrow$ (incites-to) $\rightarrow$ [Action-error].

6 Global Relations among Conceptual Graphs

6.1 Definitions

Let $G_1 = (\mathcal{C}_1, \mathcal{R}_1, \mathcal{A}_1)$ and $G_2 = (\mathcal{C}_2, \mathcal{R}_2, \mathcal{A}_2)$ two CGs of Expert_1 and of Expert_2, corresponding to the same viewpoint v and based on the common support S_{com}.

We adapt as follows the definition of graph morphism proposed by [1].

Ψ-morphism. A 3-ple of functions : $(h_c: \mathcal{C}_1 \rightarrow \mathcal{C}_2, h_r\ \mathcal{R}_1 \rightarrow \mathcal{R}_2, h_a : \mathcal{A}_1 \rightarrow \mathcal{A}_2)$ is

called a Ψ-*morphism* from G_1 to G_2 (denoted $\Psi_{G_1 \to G_2}$) iff $\forall\, l_1 = r_1(C_{11}, \dots C_{1n}) \in \mathcal{A}_1$:
$h_a(l_1) = h_r(r_1)(h_c(C_{11}), \dots h_c(C_{1n})) \wedge \text{type}(h_r(r_1)) \leq \text{type}(r_1)$
$\wedge\ \forall\, i \in [1..n]$, type $(h_c(c_{1i})) \leq$ type $(c_{1i}) \wedge$ referent $(h_c(c_{1i})) \leq$ referent type (c_{1i}).

We define several binary relations on $G_1 * G_2$.

Same graph, Subgraph, Supergraph, Incompatibility

- *IsSameGraph* (G_1, G_2) iff $\exists$ a bijective Ψ-morphism $\Psi_{G_1 \to G_2} = (h_c, h_r, h_a)$ such that $\forall\, l_1 \in \mathcal{A}_1$, is-same-link $(l_1, h_a(l_1))$.
- *IsSubgraph* (G_2, G_1) iff $\exists$ a Ψ-morphism $\Psi_{G_2 \to G_1} = (h_c, h_r, h_a)$ such that is-same-link $(h_a(l_2), l_2)$.
- *IsSupergraph* (G_2, G_1) iff *IsSubgraph* (G_1, G_2).
- *IncompatibleGraphs* (G_1, G_2) iff $\exists\, l_1 \in \mathcal{A}_1, \exists\, l_2 \in \mathcal{A}_2$ such that are-incompatible-links (l_1, l_2).

Specialization[1]**.** We define the relations *IsConceptTotalSpecialization*, *IsConceptPartialSpecialization*, *IsRelationTotalSpecialization*, *IsRelationPartialSpecialization*, *IsRelation&ConceptPartialSpecializationIs, Relation&ConceptTotalSpecialization.*

- *IsRelation&ConceptTotalSpecialization* (G_2, G_1) iff $\exists$ a Ψ-morphism $\Psi_{G_2 \to G_1} = (h_c, h_r, h_a)$ such that $\forall\, l_2 \in \mathcal{A}_2, l_1 = h_a(l_2) \in \mathcal{A}_1$ satisfies: is-relation&concept-total-specialization (l_2, l_1).

Generalization. The generalization relations are defined w.r.t. the specialization ones.

Instantiation. We define the relations *IsTotalInstantiation*, *IsPartialInstantiation,IsRelationSpecialization&TotalInstantiation* and *IsRelationSpecialization&PartialInstantiation*. For example:

- *IsRelationSpecialization&TotalInstantiation* (G_2, G_1) iff $\exists\, \Psi_{G_2 \to G_1}$ such that $\forall\, l_2 \in \mathcal{A}_2$: is-relation-specialization&total-instantiation $(l_2, h_a(l_2))$.

Conceptualization. The conceptualization relations are defined w.r.t. the instantiation relations.

For lack of room, we don’t detail the relations **«Specialization and Instantiation»** or **«Generalization and Conceptualization»** (see [4] [12] for more details).

According to the clustering of local relations among elementary links, the global relations between graphs are clustered as shown in Fig. 4. Based on this clustering, we

1.Notice that our specialization and generalization relations differ from the terminology usually adopted in Conceptual Graph community.

define the following functions:

– *most-specialized*(G_1, G_2) = G_1 iff ∃ a specialization relation between G_1 and G_2
G_2 iff ∃ a specialization relation between G_2 and G_1
() otherwise.

– *most-generalized* (G_1, G_2) = G_2 iff most-specialized(G_1, G_2) = G_1
G_1 iff most-specialized(G_1, G_2) = G_2
() otherwise.

– *most-instantiated* (G_1, G_2) = G_1 iff ∃ an instantiation relation between G_1 and G_2
G_2 iff ∃ an instantiation relation between G_2 and G_1
() otherwise.

– *most-conceptualized* (G_1, G_2) = G_2 iff most-instantiated (G_1, G_2) = G_1
G_1 iff most-instantiated (G_1, G_2) = G_2
() otherwise.

Remark: The previous Ψ-morphisms need not be injective. The interest to use or not surjective Ψ-morphisms is discussed in [12].

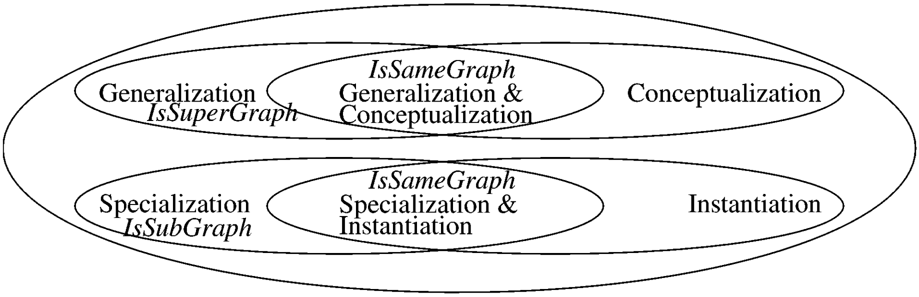

Fig. 4: Clustering of global relations among conceptual graphs.

Example. In the example of Fig. 3, the following relations hold:
IsRelation&ConceptPartialGeneralization&PartialConceptualization (G_1, G_2)
IsRelation&ConceptPartialSpecialization&PartialInstantiation (G_2, G_1)
Therefore, most-generalized (G_1, G_2) = G_1, most-specialized (G_1, G_2) = G_2
most-instantiated (G_1, G_2) = G_1, most-conceptualized (G_1, G_2) = G_2

7 Construction of the Base of Integrated Conceptual Graphs

7.1 Strategies of Integration

Unless the relation *IncompatibleGraphs* (G_1, G_2) was set up between both graphs, their integration is possible. The building of the integrated graph G_{com} from both graphs G_1 and G_2 is guided by a *strategy* for solving conflicts. To each strategy, two functions, f_{global} and f_{local} can be associated. If there exists a global relation between G_1 and G_2, the result of the function f_{global} associated to the current strategy will be included in G_{com}. If no such global relation exists, the local relations among comparable elementary links are exploited : in case of choice between two such comparable links $l_1 \in \mathcal{A}_1$, and $l_2 \in \mathcal{A}_2$ the result of the function f_{local} associated to the current strategy will be included in G_{com}. If the function f_{local} gives no result, both links l_1 and l_2 are included in G_{com}. The connection of the partial graphs thus obtained is performed with strategy-dependent join operations: we implemented another version of the maximal join operator [12]. Once the integrated graph created for each viewpoint, the common knowledge model is obtained. According to the integration strategy chosen by the KE, the graphs of this common knowledge model may represent or not the complete knowledge of both initial graph bases.

MULTIKAT supplies the KE with 9 predefined strategies among which she can make a choice. These strategies allow to carry out the integration according to various criteria. The «direct» strategies are better when the KE prefers to always restrict to what was explicitly expressed by at least one expert and takes no initiative for modifying the knowledge expressed by an expert. The «indirect» strategies are useful when the KE prefers to exploit the knowledge expressed by an expert, in order to take initiative for modifying (e.g. generalizing, specializing, conceptualizing or instantiating) the other expert's knowledge.

Strategy of the highest direct generalization:
Preconditions: An expert focuses on particular cases, while the other expert expresses general knowledge, valid in more general cases.
$f_{global}(G_1, G_2)$ = most-generalized (G_1, G_2); $f_{local}(l_1, l_2)$ = most-generalized (l_1, l_2)

Strategy of the highest indirect generalization:
Preconditions: The characteristics of the expert are the same as in the previous case.
$f_{global}(G_1,G_2)$=most-generalized (G_1, G_2); $f_{local}(l_1, l_2)$ = common-generalization (l_1, l_2)

Strategy of the highest direct specialization:
Preconditions: An expert is more specialized than the other, on a given aspect and uses more precise expressions.
$f_{global}(G_1, G_2)$ = most-specialized (G_1, G_2); $f_{local}(l_1, l_2)$ = most-specialized (l_1, l_2)

Strategy of the highest indirect specialization:
Preconditions: The preconditions on the experts are the same as in the previous case.
$f_{global}(G_1,G_2)$=most-specialized (G_1, G_2); $f_{local}(l_1, l_2)$ = common-specialization (l_1, l_2)

Strategy of the highest direct conceptualization:
Preconditions: An expert focuses on too particular cases and on too specific examples, while the other expert expresses general knowledge, at a better level of abstraction.
$f_{global}(G_1,G_2)$ =most-conceptualized (G_1, G_2); $f_{local}(l_1, l_2)$ =most-conceptualized (l_1, l_2)
If the function gives no result, both links l_1 and l_2 are included in G_{com}.

Strategy of the highest indirect conceptualization :
Preconditions: The characteristics of the expert are the same as in the previous case.
$f_{global}(G_1,G_2)$=most-conceptualized(G_1,G_2);
$f_{local}(l_1,l_2)$=common-conceptualization(l_1, l_2)

Strategy of the highest direct instantiation [1]:
Preconditions: An expert gives useful and precise examples.
$f_{global}(G_1, G_2)$ = most-instantiated (G_1, G_2); $f_{local}(l_1, l_2)$ = most-instantiated (l_1, l_2)

Strategy of the greatest competence:
Preconditions: An expert is known as having a higher level of competence in a given field.
$f_{local}(l_1, l_2)$ = specialist (l_1, l_2)

Strategy of experts' consensus:
Preconditions: (1) Both experts have the same level of competence in the considered field and the KE has no criterion for choosing one rather than the other. (2) Or, for «psychological» reasons, it is impossible to make a selection between both experts. (3) Or, the future KBS or corporate memory to be built is explicitly aimed at relying only on the intersecting knowledge of both experts.
$f_{local}(l_1, l_2) = l_1$ iff is-same-link (l_2, l_1)
the link on which both experts agree otherwise.

7.2 Method for Using Integration Strategies

Among those predefined strategies, the KE chooses the integration strategy, according to the individual characteristics of the experts and to their expertises: their specialities, the way they expressed during the elicitation sessions (level of precision, abstraction of their expressions, presence or absence of examples illustrating abstract knowledge, capability to abstract knowledge from particular cases...). The KE can choose a global strategy, to be applied throughout the integration algorithm, or, on the contrary, a local strategy to be changed according to the context. So, throughout a given session, the KE can combine the previously described integration strategies, with the help of the experts. For example, in traffic accident analysis, if the two psychologists are respectively known as specialists of GTI vehicle drivers and of drivers' errors, the KE can adopt (a) the strategy of the greatest competence when comparing parts of the graphs concerning drivers of GTI vehicles or drivers' errors, (b) otherwise, the strategy of the highest direct specialization whenever it can be applied.

Fig. 5 shows an example of integration with two different strategies.

8 Conclusions

8.1 Implementation and Further Work

The interested reader can find the details of all algorithms in [12]. MULTIKAT was implemented in C/ C++, above the COGITO platform [11]. We restricted to binary relations. The implementation of Ψ-morphisms relied on COGITO projection. We also implemented another version of the maximal join operator. For the interaction with the user (i.e. the KE), a graphic user interface was implemented using JAVA and JDK-1.1b, in order to allow a distant use over the Internet. The algorithm of comparison commu-

1.The definition of a strategy of the highest indirect instantiation seems difficult.

nicates with the user interface via a TCP/IP socket connection. It functions as a client-server communication. MULTIKAT was tested on different examples stemming from knowledge models in traffic accident analysis [4] [12]. As a further work, we will take into account the notion of viewpoints proposed in [16], in order to extend our criteria of comparison among the graphs.

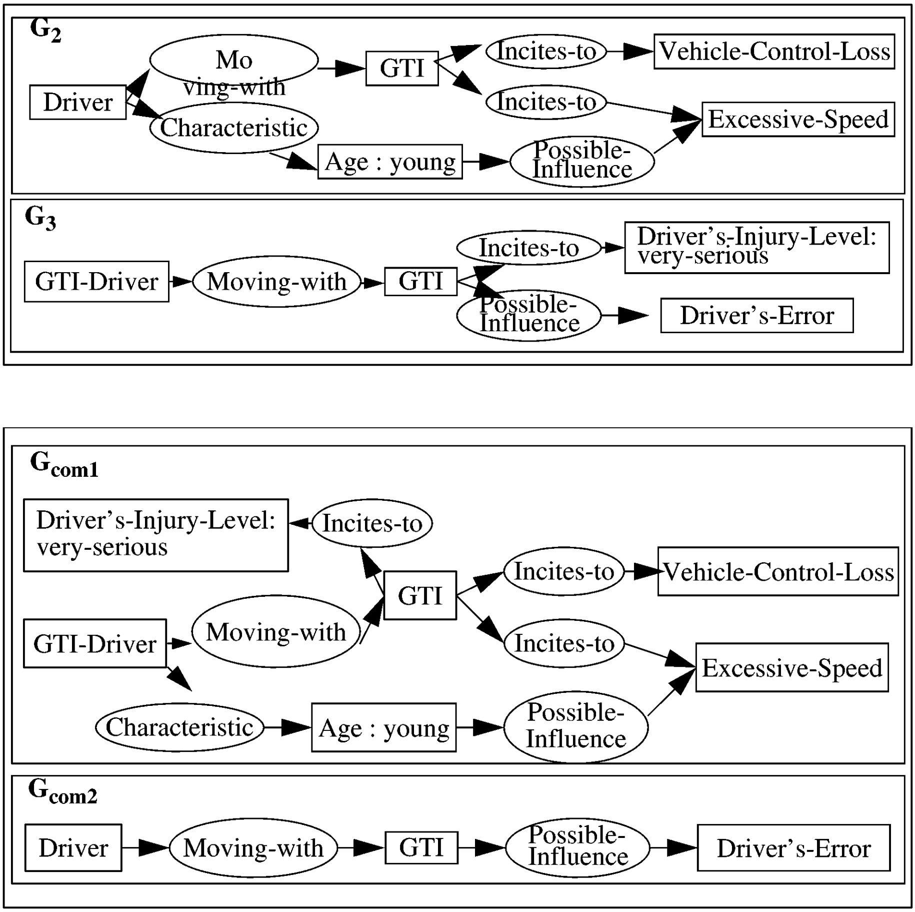

Fig. 5. Integrated graph of G_2 and G_3, with a strategy of the highest specialization (resp. generalization)

8.2 Related Work

In this paper, we described a tool MULTIKAT implementing an algorithm for comparison and integration of CGs representing knowledge of several experts. Techniques for comparing several viewpoints and solving conflicts among them are described in [6]. Techniques for integrating new knowledge into an existing knowledge base are proposed in [14]. Our techniques of comparison between several CGs can also be compared to research on algorithms for similarity-based matching of CGs [15] or for merging

CGs [9] or even on graph isomorphism [10]. Our exploitation of local and global relations among graphs, and of integration strategies seems original. Moreover, we combine both automatic and cooperative integration and we allow the combination of several comparison criteria thanks to several integration criteria.

References

1. Chein, M., Mugnier, M.-L. Conceptual Graphs: Fundamental Notions. *RIA,* **6**(4):365-406, (1992).
2. Dieng, R. Conflict Management in Knowledge Acquisition. In I. Smith (Ed), *AIEDAM, Special Issue on Conflict Management in Design*, **9**(4):337-351, (1995).
3. Dieng, R. Comparison of Conceptual Graphs for Modelling Knowledge of Multiple Experts. In Ras & al eds., *Foundations of Intelligent Systems, ISMIS'96*, Springer-Verlag, LNAI 1079, (1996) 78-87.
4. Dieng, R. *Comparison of Conceptual Graphs for Modelling Knowledge of Multiple Experts: Application to Traffic Accident Analysis.* INRIA Research Report RR-3161, (1997). Also in http://www.inria.fr/RRRT/RR-3161.html
5. Dieng, R., Hug., S. Comparison of «Personal Ontologies» Represented through Conceptual Graphs. To appear in Prade ed., *Proc. of ECAI'98,* John Wiley & Sons, Ltd, Brighton, UK (1998) 341-345.
6. Easterbrook, S. Handling conflict between domain descriptions with computer-supported negotiation. *Knowledge Acquisition*, **3**(3):255–289, (1991).
7. Gaines, B., Shaw, M. Comparing the Conceptual Systems of Experts. In *Proc. of IJCAI'89*, Detroit, USA (1989) 633–638.
8. Garcia, C. Co-Operative Building of an Ontology within Multi-Expertise Framework. *Proc. of the 2nd Int. Conf. on the Design of Cooperative Systems (COOP'96),* Juan-les-Pins (1996) 435-454.
9. Garner, B. J., Lukose, D. Knowledge Fusion. In H. D. Pfeiffer &al eds, *Conceptual Structures: Theory and Implementation*, Springer-Verlag, LNAI 754, Las Cruces, NM, USA (1992) 158–167.
10. Guinaldo, O. Conceptual Graph Isomorphism: Algorithm and Use. In Eklund et al, eds, *Conceptual Structures: Knowledge Representation as Interlingua*, Spring.-Verl., LNAI 1115, Sydney, Australia (1996) 160–174.
11. Haemmerlé, O. *CoGITo: une plate-forme de développement de logiciels sur les graphes conceptuels.* Ph.D thesis, Montpellier II University, France (1995).
12. Hug., S. *Vergleich von Begriffsgraphen für die Modellierung von Wissen mehrerer Experten.* Master Thesis, Univ. of Karlsruhe, Germany (1997).
13. Mineau, G. W., Allouche, M. Establishing a Semantic Basis: Toward the Integration of Vocabularies. In Gaines et al eds *Proc. of KAW'95*, pp. 2-1 – 2-16, Banff, Canada (1995).
14. Murray, K., Porter, B. Developing a Tool for Knowledge Integration: Initial Results. *Int.Journal of Man-Machine Studies*, **33**:373–383 (1990).
15. Poole, J., Campbell, J. A. A Novel Algorithm for Matching Conceptual and Related Graphs. In G. Ellis et al eds, *Conceptual Structures: Applications, Implementation and Theory*, Springer-Verlag, LNAI 954, Santa Cruz, CA, USA (1995) 293–307.
16. Ribière, M., Dieng, R. Introduction of Viewpoints in Conceptual Graph Formalism. In Lukose & al eds, *Fulfilling Peirce's Dream, ICCS'97,* Springer-Verlag, LNAI 1257, Seattle, USA,(1997) 168-182.
17. Shaw, M. L. G., Gaines, B. R. A methodology for recognizing conflict, correspondence, consensus and contrast in a knowledge acquisition system. *Knowledge Acquisition*, **1**(4):341–363, (1989).
18. Sowa, J. F. *Conceptual Structures: Information Processing in Mind and Machine*. Reading, Addison-Wesley, (1984).
19. Tennison, J., Shadboldt, N. APECKS: a Tool to Support Living Ontologies. In Gaines et al eds, *Proc. of the 11th Banff Workshop on Knowledge Acquisition, Modeling and Management (KAW'98)*, Banff, Canada, (1998). Also in http://ksi.cpsc.ucalgary.ca/KAW/KAW98/tennison/
20. Willems, M. Projection and Unification for Conceptual Graphs. In Ellis et al eds, *Conceptual Structures: Applications, Implementation and Theory*, Springer-Verlag, LNAI 954, Santa Cruz, CA, USA, (1995) 278-292.