# Formalizing partial matching and similarity in CBR with a description logic*

**Pascal Coupey** pascal.coupey@ura1507.univ-paris13.fr
**Christophe Fouqueré** christophe.fouquere@ura1507.univ-paris13.fr
**Sylvie Salotti** sylvie.salotti@ura1507.univ-paris13.fr
***LIPN-CNRS URA 1507,***
***Université Paris-Nord,***
***93430 Villetaneuse,***
***France***

## Abstract

Our aim is to use a description logic including default ($\delta$) and exception ($\epsilon$) connectives as a formal framework for a CBR system. This approach allows the retrieval of similar cases to be formalized. Subsumption and (sure, probable, typical and exceptional) inheritance relations of the description logic are the foundations for the different retrieval tasks: abstracting the new case, classifying it in the index base (full and partial matching), evaluating the similarity with *conceptual preference criteria* of the conceptual abstraction of the new case with the concepts of the index base, retrieving similar cases (instances) and applying *instance preference criteria* to order them. Our preference criteria are *symbolic* rather than *numerical* or those of *fuzzy logic*. Using description logic offers several advantages: the classification process can be used to retrieve similar cases, the formal properties and the efficiency of the system can easily be evaluated, preference criteria are homogeneously based on the formal description logic framework. Moreover preference criteria are independent of the knowledge and can thus be used in other applications.

**Key words :** case-based reasoning, retrieval, description logic, similarity, matching, default and exception.

## 1. Introduction

The main objective of case-based reasoning systems (CBR) is to resolve a problem (new case) by retrieving from a base similar problems (old cases) which have already been handled before and by reusing (modulo possible adaptations) the solutions which have been applied to these old cases. The application domains are varied: diagnosis, planification, decision-making, etc. Generally the whole CBR process is divided into several steps (Aamodt and Plaza, 1994):

1. *retrieval* of old cases,
2. *reuse* of the solutions,
3. *evaluation* of the chosen solutions,

* This research is supported as part of the "Creation and Enrichment of a Knowledge-Base: Application to the Supervision of a Telephone Network" project involving CNRS-cognisciences, CNET (National Center for Telecommunication Studies), INRIA, LIPN.

4. *learning* of the new case.

Each of these steps raises a lot of problems which are the subject of much research (WCBR89., 1989; WCBR91., 1991; EWCBR93., 1993; EWCBR94., 1994; Kolodner, 1993). In this paper, only the retrieval of old cases is considered. Considering a base of old stored cases the task is to compare a new case with the old cases in order to find "similar cases". The similar cases are then used to "resolve" the new case. If the new case coincides completely with an old case the new case is said to "(fully) match" the old case. If however, some properties do not coincide it is said to "partially match" the old case. The similarity varies according to certain *preference criteria* (e.g. the efforts made by the system to find these similar cases). There are therefore cases which are more similar than others.

Different solutions have been proposed in CBR research for the retrieval and preference criteria of similar cases (Kolodner, 1993). However these solutions are very often *informal* and *heterogeneous* since they are based on extremely different properties, and are very *dependent* on the described applications. Consequently it is very difficult to:

- evaluate the formal properties of these problems and proposed solutions such as decidability, soundness of the results or completeness;
- use these solutions in other applications;
- compare these solutions from one system to another;
- calculate the complexity of the algorithms used.

Several researchers (Koehler, 1994; Kamp, 1995; Napoli and Lieber, 1993) suggest using description logics (or terminological logics) to formalize the retrieval task of CBR systems. Description logics (DLs) are used to represent concept hierarchies (or terminologies). They are based on KL-ONE language (Brachman and Schmolze, 1985) and mostly formalize the idea of concept definition and reasoning about these definitions. DLs employ two kinds of formalisms for the representation of knowledge. The terminological formalism is used to describe conceptual knowledge (T-box) (concepts and their analytic interrelations) while assertional formalism allows facts to be stated (A-box) (e.g. an individual (or object) is an instance of a concept). Concepts are partially ordered by a subsumption relation: a concept $B$ is subsumed by a concept $A$ iff $A$ is more general than $B$. Based on this subsumption principle two kinds of inferences can be collectively referred to as classification. Classification of a new defined concept means automatically inserting it at the most specific place in the terminology and classification of an object (or recognition) consists in determining the most specific concepts the object is an instance of. An undeniable advantage of description logics is the automatic creation of the hierarchy from concept definitions.

For example in (Koehler, 1994) the author uses a description logic for the retrieval of plans in a case-based planning system. The author describes a system (MRL) where a terminological logic is used to create an index from the case description and to search for similar indexes with the help of the classification process. An index is a concept whose description is a conceptual abstraction[1] of properties that several old stored instances have

1 "Abstraction" of an object (i.e. instance) is a well known operation in description logic which consists in computing the most specific concept which generalizes this instance (cf. for example (Nebel, 1990)).

in common, i.e. it is a generalization of these instances. A new case is described with two components: a set of properties which describe the initial states ($pre_{new}$) of the new case and a set of properties which describe the final states ($goal_{new}$) to be achieved. The author goes on to propose two classifications in the MRL system: strong classification which needs the subsumption relation between $pre_{new}$ and $pre_{old}$ **and** between $goal_{old}$ and $goal_{new}$, and weak classification which needs only the subsumption relations between $pre_{new}$ and $pre_{old}$. The application suggested in Koehler's paper is for the reuse of plans to handle a mail tool under UNIX. The advantage of this approach is undeniable: description logics possess a formal semantics and a lot of research (Brachman and Levesque, 1984; Patel-Schneider, 1984; Nebel, 1988, 1990; Smolka, 1988; Schmidt-Schau$\beta$, 1989; Donini et al., 1991a, 1991b; Doyle and Patil, 1991; Dionne et al., 1993a; Borgida and Patel-Schneider, 1994; Schaerf, 1994) has described very subtle results concerning the calculability of the subsumption and classification operations, for various description languages.

However in the MRL system, the matching fails if there is no subsumption relation between $pre_{new}$ and $pre_{old}$ (i.e. no partial matching is proposed) and few criteria are proposed to find the "best" matching:

- For strong classification, the best candidate has the smallest number of subgoals which occur in $goal_{old}$ but not in $goal_{new}$.
- For weak classification, the best candidate has the largest number of subgoals which occur in the intersection of $goal_{old}$ and $goal_{new}$ but this number must be upper than the half of the subgoals in $goal_{new}$. Even if it seems to give good results in the described application, this is rather informal and empirical (why the *half*?). Moreover it is not possible to distinguish subgoals which *must be* from those which *may not be* in the intersection for the matching to succeed.

The solution proposed in this paper uses a description logic described in (Coupey and Fouqueré, 1994a, 1994b, 1995) to formalize the retrieval process of a CBR system. The particularity of this description logic is that it includes two new connectives: a connective to describe a by default concept ($\delta$) and a connective to describe an exception to a concept ($\epsilon$). The originality of this approach is the introduction of a *definitional point of view* for defaults and exceptions: defaults and exceptions are an integral part of concept definitions. Therefore they are used in subsumption and classification operations. The description logic is used to formalize search in the index base (like in (Koehler, 1994)), several preference criteria (*the most specific, sure/probable, typical/exceptional, the least number of exceptions*) and *partial matching* using subsumption, inheritance (inheritance relations are based on the subsumption relation and reflect the classical inferential point of view for defaults), default knowledge, exceptions and least common subsumption operation[2].

The application is a computer-assisted diagnosis system to help human supervisors diagnose and resolve incidents on the French telephone network (Coupey and Fouqueré, 1995; Biébow et al., 1994). This approach has many advantages:

- The formal framework allows the properties of the system to be clearly identified.

2 The least common subsumption operation computes the most specific concept which subsumes two concepts.

- The system is homogeneous since the entire reasoning process depends on formal relations and operations of the description logic.
- The complexity of the subsumption relation has been studied in detail, thus allowing the efficiency of the retrieval process to be evaluated easily.
- The preference criteria are various and independent of our application and can therefore be reused in other applications.
- The partial matching operations can be monitored by drawing a distinction between strict and default properties.

In the next section French telephone network supervision application led to this research is presented. This is followed by a partial description of the $\mathcal{ALN}_{\delta\epsilon}$ description logic with *default* ($\delta$) and *exception* ($\epsilon$) connectives (cf. (Coupey and Fouqueré, 1993, 1994b) for a complete presentation). Section 3 gives the definition of the definitional point of view of defaults, which allows the relations (subsumption and inheritance) between concepts and the instance-concept relations to be understood. These relations are the foundations for our similarity preference criteria in the retrieval process of our CBR systems (section 4). Section 5 describes some recent work combining description logics and CBR systems.

## 2. Brief presentation of the application

### 2.1 Global presentation

The main objective of this application is to interactively assist the human operators who supervise the French telephone network by automatically diagnosing a new incident and proposing:

1. A list of actions to be performed.
2. A list of stored old incidents which are similar to the new incident.

Figure 1 summarizes the steps required to process a new incident.

- (1) An incident occurs in the network, a set of parameters is sent to the supervision equipment.
- (2) Some alarms light up at the operator's workstation.
- (3) The operator makes a description of the incident via an interface to our system.
- (4) The system proposes an action form containing the actions the operator must perform, and a list of similar cases to this incident. Thus the operator can consult the full description of these similar incidents (the alarms that it generated, the duration, the consequences on the traffic, the actions that were performed and their results, etc.).
- (5) The operator executes commands to resolve the incident.

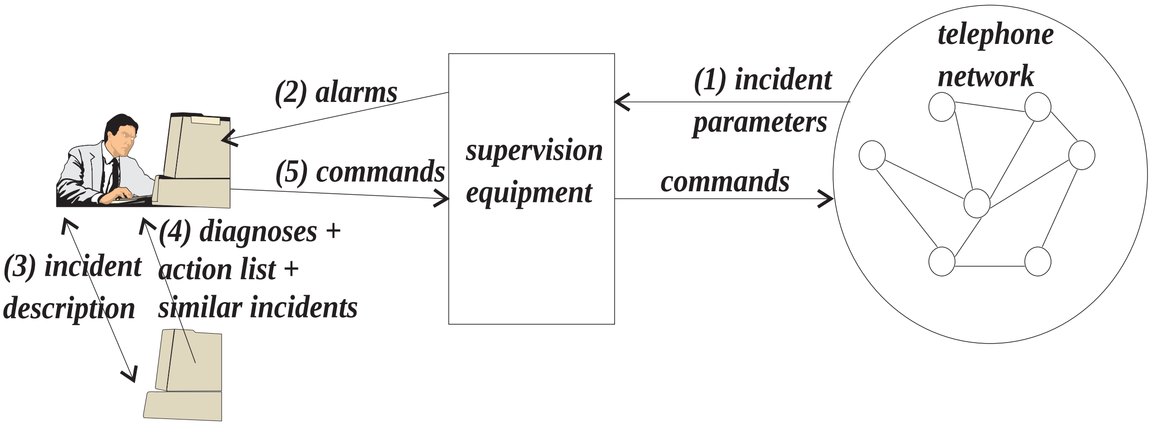


Figure 1: Processing of a new incident

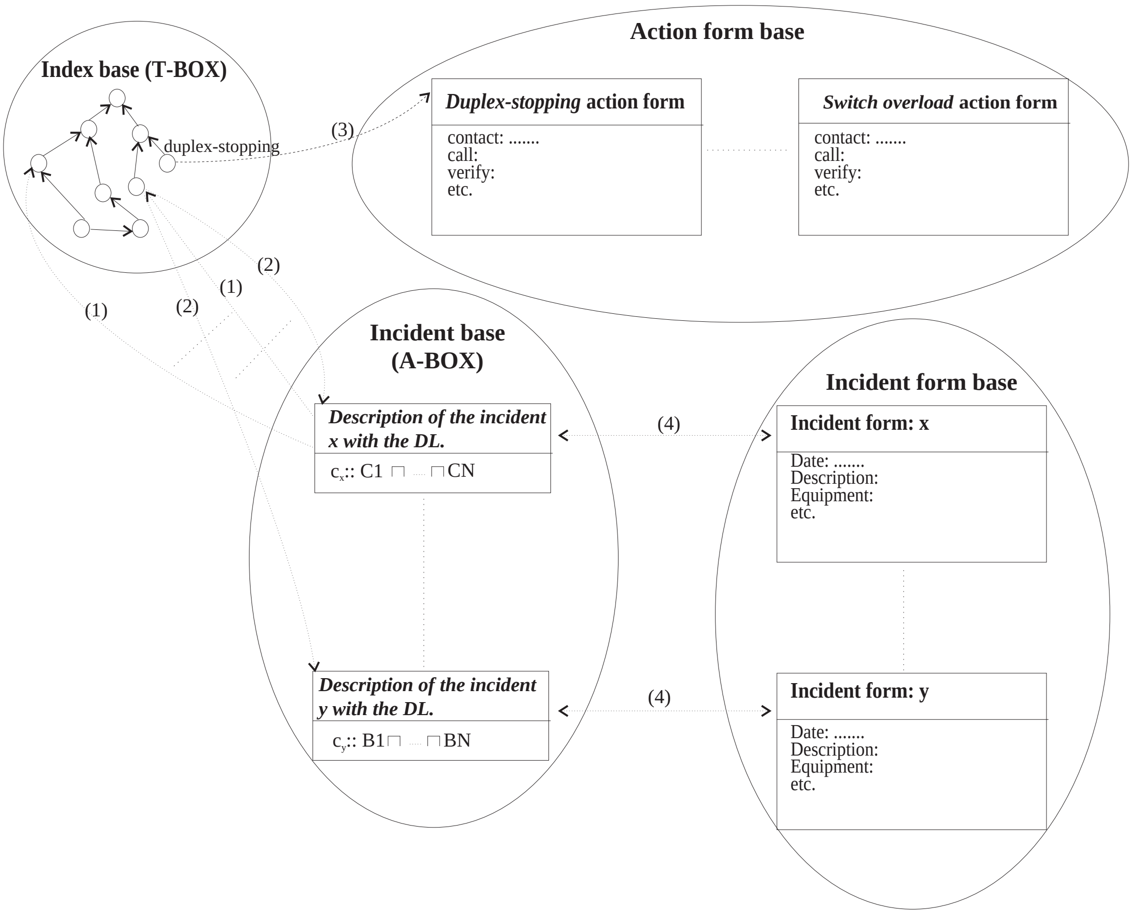


Figure 2: Knowledge bases in the application

| CTZP DUPLEX STOPPING | | |
|---|---|---|
| **1-Preparation phase (5 minutes after detection of the incident):** | | |
| **1.1-Improvements in information *** :** | | |
| LIST | in the base **** | the beams in the back zone of the CTZP, the CU of the corresponding CTZP group |
| PUT | under supervision, if they are not yet there, | the above beams and the V codified beams to the CTZP concentrator in the same numbering zone |
| IDENTIFY | ... | ... |
| VERIFY | ... | ... |
| **1.2-Preparation of information actions:** | | |
| WARN | ****** | |
| ASK FOR | ... | authorization to use the CTS |
| UPDATE | Vocal server | |
| PREPARE | Information Telexes | ... |
| ⋮ | ⋮ | ⋮ |
| **2-Action phase 1 (20 minutes after the detection of the incident) :** | | |
| **2.1-Information actions:** | | |
| SEND | Information Telex ****** | |
| ⋮ | ⋮ | ⋮ |
| **2.1-Network modification actions:** | | |
| CONTACT BY PHONE | to bring measurements into operation of *** for the CAA | to the traffic correspondents of the *** concerned |
| SEND | ... | ... |
| ⋮ | ⋮ | ⋮ |

Figure 3: Extract of the *CTZP duplex stopping* form (translated from French)

### 2.2 Knowledge in the application

Figure 2 shows the knowledge bases used in the application.

- **The index base** is a terminology of concepts (T-box part of our description logic). This base includes *incident concepts* such as *minor-incident*, *serious-incident*, *major-incident*, *simplex-stopping*, *duplex-stopping*, *ineffective-calls-rush*, *measurement-miss*, *overload*, etc. The sub-concepts are distinguished mainly by the kind of equipment which is involved in the incident. These conceptual definitions have been elaborated from documentation supplied by our industrial partner (CNET) and from a study on a supervision site (Perron, 1995; Nobecourt, 1995). The terminology has been validated by an expert of the site (Biébow et al., 1994). Moreover this base can evolve dynamically during learning cycles (Ventos et al., 1995b)[3]. Each incident concept of the index base is linked to its action form (3) and to the incidents (2) of the incident base which are instances of the incident concept.
- **The action form base** is a set of forms which describe certain actions the operator has to perform for each category of incidents. There is one action form for each incident concept of the index base. Figure 3 is an extract[4] of the *CTZP* duplex stopping[5] action form. At the moment there are about thirty incident forms[6].
- **The incident form base** is a set of forms, each of which contains all the information relative to an incident: date, actions performed, duration, consequences, causes, etc. There are approximately one hundred incidents a month. Figure 4 is an extract of an incident form.
- **The incident description base (A-box)** is a set of entries, each of them corresponds to an incident. It contains:
  - the properties of the incident in the description logic (it is a codification of a subset of the properties described in the incident form[7]),
  - a link (4) with the corresponding incident form,
  - a link (1) with the most specific incident concepts of the index base of which the incident is an instance (inverse link of (2)).

### 2.3 Benefits of the application

The objectives of this application have been dictated by several problems and insufficiencies in the present supervision tasks:

3 The learning process is beyond the scope of this paper.

4 CTZP is for Commutateur de Transit de Zone Périphérique *Transit Switch of Peripheral Zone*.

5 Stars (*) are put in place of certain confidential information.

6 This number is a rough indication. Indeed one of the main objectives of this project is to restructure all the knowledge concerning telecommunication supervision. A team from INRIA is working with the experts on a supervision site (Perron, 1995; Caulier and Perron, 1995) and a team from LIPN is studying the documentation (Nobecourt, 1995). The number of action forms is expected to increase but should not exceed fifty.

7 These properties correspond to those which have been used for the concept definitions of the index base.

| | |
|---|---|
| DR : TRAFFIC | CPE : CPRR ******* MAJOR INCIDENT |
| ******* CTU3 | 18 NOV 94 06:28 Duration : 21mn |
| EVENT: | Duplex stopping of ******* CTU3. |
| DETECTION: | ineffective beams:<br>RU**-RP**<br>RU**-RP**<br>totally busy beams:<br>RU**-RP**<br>All the overflowing beams are ineffective |
| SETIF ACTION: | action form: 14<br>Duty operator ***** called, comes, sees nothing abnormal.<br>The number of communications in progress is less than the number given by *****.<br>Search in the data base ****** : polluting telephone number ** ** ** **.<br>Call information for identification: not assigned.<br>... |
| CAUSES | polluting number ** ** ** **. |
| TRAFFIC IMPACT | 0 denials 36069 ineffective calls |
| COMMENTS | The numbers ** ** ** ** and ** ** ** ** correspond to two numéris groups by the name of ********.<br>The customer will be contacted.<br>.... |

Figure 4: Extract of an incident form

- The supervision and network equipment become more and more complex and varied therefore the operators are snowed under a stream of information to manage.
- The incident and the action forms are hand written and the operators spend a lot of time to find and consult the documents.
- The evolution of the equipment induces new classes of incidents which require new action forms to be elaborated. This work is done by hand studying incident descriptions.
- The supervision employees often changed and the knowledge of the experimented operators is not capitalized in order to be exploitable efficiently by the novice operators.
- An incident which is not quickly resolved can affect a lot of telephone subscribers and induce a considerable losses of money for the company.

The application allows to reduce these problems by:

- Using a knowledge based system to capitalize the experience of the operators.
- Quickly proposing a list of actions to be performed in order to resolve the incident and thus avoid a great number of suscribers to be affected.
- Relieving the operators by assisting them in the diagnosis task and automatically retrieving information about old similar incidents.
- Being a formative tool for novice operators.
- Assisting the operator to enrich the index base. The learning process automatically propose new incident concepts creation from incident descriptions.

### 2.4 Default knowledge in the application

From initial investigations on human supervisor expertise, it has been observed that end-users essentially manipulate concepts with default knowledge (Grelet, Marcerou, Pasdeloup, Périchon, and Ratel, 1992). however these defaults often contribute to the definition of the concepts. We then define a description language with two connectives to describe *default* and *exception* knowledge which allows such definitions. In the next subsections we give the intuition of our point of view for default knowledge and the underlying formalism. The benefits of the ability to describe *default* and *exception* knowledge for the application are:

- An adaptability in the description of the knowledge base since finding concept definitions with exclusively strict knowledge is not obvious and sometimes impossible.
- It is closed to the knowledge definition of human supervisors.
- It allows *definitions* (necessary and sufficient conditions to recognize an instance) for concept which otherwise would only be partially defined (necessary but not sufficient conditions). This is an indispensable feature for incident diagnosis where most of the concepts need to be defined for automatic classification.
- Default and exception knowledge are the foundations for the definition of the preference criteria and the partial matching in the similar cases retrieval.

## 3. Defaults in concept definitions

### 3.1 Introduction

In (classical) description logics concept definitions are exclusively made of *strict* knowledge. A conceptual definition is a set of necessary and sufficient conditions to recognize an instance: an object $o$ is an instance of a defined concept $C$ if it satisfies the definition of $C$ and all instances of $C$ necessarily satisfy this definition. However, much research (Doyle and Patil, 1991; MacGregor, 1991)(Padgham and Zhang, 1993, page 666) has shown that few concepts are defined with strict knowledge. It is a real problem since the partial definition of these concepts is not sufficient to recognize its instances and the concept can not be classified in the terminology. The same problem arose when creating the index base in our application, but it was possible to find a necessary and sufficient definition for these concepts by integrating default knowledge. For example, an incident which lasts more than 10 minutes and which affects at least 200 people must be recognized as a serious incident but some serious incidents which last more than 10 minutes but which are known to be exceptional relative to *at least 200 people* knowledge may not affect at least 200 people. We therefore extended a classic description logic with two new unary connectives *default* ($\delta$) and *exception* ($\epsilon$) in such a way that they can be part of concept definitions; we call it the *definitional point of view* of default knowledge. From this point of view a concept definition $C$ (including default knowledge) is still both *necessary* and *sufficient* to recognize an instance (constitutive property): *an object is an instance of a concept C iff it satisfies the strict definitional concepts of C, and satisfies or is explicitly "exceptional" w.r.t. the default concepts of C.* For example, the *serious-incident* concept can be defined as incident which lasts more than 10 minutes and which by default affects at least 200 people. Thus all serious incidents *necessarily* last more than 10 minutes and affect at least 200 people or are explicitly exceptional relative to *at least 200 people* knowledge, and conversely all incidents which last more than 10 minutes and which affect at least 200 people or which are exceptional relative to this default knowledge are recognized as serious incidents. Note that an exception is not a negation. Nothing can be said about the gray color of instances of a concept which definition is *exception to gray.* What an explicit exception does mean is that to be gray is no longer a characteristic of these instances (i.e. it can not be inferred they are gray or not gray but it is known that they are exceptional relative to gray characteristic). When there is an amalgam between negation and exception all the classical problems raised by Brachman appear. For example, yellow birds should be recognized as elephants as they are not gray and have no trunk and no tusks and all these properties are by default for the elephants (cf. (Brachman, 1985) for these examples).

### 3.2 Subsumption and inheritance: the intuition

Let us explain how our point of view on default knowledge is compatible with subsumption and classification of *standard* description logics. In *standard* description logics a concept $A$ is subsumed by a concept $B$ iff the set of instances of $A$ is a subset of the set of the instances of $B$ for all interpretations of $A$ and $B$. Our purpose was to be able to define concepts with default and exception knowledge and still remain compatible with standard semantics of subsumption and consequently with standard classification process. From the classical (default inference) point of view for default knowledge $A$'s are by default $B$'s means that "in general" $A$'s are $B$'s but there exist some $A$'s that are exceptional and that may be not $B$'s. Such a point of view for default knowledge alone is incompatible with standard subsumption and classification since the classifier can not handle probable knowledge about a concept to classify it in a terminology (e.g. it can not classify $A$ under $B$ if there exist $A$'s which are not $B$'s). We then decided to separate what is true for all interpretations (i.e. sure) about default and exception knowledge from what is probable (not true in all interpretations) and associate them two levels: the *standard subsumption and classification* for the former and *inheritance* for the latter. The definitional point of view with its constitutive property defined in the previous subsection induces *sure* subsumptions: a 'by default concept $C$' ($\delta C$) is a set of instances which includes instances of $C$ and instances "exceptional" w.r.t. $C$ (e.g. $C^\epsilon$, $C^{\epsilon^\epsilon}$, $\delta(C^\epsilon)$, $\delta(C^{\epsilon^\epsilon})$, $C^{\epsilon^{\epsilon^\epsilon}}$, etc. (cf. section 3.4)). It means that for each concept $C$, $C^\epsilon$, $C^{\epsilon^\epsilon}$, $\delta(C^\epsilon)$, $\delta(C^{\epsilon^\epsilon})$, $C^{\epsilon^{\epsilon^\epsilon}}$, etc. as well as $C$ itself are subsumed by $\delta C$. Based on these (sure) subsumptions the standard classification process can be applied to insert a new defined concept in a terminology. Whereas these subsumptions are true for all interpretations, inheritance level consists in complementing subsumption level by interpreting default in a classical (default inference) way where an exception inhibits a default. For example, if in the subsumption level it is known that $A$ is subsumed by $\delta C$ (i.e. instances of $A$ are instances of $\delta C$) but $A$ is not subsumed by an "exception" to $C$ (e.g. $C^\epsilon$) then the inheritance level will infer that $A$ is in *probable* inheritance relation with $C$ (i.e. in general instances of $A$ are instances of $C$). On the contrary if $A$ is subsumed by $C^\epsilon$ then the inheritance level can not infer that $A$ is in *probable* inheritance relation with $C$ (it is a classical default inference inhibited by an exception). Moreover for example if a concept $B$ is subsumed by $\delta C$ and by $\delta C^\epsilon$ and by $C^{\epsilon^\epsilon}$ then the inheritance level considers that $C$ is "counter-excepted" for $B$ therefore $B$ is in probable inheritance relation with $C$. Subsumption and inheritance are thus two complementary levels which match the classical default inferences but the advantage of our approach is the ability to classify a concept definition of which includes default and exception knowledge.

Formally we distinguish different *inheritance* relations between concepts (*sure*, *probable*, *typical*, *exceptional*), sure inheritance being subsumption exactly, probable inheritance relation matching the standard notion of default inference. Among sure and probable inheritance relations we showed that it is possible to distinguish *typical* and *exceptional* inheritance. $C$ is in typical inheritance relation with $D$ iff $C$ inherits all concepts inherited by $D$. $C$ is in exceptional inheritance relation with $D$ if $C$ does not inherit at least one concept which is inherited by $D$ (i.e. the inheritance is inhibited by an exception for $C$).

Let us show the relations between classical default inference point of view and these two levels (subsumption, inheritance) with an example (figure 5). Let the graph (G1) in figure 5 be a classical inheritance graph where a thin arrow is a *default link*, a heavy arrow is a *strict link* and a dashed arrow is an *exception link*. The relation between concepts is an inheritance relation and an exception inhibits the inheritance of a concept. Thus $D$ inherits $E$, $C$ inherits $D$ but $C$ does not inherit $E$, $F$ inherits $C$, $D$, $H$, $G$ but does not inherit $E$. Let us suppose now the introduction of the default and exception unary connectives (i.e. $\delta$ and $\epsilon$) to describe 'by default concept' and an 'exception to a concept'. The previous graph can thus be rewritten in a graph where all default links have been changed in strict links to the concepts 'by default concepts' and all exception links in strict links to concepts 'exception to

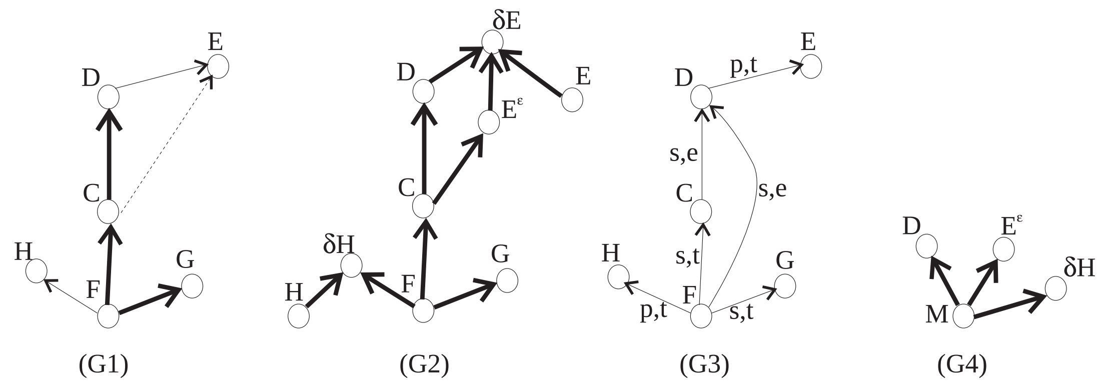


Figure 5: Default inference/subsumption and inheritance.

a concept' (figure 5 (G2)). In fact a terminology of concepts in our description logic can be viewed as an inheritance graph where only strict links remain. In such an inheritance graph inheritance relations are sure. This relation is exactly the subsumption relation. However some information in the first graph seem to be absent in the second one: *$D$ and $C$ are subsumed by $\delta E$ but what about $D$ inherits $E$ whereas $C$ does not?* The inheritance level allows to compute inheritance relations between concepts based on subsumption relation. For examples, $C$ is in a sure and exceptional inheritance relation with $D$ because $C$ is subsumed by $D$ but $D$ inherits $E$ whereas $C$ does not inherit; $F$ is in sure and typical inheritance relation with $C$ because $F$ is subsumed by $C$ and all concepts inherited by $C$ are inherited by $F$. In the graph (G3) several inheritance relations are given ($s$ stands for *sure*, $p$ for *probable*, $t$ for *typical* and $e$ for *exceptional*). The main consequence of the introduction of the $\delta$ and $\epsilon$ unary connectives and the subsumption level is the ability to classify concepts, definitions of which include default knowledge (i.e. 'by default concepts') and exceptions. Let us suppose concepts $D$, $C$, $F$ in the figure 5 to be (fully) defined (i.e. their definitions are necessary and sufficient), let $M$ (figure 5 (G4)) be a new defined concept, it will be classified under $C$, and above $F$.

Subsumption and inheritance are finally the base for instance checking (section 3.4.6.). Instance checking consists in finding all concepts of the terminology whose an object (i.e. an individual) is an instance of. Put simply, a conceptual abstraction $O$ of an object $o$ is elaborated from the description of the object $o$ and classified in the terminology (subsumption level). Then all concepts whose $o$ is an instance of are exactly all concepts that $O$ inherits (inheritance level). Thus $o$ is a sure/probable and a typical/exceptional instance of a concept $C$ iff $O$ is in a sure/probable and typical/exceptional inheritance with $C$.

Let us finally remark that the originality of this work is the subsumption level with classification of concepts defined with default knowledge. The inheritance level with the sure/probable and typical/exceptional inheritance relations are very useful for our CBR system (cf. section 4), but it is not in the tradition of non-monotonic inheritance work which essentially deal with ambiguity problems by searching criteria (e.g. specificity) to "resolve" them. That explains why we do not compare our work concerning inheritance with previous work. However we will see in section 3.4.1. that developments are currently being studied to extend this work in order to take into account specificity criteria.

### 3.3 Description logic

The description language $\mathcal{ALN}_{\delta\epsilon}$ is inductively defined from a set **R** of primitive roles, a set **P** of primitive concepts (primitive components in (Nebel, 1990)) and the constant concept $\top$ (top) with the syntax rule:

| | | |
|---|---|---|
| $C, D \rightarrow$ | $\top$ | the most general concept |
| | $\mid P$ | primitive concept |
| | $\mid C \sqcap D$ | concept conjunction |
| | $\mid \forall R : C$ | value restriction |
| | $\mid R$ AT-LEAST n | cardinality restriction on R (minimum) |
| | $\mid R$ AT-MOST n | cardinality restriction on R (maximum) |
| | $\mid \neg P$ | negation of a primitive concept |
| | $\mid \delta C$ | default concept |
| | $\mid C^{\epsilon}$ | exception to the concept $C$ |

**Example 1**
*ineffective-beam AT-LEAST 2* $\sqcap$ *switch*, describes all the switches which have at least two ineffective beams,
$\forall$*working-state:alarm* $\sqcap$ *equipment*, describes all the equipments where the working state is alarm,
$\forall$*linked-to:*$\neg$*CT-Paris* $\sqcap$ *switch*, describes all the switches which are not linked to a CT-Paris,

A (full) definition of an atomic concept defined from a term $C$ of $\mathcal{ALN}_{\delta\epsilon}$ is noted $A \equiv C$, a partial definition[8] is noted $A \sqsubseteq C$. In addition, $o::C$ expresses the fact that $o$ is an instance of the concept $C$ and $o_1::R:o_2$ the fact that $o_1$ and $o_2$ stand in the relation $R$.

**Example 2** [9]
The definitions of four incident concepts (i.e. they are subsumed by the concept *incident*) and the partial definition of another concept are given.

| | |
|---|---|
| *serious-incident* | $\equiv$ *incident* $\sqcap$ *more-than-10-minutes* $\sqcap$ $\delta$*at-least-200-affected-people* |
| *serious-service-incident* | $\equiv$ *incident* $\sqcap$ *service-number-affected* $\sqcap$ *more-than-10-minutes*, |
| *equipment-incident* | $\equiv$ *equipment-dependent* $\sqcap$ *incident*, |
| *serious-service-equipment-incident* | $\equiv$ *incident* $\sqcap$ *service-number-affected* $\sqcap$ *more-than-10-minutes* $\sqcap$ *equipment-incident* |

8 Note that partial definitions can be converted into full definitions by using new primitive concepts. Let an atomic concept $A$ be partially defined w.r.t. a term $C$, and $A'$ a new primitive concept the partial definition can be replaced with an equivalent full definition: $A \equiv A' \sqcap C$.

9 The real definitions are rather more complicated but for the sake of clarity and concision have been simplified in this paper.

| | |
|---|---|
| *service-number-affected* | $\sqsubset$ *number-overcall* |
| | $\sqcap$ *number-affected AT-LEAST 1* |
| | $\sqcap$ $\forall$*number: service* |
| | $\sqcap$ *at-least-200-affected-people*$^{\epsilon}$ |

### 3.4 Formal semantics

In (Coupey and Fouqueré, 1994a, 1994b) the $\mathcal{AL}_{\delta\epsilon}$ language (which includes $\delta$, $\epsilon$, $\forall$, $\neg$ and $\sqcap$ connectives) is described and we detail the framework in which subsumption and inheritance are defined. In order to relate formally a concept viewed as a set of instances and its algorithmic use (normalization and comparison, e.g. (Nebel, 1990)), we adopt an algebraic process. We give an equational system outlining the properties of the connectives. Thanks to the existence of an initial algebra w.r.t. this equational system (the descriptive semantics), we show that the various semantic viewpoints (i.e. algebra) coincide exactly with the equational one. The extensional semantics is the (traditional) point of view used in DLs: a concept is a set of instances. In the structural semantics, concepts are denoted exactly by their properties (normal form). Two concepts are equivalent if they have the same normal form. This semantics is operational in that it corresponds to an algorithm for subsumption. In the following subsections a brief and non technical presentation of this formal semantics is done in order to give to the reader a precise idea of subsumption and inheritance relations for the interpretation of the relations (defined in section 4) used in our CBR system.

**Remark 1** In (Coupey and Fouqueré, 1994b) it is shown that subsumption and inheritance relations in $\mathcal{AL}_{\delta\epsilon}$ are well-founded, complete, and the computations are performed in a polynomial time. Since handling the *AT-LEAST* and *AT-MOST* connectives does not increase the complexity (the computation of subsumption in $\mathcal{ALN}$ is polynomial (Donini et al., 1991a, 1991b) and there is no equation linking these connectives to $\delta$ and $\epsilon$ (cf. next subsection)) all these results and the formal semantics can obviously be extended to $\mathcal{ALN}_{\delta\epsilon}$ (cf. (Ventos et al., 1995a)).

#### 3.4.1 The equational system

The equational system gives the properties of the connectives. Thus the specificity relations between $\delta C$, $C^{\epsilon}$, $C$, $\delta\delta C$, $C^{\epsilon^{\epsilon}}$, etc. induced by the definitional point of view have been made explicit:

$\forall A, B, C \in \mathcal{ALN}_{\delta\epsilon}$:

| | | | |
|---|---|---|---|
| 1. | • $\sqcap$: | | $(A \sqcap B) \sqcap C = A \sqcap (B \sqcap C)$ |
| 2. | | | $A \sqcap B = B \sqcap A$ |
| 3. | | | $A \sqcap A = A$ |
| 4. | | | $\top \sqcap A = A$ |
| 5. | • $\forall$: | | $\forall R : (A \sqcap B) = (\forall R : A) \sqcap (\forall R : B)$ |
| 6. | | | $\forall R : \top = \top$ |
| 7. | • $\epsilon$: | $(\epsilon 1)$ | $(\delta A)^{\epsilon} = A^{\epsilon}$ |
| 8. | • $\delta$: | $(\delta 1)$ | $\delta(A \sqcap B) = (\delta A) \sqcap (\delta B)$ |

| | | | |
|---|---|---|---|
| 9. | | ($\delta 2$) | $A \sqcap \delta A = A$ |
| 10. | | ($\delta 3$) | $A^{\epsilon} \sqcap \delta A = A^{\epsilon}$ |
| 11. | | ($\delta 4$) | $\delta\delta A = \delta A$ |
| 12. | • AT-LEAST: | | $R$ AT-LEAST m$\sqcap R$ AT-LEAST n = $R$ AT-LEAST maxi(m,n) |
| 13. | | | $R$ AT-LEAST 0 = $\top$ |
| 14. | • AT-MOST: | | $R$ AT-MOST m$\sqcap R$ AT-MOST n = $R$ AT-MOST mini(m,n) |

Except those which are relative to $\delta$ and $\epsilon$, these equations are classical (Dionne et al., 1993a, 1993b). ($\delta 2$) and ($\delta 3$) are the two most important equations. They express a subsumption relation between $A$ and $\delta A$ ($A$ is subsumed by $\delta A$)[10] and between $A^{\epsilon}$ and $\delta A$ ($A^{\epsilon}$ is subsumed by $\delta A$). From a structural point of view this shows that if a concept contains $\delta A$ in its definition then it subsumes all the concepts which contain $\delta A$ or $A^{\epsilon}$ or $A$ in their definitions which corresponds to the definitional point of view for default given in the introduction to this section. From an extensional point of view the set of $\delta A$'s instances is seen as a superset of $A$'s instances and a superset of the instances which are exceptions to $A$ (i.e. to be an $A$ is more specific than to be a $\delta A$ and to be an $A^{\epsilon}$ is more specific than to be a $\delta A$). $\epsilon 1$ presupposes that an exception has a meaning only if it concerns a *default concept*. It is rather pragmatic since it considers that the end-users write $gray^{\epsilon}$ whereas they actually have $(\delta gray)^{\epsilon}$ in mind. Thanks to $\epsilon 1$ the two are equivalent. ($\delta 4$) allows redundant chains to be removed and ($\delta 1$) is a classical distributivity property. In fact these last two equations mean that certain criteria (e.g. specificity) used in path-based non-monotonic inheritance are not taken into account. One could, for instance, imagine a system where $\delta\delta C$ has a weakened meaning relative to $\delta C$ which would make $\delta 1$ obsolete. However let us recall that our main objective is to be able to define concept with default knowledge and classify them. The specificity criteria is useful to "resolve" inheritance ambiguities (e.g. a concept inherits by default a concept and its negation) but "ambiguities" do not reflect (up to now) real cases in our application. However developments are currently being studied to extend the equational system in order to take into account specificity criteria. Put simply, in this extension[11] $\delta$ is a *binary* connective linking a *"context"* (a concept) and the 'by default concept'. In case of ambiguity the most specific context will be preferred.

Certain equations linked to "inconsistency" (e.g. $A \sqcap \neg A$ is "inconsistent") seem to miss. In fact this is not the case. Let us suppose that we have the constant $\perp$ in our language denoting "inconsistency". First the absorption property of $\perp$ is undesirable in our framework. For instance, $\delta C \sqcap \delta(\neg C) \sqcap \delta(C^{\epsilon}) \sqcap \delta D$ would be equivalent to $\delta\ \perp$ (equation 8 and absorption property of $\perp$) which is really counter-intuitive as in particular $\delta D$ has been absorbed. Second our equational system allows *intentional* subsumptions between concepts that are equivalent to $\perp$ from an extensional point of view (i.e. $\emptyset$) to be detected (e.g. a *triangle which has four sides* is intentionally different from a *square circle* even if their extension is equal to the empty set (cf. (Woods, 1991) for a discussion about this subject)). However (section 3.4) inconsistencies (and ambiguities) are detected in the inheritance level.

The non advisability of some other equations are discussed in (Coupey and Fouqueré, 1994b).

10 Note that subsumption can be defined with equality: $A$ is subsumed by $B$ iff $A \sqcap B = A$.

11 This work is beyond the scope of this paper.

**Example 3 : Equality examples**

$\delta(\forall R : (A \sqcap B)) \sqcap (\forall R : A)^\epsilon$
$=$(eq 8) $\delta(\forall R : A \sqcap \forall R : B) \sqcap (\forall R : A)^\epsilon$
$=$(eq 5) $\delta(\forall R : A) \sqcap \delta(\forall R : B) \sqcap (\forall R : A)^\epsilon$
$=$(eq 10) $\delta(\forall R : B) \sqcap (\forall R : A)^\epsilon$

$\delta(R$ AT-LEAST 3$) \sqcap (R$ AT-LEAST 4$)^\epsilon$
$=$(eq 10) $\delta(R$ AT-LEAST 3$) \sqcap (R$ AT-LEAST 4$)^\epsilon \sqcap \delta(R$ AT-LEAST 4$)$
$=$(eq 8) $\delta(R$ AT-LEAST 3 $\sqcap$ $R$ AT-LEAST 4$) \sqcap (R$ AT-LEAST 4$)^\epsilon \sqcap$ $\delta(R$ AT-LEAST 4$)$
$=$(eq 12) $(R$ AT-LEAST 4$)^\epsilon \sqcap \delta(R$ AT-LEAST 4$)$
$=$(eq 10) $(R$ AT-LEAST 4$)^\epsilon$

$\delta(A) \sqcap (A \sqcap B)^\epsilon$
$=$(eq 10) $\delta A \sqcap (A \sqcap B)^\epsilon \sqcap \delta(A \sqcap B)$
$=$(eq 8) $\delta A \sqcap (A \sqcap B)^\epsilon \sqcap \delta A \sqcap \delta B$
$=$(eq 3) $(A \sqcap B)^\epsilon \sqcap \delta A \sqcap \delta B$
$=$(eq 8) $(A \sqcap B)^\epsilon \sqcap \delta(A \sqcap B)$
$=$(eq 10) $(A \sqcap B)^\epsilon$

$\forall R : \delta A \sqcap \forall R : \delta(A^\epsilon) \sqcap \forall R : A^{\epsilon^\epsilon}$
$=$(eq 5) $\forall R : (\delta A \sqcap \delta(A^\epsilon) \sqcap A^{\epsilon^\epsilon})$
$=$(eq 10) $\forall R : A^{\epsilon^\epsilon}$

**Remark 2** The equational system and these examples above show that equations on $\delta$ and $\epsilon$ work in any case where they are applied (i.e. $\delta(\forall R : C)$, $\delta$(AT-LEAST N R), $\delta$ (AT-MOST N R) and $\forall R : \delta C$ connectives). It should be noted that our language does not include connectives to construct roles (e.g. range, domain, AND-ROLE) therefore $\delta$ and $\epsilon$ are always applied to concepts (whatever they are).

#### 3.4.2 Descriptive subsumption

Following Birkhoff's theorem (cf. (Gratzer, 1968; Jacobson, 1989) for a presentation of universal algebras and sets of equations), this set of equations induces an equational class of algebras that we call $C_{\delta\epsilon}$-algebras. Connectives and constants of $\mathcal{ALN}_{\delta\epsilon}$ are interpreted as operations and elements in algebras of this class, thus giving the "meaning" of terms. Given a set $X$ of variables $\{x_1, \ldots, x_n\}$, we denote $\mathcal{ALN}_{\delta\epsilon}[X]$ the set of terms over $X$ and the signature of $\mathcal{ALN}_{\delta\epsilon}$. The quotient $\mathcal{ALN}_{\delta\epsilon}[X]_{|Eq}$ is a *free* $C_{\delta\epsilon}$-algebra, that is to say the only equalities valid in $\mathcal{ALN}_{\delta\epsilon}[X]_{|Eq}$ are those valid in all $C_{\delta\epsilon}$-algebras. $\mathcal{ALN}_{\delta\epsilon}[\emptyset]_{|Eq}$ is the initial algebra of the set of $C_{\delta\epsilon}$-algebras: for any $C_{\delta\epsilon}$-algebra $\mathcal{A}$, there exists a unique homomorphism from $\mathcal{ALN}_{\delta\epsilon}[\emptyset]_{|Eq}$ into $\mathcal{A}$.

Let us go back to the description knowledge base: it is a set of definitions $\Delta = \{x_1 \equiv C_1, \ldots, x_n \equiv C_n\}$ where $C_i \in \mathcal{ALN}_{\delta\epsilon}[\emptyset]_{|Eq}$[12]. We next consider the least congruence $\equiv_\Delta$

12 In this paper we ignore cyclic definitions.

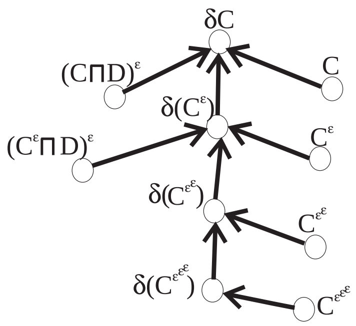


Figure 6: Examples of descriptive subsumption.

on $\mathcal{ALN}_{\delta\epsilon}[\emptyset]_{|Eq}$ satisfying $\Delta$. The quotient algebra, noted $\mathcal{D}_\Delta[X]$, is called the *descriptive algebra* as it provides a description of the classes of terms w.r.t. the knowledge base. Note that the (unique) homomorphism from the initial algebra $\mathcal{ALN}_{\delta\epsilon}[\emptyset]_{|Eq}$ into $\mathcal{D}_\Delta[X]$ is one-to-one.

**Definition 1** Let $A$,$B$ be elements of $\mathcal{ALN}_{\delta\epsilon}$, $B \sqsubset_d A$, i.e. *A descriptively* subsumes $B$, iff $B \sqcap A =_{\mathcal{D}_\Delta[X]} B$.

Since equations valid in this algebra are the only ones valid in *all* $\mathrm{C}_{\delta\epsilon}$-algebras, it captures the properties of concepts.

The figure 6 shows several concepts which are descriptively subsumed by $\delta C$.

### 3.4.3 Structural subsumption

Structural concept algebra is used to give an intentional semantics. It also captures the way subsumption testing is computed in implementations. A denotation (normal form) is computed for each concept defined by the end-user: this is the fundamental data structure handled by the subsumption algorithm. Our algebra distinguishes between primitive concepts, negation of primitive concepts, exception properties and role properties, either strictly or by default: an element of the domain is a pair *strict part, default part* of 4-uples *primitive part, negation part, role part, exception part*, each tuple itself being a set. The structural semantics of $\mathcal{AL}_{\delta\epsilon}$ is given in (Coupey and Fouqueré, 1994b, 1995). Each connective is interpreted as a semantic operation on elements of the domain. Testing structural subsumption is then based on the comparison of normal forms. It is shown in (Coupey and Fouqueré, 1994b) that structural subsumption is equivalent to descriptive subsumption for $\mathcal{AL}_{\delta\epsilon}$. It is useless in this paper to further develop the structural semantics of $\mathcal{ALN}_{\delta\epsilon}$ which is an obvious extension of $\mathcal{AL}_{\delta\epsilon}$ one's.

### 3.4.4 Extensional subsumption

In the literature, subsumption is generally defined from a model-theoretic point of view. The formal meaning of concept descriptions is classically (Nebel, 1990) given by an interpretation $I = (\mathbf{D}, \|.\|^I)$. $\mathbf{D}$ (the domain) is an arbitrary non-empty set of individuals and $\|.\|^I$ is an interpretation function such that every concept is mapped onto a subset of $\mathbf{D}$ and every role onto a subset of $\mathbf{D} \times \mathbf{D}$. Extensional $C_{\delta\epsilon}$-algebras and corresponding extensional subsumptions can be defined in the following way ($\epsilon$ and $\delta$ are interpreted as unary semantic functions from $\mathbf{D}$ to $\mathbf{D}$):

$$\begin{array}{rcl}
\|\top\|^I & = & \mathbf{D} \\
\|P\|^I & \subset & \mathbf{D}, \text{ where } P \text{ is a primitive concept} \\
\|C \sqcap D\|^I & = & \|C\|^I \cap \|D\|^I \\
\|\neg A\|^I & = & \mathbf{D} \backslash \|A\|^I \\
\|\delta C\|^I & = & \delta^I(\|C\|^I) \\
\|C^\epsilon\|^I & = & \epsilon^I(\|C\|^I) \\
\|\forall R : C\|^I & = & \{x \in \mathbf{D} / \forall y, \text{ if } (x,y) \in \|R\|^I \text{ then } y \in \|C\|^I\} \\
\|R \text{ AT-LEAST n}\|^I & = & \{x \in \mathbf{D} / \text{card}(\{y/(x,y) \in \|R\|^I\}) \geq n\} \text{ where card(.) is the} \\
& & \text{cardinality fonction} \\
\|R \text{ AT-MOST n}\|^I & = & \{x \in \mathbf{D} / \text{card}(\{y/(x,y) \in \|R\|^I\}) \leq n\}
\end{array}$$

such that all the interpretations of $\delta$ ($\delta^I$) and $\epsilon$ ($\epsilon^I$) respect the equational system. $\delta^I$ and $\epsilon^I$ are semantic functions which give a subset of the domain such that $\delta^I(\|C\|^I)$ is a set which includes $\|C\|^I$ and $\|C^\epsilon\|^I$. An interpretation $I$ is a model for a concept $C$ if $\|C\|^I$ is non-empty. Based on this semantics we give the following definition:

**Definition 2** Let $A$,$B$ be elements of $\mathcal{ALN}_{\delta\epsilon}$,$B \sqsubset_e A$, i.e. $B$ is extensionally subsumed by $A$ iff $\|B\|^I \subset \|A\|^I$ for each interpretation $I$.

**Theorem 1** Let $C$ and $D$ be elements of $\mathcal{AL}_{\delta\epsilon}$ such that, for each interpretation $I$, $\|C\|^I \neq \emptyset$ and $\|D\|^I \neq \emptyset$, $C \sqsubset_s D$ iff $C \sqsubset_e D$.

**Proof:** (sketch) As $\delta$ and $\epsilon$ satisfy the equational system, and the interpretation of the other connectives is standard, structural subsumption implies extensional subsumption. To prove the converse, note that an interpretation is made up of an interpretation of $\delta$, of $\epsilon$ and the remainder. The properties satisfied by $\delta$ and $\epsilon$ allow us to restrict our attention to the standard part. Because there is no bottom, we need the unemptiness condition on the denotations in order to avoid inconsistency (cf. the following remark). The proof is then similar to the classical one.

In fact the only differences between *extensional* and *structural* subsumption concern *"inconsistency"* (empty extension). Indeed, the structural algebra matches the intentional point of view since $P_1 \sqcap \neg P_1$ is structurally different from $P_2 \sqcap \neg P_2$ when they are both extensionally equivalent to $\emptyset$.

In order to illustrate the subsumption relation[13] let us look again at the definitions of the example in section 3.3. Figure 7 is a graphical representation of these definitions. A thin arrow represents a '*by default* concept', a heavy one a *strict* concept, a dashed one an *exception* to a concept, a square represents a role, and a circle a concept. A star near a concept means that the concept is primitive or partially defined (necessary but not sufficient conditions). A first set of obvious subsumptions is given by *strict* links and by the transitivity

13 In the remainder of the paper $\sqsubset$ will be used to denote a subsumption relation.

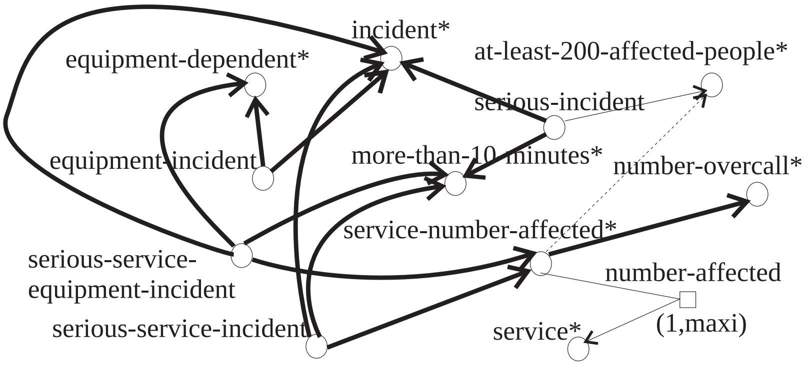


Figure 7: Concept terminology

of the subsumption relation. For example *serious-service-incident* is subsumed by *service-number-affected* and by *number-overcall*. Other subsumptions are shown by default links. Thus *serious-incident* is subsumed by $\delta$*at-least-200-affected-people* (but not by *at-least-200-affected-people*). In addition to these obvious subsumptions other subsumption relations can be computed. For example *serious-service-equipment-incident* is subsumed by *equipment-incident* and *serious-service-incident* because it contains in its definition all the properties of *equipment-incident* and *serious-service-incident*. There is also a subsumption between *serious-service-incident* and *serious-incident* because *serious-service-incident* contains in its definition all the properties of *serious-incident* except $\delta$*at-least-200-affected-people* but it contains an exception to this property (*at-least-200-affected-people*$^{\epsilon}$) and as *at-least-200-affected-people*$^{\epsilon}$ is subsumed by $\delta$*at-least-200-affected-people* ($\delta 3$ equation), *serious-service-incident* is subsumed by *serious-incident*. These implicit subsumptions are computed by the classifier which inserts a new defined concept $C$ "under" the most specific concepts it is subsumed by and "above" the most general concepts it subsumes. In figure 8 the concepts have been classified.

### 3.4.5 Inheritance relations

Subsumption relation allows in particular to know that a concept $D$ is more specific than a concept $\delta C$. However, for inheritance, it is necessary to detect that $D$ is subsumed or not by an exception to $C$ to know if $D$'s are probable instances or not of $C$. The inheritance matches the classical default inference point of view for default where an exception has an inhibitory effect on a default (i.e. a counter-exception $(C^{\epsilon^{\epsilon}})$ as a inhibitory effect on a default exception $(\delta(C^{\epsilon}))$. Thus if a concept $C$ is subsumed by $\delta A$ and by $A^{\epsilon}$ then $C$ does not inherit $A$, but if $C$ is subsumed by $\delta A$, $\delta(A^{\epsilon})$ and by $A^{\epsilon^{\epsilon}}$ then $C$ inherits $A$ (it is a counter-exception). We chose the normal-form of our structural subsumption such that the inheritance relation computations are very easy ((Coupey and Fouqueré, 1994a, 1994b)). In fact this normal form is the smallest one because it corresponds to the use of the equations

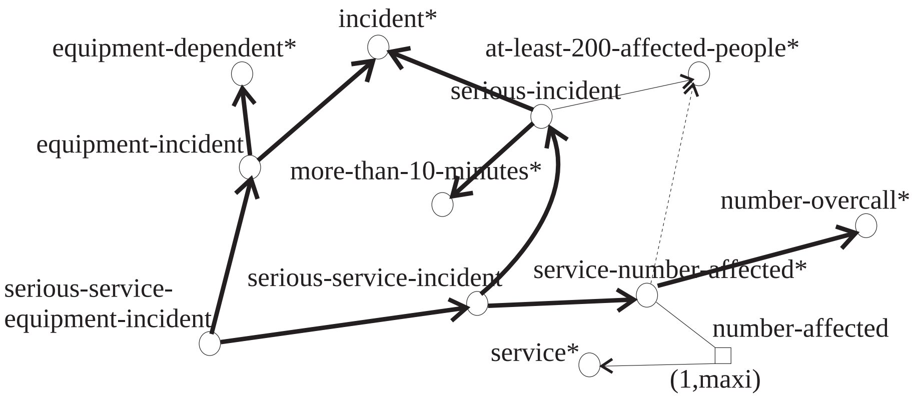


Figure 8: Classification

such that it reduces the computed form at each step (e.g. $\delta A \sqcap A^{\epsilon}$ "is rewritten in" $A^{\epsilon}$). Thus the resulting form does not contain all the concepts which have been excepted.

Let us simply consider that if the set of the most specific concepts that subsume $D$ include an exception to $C$ at an odd level (e.g. $C^{\epsilon}$, $C^{\epsilon^{\epsilon^{\epsilon}}}$, etc.) then $C$ is said to be "excepted" (i.e. the inheritance of $\delta C$ is inhibited). On the contrary if this set includes no exception to $C$ or an exception to $C$ at an even level (i.e. a counter-exception) then $C$ is said to be not excepted (i.e. the inheritance of $\delta C$ is not inhibited). Two inheritance relations (sure and probable) can be distinguished whose definitions are[14]:

**Definition 3**

- A concept $C$ **is in a sure inheritance relation** with a concept $D$ ($C \stackrel{s}{\to} D$) iff $C$ is subsumed by $D$ (it is said that $C$ *s-inherits* $D$).
- A concept $C$ **is in a probable inheritance relation** with a concept $D$ ($C \stackrel{p}{\to} D$) iff $C \stackrel{s}{\not\to} D$, $C \stackrel{s}{\not\to} \neg D$, $C$ is subsumed by $\delta D$ and neither $D$ nor a concept s-inherited by $D$ is excepted for $C$ (it is said that $C$ *p-inherits* $D$).

Moreover among these sure and probable inheritance relations it is possible to distinguish those which are *typical* from those which are *exceptional*:

- An inheritance is *typical* ($C \stackrel{t}{\to} D$) when all the concepts inherited by $D$ are also inherited by $C$.

14 cf. (Coupey and Fouqueré, 1994a, 1994b) for detailed definitions.

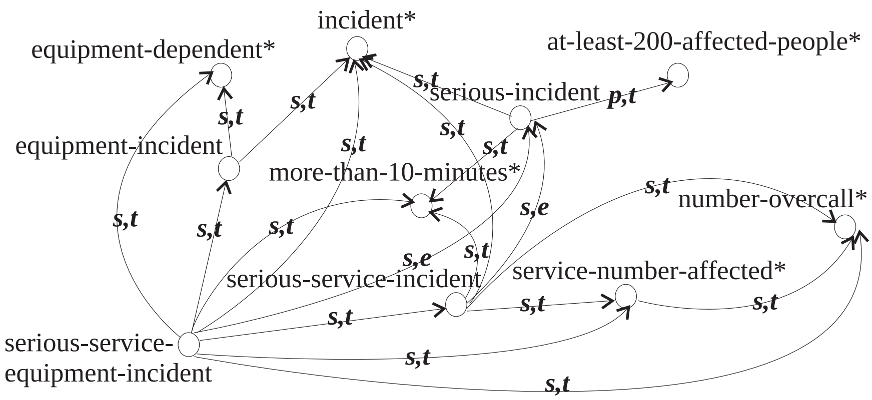


Figure 9: Inheritance relations

- An inheritance is *exceptional* ($C \stackrel{e}{\rightarrow} D$) when there exists at least one concept inherited by $D$ which is not inherited by $C$.

**Remark 3**

1. We say that $C$ inherits $D$ ($C \rightarrow D$) iff $C \stackrel{s}{\rightarrow} D$ or $C \stackrel{p}{\rightarrow} D$.

2. Ambiguities and inconsistencies are detected in our system. A concept $C$ is ambiguous iff it p-inherits both a primitive concept and its negation or a role of $C$ inherits an ambiguous value restriction. A concept $C$ is inconsistent iff it s-inherits both a primitive concept and its negation or a role of $C$ inherits an inconsistent value restriction.

Figure 9 shows the inheritance relations beween all concepts. For example *serious-service-incident* s-inherits *serious-incident* and this inheritance is exceptional because it does not inherit *at-least-200-affected-people*. *serious-service-equipment-incident* s-inherits *serious-service-incident* and this inheritance is typical because it inherits all concepts inherited by *serious-service-incident*.

### 3.4.6 Instance-concept relations

So far now we have seen relations between *concepts*, where the *instance-concept* relations correspond to the relations between objects (or individuals) and concepts. From the description of an object, the aim is to determine all the concepts of the terminology of which the object is an instance. In description logics without $\delta$ and $\epsilon$ connectives, this process is based on subsumption relation and can be globally defined in the following way:

- From the definition of the object $o$, an *abstract* concept[15] $O$ is created (cf. (Nebel, 1990) for a detailed description of abstraction) which is in fact the most specific

15 In the remainder of the paper we note $C$ the abstract concept of an object $c$.

concept which generalizes $o$. For example let the set of facts $\{$*o1 ::more-than-10-minutes*, *o1 ::number-affected: o2*, *o1 ::incident*$\}$ (i.e. $o1$ is an incident which lasts more than 10 minutes and which affects the number $o2$) be associated with $o1$ and the set of facts $\{$*o2 ::¬service*$\}$ (i.e. $o2$ is not a service number) be associated with $o2$; the definition of the abstract concept 01 is 01 $\equiv$ *more-than-10-minutes* $\sqcap$ $\forall$*number-affected:¬service*[16] $\sqcap$ *number-affected AT-LEAST 1* $\sqcap$ *number-affected AT-MOST 1*$\sqcap$ *incident.*

- $o$ is an instance of all the concepts which subsume 0.

In our description logic the connectives $\delta$ and $\epsilon$ allow the instance-concept relations to be distinguished (as for inheritance relations). An object can be a *sure* or a *probable* instance and an instance can be *typical* or *exceptional.* The definitions are:

**Definition 4**

- $o$ is a *sure instance* of a concept $C$ iff $0 \overset{s}{\rightarrow} C$,
- $o$ is a *probable instance* of a concept $C$ iff $0 \overset{p}{\rightarrow} C$,
- $o$ is a *typical instance* of a concept $C$ iff $0 \overset{t}{\rightarrow} C$,
- $o$ is an *exceptional instance* of a concept $C$ iff $0 \overset{e}{\rightarrow} C$,

**Example 4**

Let us give the following descriptions for $o2$ and $o3$ :

*o2 ::equipment-dependent*,
*o2 ::serious-incident*,
*o2 ::defective-equipment :o3*,
*o3 ::switch*

The abstract concept is defined as 02 $\equiv$ *serious-incident* $\sqcap$ *equipment-dependent* $\sqcap$ $\forall$*defective-equipment* :*switch.*

02 is therefore subsumed by:
*incident*
*serious-incident*
*more-than-10-minutes*
*equipment-dependent*
*equipment-incident*
$\delta$*at-least-200-affected-people.*

02 is in a sure and typical inheritance relation with:
*incident*
*serious-incident*

16 All the roles are closed when computing the abstract concept.

*more-than-10-minutes*
*equipment-dependent*
*equipment-incident*
$o2$ is therefore a sure and typical instance of these concepts.

02 is in a probable and typical inheritance relation with:
*at-least-200-affected-people*
$o2$ is therefore a probable and typical instance of this concept.

**Remark 4** If we add the fact *o2 ::service-number-affected*, $o2$ will no longer be an instance of *at-least-200-affected-people*.

## 4. The retrieval process

Our retrieval process is based on the classification process and the inheritance relations to retrieve similar incidents to a new incident and apply some preference criteria to perform a suitable display to the user. There are two levels in this process. The *conceptual level* consists in finding similar incident concepts of the index base in order to access their action forms which are displayed to the user. In the *instance level* old incident instances which are similar to the new incident are retrieved and the user can access their form. For each level (conceptual and instance), criteria which are totally independent of any application are firstly described and then it is shown how they are used in the application.

### 4.1 Presentation of the reasoning

Figure 10 shows the succession of steps required to go from case description $c_{new}$ to obtaining the set of similar old cases. Each step will be further developped. The scenario is as follows:

1. The abstract concept $C_{new}$ is created from the description of the new incident $c_{new}$ given by the operator, as described in the section 3.4.
2. All the (sure, probable, typical, exceptional) inheritance relations between $C_{new}$ and the concepts of the index base are then computed. This set of index concepts which are inherited by $C_{new}$ is the set of *similar* concepts to $C_{new}$.
3. Criteria are applied to these similar index concepts in order to discriminate and present them in a suitable display to the operator.
4. In order to find other index concepts, $C_{new}$ is refined by adding exceptions to its description (*partial matching*). This operation is an extrapolation of the new case that considers $C_{new}$ exceptional w.r.t. certain default knowledge, in order to find some other index concepts $C_{new}$ inherits.
5. From these similar index concepts:

    (a) Associated action forms are accessed (cf. figure 2).

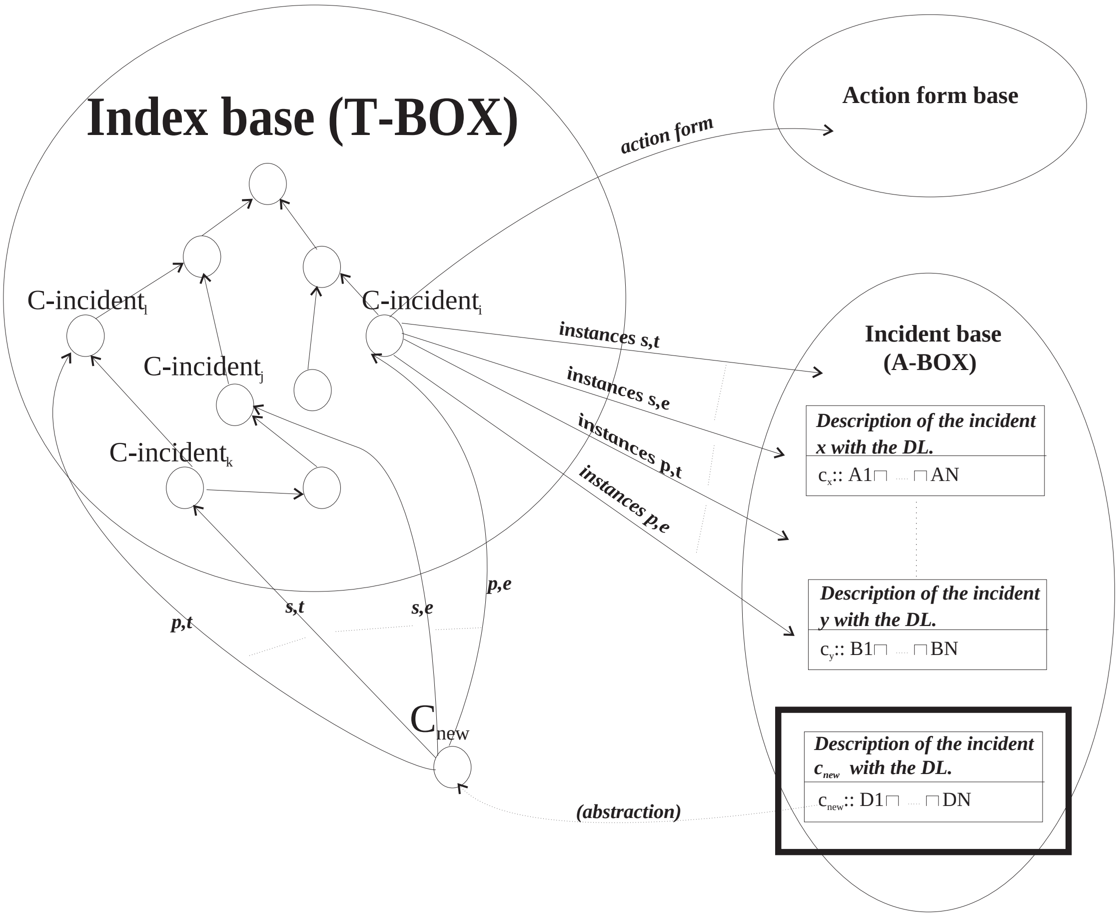


Figure 10: Processing a new incident

(b) Their instances are accessed. This link has already been described in figure 2. It symbolizes the instance-concept *specificity* relation defined in the previous section. The most representative instances of the new case are chosen. From these instances the incident forms to be presented to the operator are obtained (cf. figure 2).

6. Once the incident has been resolved, the operator makes an incident form for the new incident and this form is integrated in the incident base.

### 4.2 An example

The reader may find in the appendix a presentation of the implementation of our system. This presentation includes an example we formally describe here. The definitions[17] of six incident concepts (i.e. they are subsumed by *incident* concept) and the definitions of three other concepts are given:

$$
\begin{array}{lll}
\textit{serious-incident} & \equiv & \textit{incident} \sqcap \textit{more-than-10-minutes} \sqcap \\
 & & \delta \textit{at-least-200-people-affected}, \\
\textit{CTU-duplex-stopping} & \equiv & \textit{incident} \sqcap \textit{total-busy-beam AT-LEAST 1} \sqcap \\
 & & \textit{ineffective-beam AT-LEAST 2} \sqcap \textit{CTU-dependent} \\
 & & \sqcap\ \delta(\textit{ineffective-overflowing-beam AT-LEAST 1}), \\
\textit{CTU-EM-DS} & \equiv & \textit{CTU-duplex-stopping} \sqcap \textit{CTU-EM-dependent}, \\
\textit{equipment-incident} & \equiv & \textit{equipment-dependent} \sqcap \textit{incident}, \\
\textit{serious-CTU-EM-DS} & \equiv & \textit{CTU-EM-DS} \sqcap \textit{serious-incident}, \\
\textit{CTU-fire} & \equiv & \textit{fire-alarm} \sqcap \delta \textit{CTU-duplex-stopping}, \\
 & & \\
\textit{CTU-dependent} & \sqsubset & \delta \textit{equipment-dependent}, \\
\textit{CTU-EM-dependent} & \equiv & \textit{CTU-dependent} \sqcap \\
 & & (\textit{ineffective-overflowing-beam AT-LEAST 1})^{\epsilon}, \\
\textit{high-temperature} & \sqsubset & \delta \textit{fire-alarm},
\end{array}
$$

Let $c1_{new}$ be a new incident whose description is:

$$
\begin{array}{l}
c1_{new} :: \textit{incident}, \\
c1_{new} :: \textit{CTU-dependent}, \\
c1_{new} :: \textit{ineffective-beam AT-LEAST 3}, \\
c1_{new} :: \textit{total-busy-beam AT-LEAST 2}, \\
c1_{new} :: \textit{ineffective-overflowing-beam AT-MOST 0}, \\
c1_{new} :: \textit{more-than-10-minutes}, \\
c1_{new} :: \textit{at-least-200-affected-people}, \\
c1_{new} :: \textit{high-temperature}
\end{array}
$$

$c1_{new}$ is therefore an incident which depends on a CTU which has at least 3 ineffective beams, 2 totally busy beams, which does not have ineffective overflowing beams[18], which lasts more than 10 minutes, which affects at least 200 people and for which a high temperature is detected. Its abstract concept is:

17 CTU: urban transit switch, CTU-EM: electro-mechanical CTU and DP: duplex stopping.

18 Using $R$ AT-MOST 0 allows to state that a concept has no relation $R$.

$$
\begin{aligned}
C1_{new} \equiv\ & \textit{incident} \sqcap \textit{CTU-dependent} \\
& \sqcap \textit{ineffective-beam AT-LEAST 3} \\
& \sqcap \textit{total-busy-beam AT-LEAST 2} \\
& \sqcap \textit{ineffective-overflowing-beam AT-MOST 0} \\
& \sqcap \textit{more-than-10-minutes} \sqcap \textit{at-least-200-affected-people} \\
& \sqcap \textit{high-temperature.}
\end{aligned}
$$

Comparing the description of $C1_{new}$ and the description of the above concepts, some inheritance relations can be detected. For example[19]:
$C1_{new} \stackrel{s,t}{\rightarrow}$ *serious-incident*,
$C1_{new} \stackrel{p,t}{\rightarrow}$ *equipment-incident*,
but $C1_{new} \not\rightarrow$ *CTU-duplex-stopping*[20].

### 4.3 The conceptual level

The first phase of the conceptual level in our retrieval process consists in comparing the abstract concept of the new case (incident) with the index concepts of the index base to retrieve similar indexes to the new abstract concept incident. The inheritance relation allows all the index concepts which are inherited by the new incident concept to be retrieved. However, it is possible to discriminate between these indexes using criteria (specificity, sure, typical) formalized below.

The second phase is a direct application of exception and default knowledge in the description logic and involves considering the new case as a possible exceptional case w.r.t. certain default knowledge of the index concepts. It corresponds to the following question. *If I extrapolate from the new case and consider that it is exceptional w.r.t. certain default knowledge, could I find some other index concepts $C_{new}$ inherits?* The operation which adds exceptions to the initial description of a concept is called "concept refinement". Each "refined concept" may inherit index concepts that the initial description does not inherit. Then, a preference criterion (*exception criterion*) between refined concepts, based on the number of exceptions added, is defined. The objective of our concept refinement operation is similar to the weak classification of Koehler (cf. introduction) i.e. trying to find concept which are partially similar to the new case. However our partial matching is "semantically bounded" by the kind of knowledge since only defaults (and no strict knowledge) can be excepted therefore avoiding unjustified "similarity".

**Specificity criterion:** Let us first look at the formal definition of the *specificity* property. Among the set of concepts $\mathcal{S}$, we can distinguish concepts which have no inheritance relations with the other concepts. These concepts are in fact the leaves of the inheritance graph reduced to the set $\mathcal{S}$. They are called the *most specific* concepts.

**Definition 5** The function which selects the most specific concepts from a set of concepts is defined as follows:

---

19 $p$ is for *probable*, $s$ for *sure*, $t$ for *typical* and $e$ for *exceptional*.

20 $C1_{new}$ should be known to be exceptional w.r.t. *(ineffective-overflowing-beam AT-LEAST 1)* in order to have $C1_{new} \stackrel{p,e}{\rightarrow}$ *CTU-duplex-stopping*.

$$\begin{array}{lll} \mathcal{MS}: 2^{\mathcal{ALN}_{\delta\epsilon}} \rightarrow 2^{\mathcal{ALN}_{\delta\epsilon}} \\ \quad \mathcal{S} & \mapsto \{D_i \in \mathcal{S}, \forall D_j (j \neq i) \in \mathcal{S}, D_j \not\rightarrow D_i\} \end{array}$$

Let us define the following sequence $\mathcal{U}$:

$$\begin{array}{lll} \mathcal{U}: 2^{\mathcal{ALN}_{\delta\epsilon}} \rightarrow 2^{\mathcal{ALN}_{\delta\epsilon}}: \\ \quad \mathcal{U}^1(\mathcal{S}) & = & \mathcal{MS}(\mathcal{S}) \\ \quad \mathcal{U}^n(\mathcal{S}) & = & \mathcal{MS}(\mathcal{S} \backslash \bigcup_1^{n-1} \mathcal{U}^i(\mathcal{S})) \end{array}$$

Then the definition of the function $\mathcal{H}$ which gives the specificity value of a concept in a set $\mathcal{S}$ of concepts (i.e. a kind of height in the inheritance graph $\mathcal{S}$) is:

$$\begin{array}{ll} \mathcal{H}: \mathcal{ALN}_{\delta\epsilon} \times 2^{\mathcal{ALN}_{\delta\epsilon}} \rightarrow \mathbb{N}^*: \\ \quad (C, \mathcal{S}) & \mapsto i \text{ s.t. } C \in \mathcal{U}^i(\mathcal{S}) \end{array}$$

The more specific a concept that $C_{new}$ inherits is (i.e. lower its specificity value is) the more preferred it is relative to the *specificity criterion.* For example the most specific concepts that $C1_{new}$ inherits are *serious-incident* and *equipment-incident.*
The common-sense justification is that the more specific a concept that $C_{new}$ inherits is, the more properties it has in common with $C_{new}$.

**Sure criterion:** If $C_{new} \overset{s}{\rightarrow} C_{old}$ and $C_{new} \overset{p}{\rightarrow} C'_{old}$ then $C_{old}$ is preferred relatively to the *sure criterion.* The common-sense justification of this criterion is that what is *sure* is preferable to what is *probable.* For example, *serious-incident* is preferred to *equipment-incident* because $C1_{new} \overset{s}{\rightarrow}$*serious-incident*, whereas $C1_{new} \overset{p}{\rightarrow}$ *equipment-incident.*

**Typical criterion:** If $C_{new} \overset{t}{\rightarrow} C_{old}$ and $C_{new} \overset{e}{\rightarrow} C'_{old}$ then $C_{old}$ is preferred relative to the *typical criterion.* This is justified by the fact that "typical" means (in opposition to "exceptional") that $C_{new}$ inherits all the concepts that $C_{old}$ inherits (including the default ones).

**Exception criterion:** Let us begin by defining the notion of refinement of a concept $C$. Refining a concept $C$ consists in adding exceptions to its description.

A refined concept of a concept $C$ is a concept $C'^{ex}$ s.t. there exists a finite set of concepts $F_i$ and $C'^{ex} = C$

**Theorem 2** The refinement preserves the subsumption relation.
Let $C^{ex}$ be a refined concept of $C$, if $C \sqsubset D$ then $C^{ex} \sqsubset D$.

**Proof** The proof obviously follows from the monotonicity property of the subsumption relation. $\forall C, D, E \in \mathcal{ALN}_{\delta\epsilon}$, if $C \sqsubset D$ then $C \sqcap E \sqsubset D$ therefore in particular $C \sqcap \sqcap_i F_i^{\epsilon} \sqsubset D$

For example the refinement of $C1_{new}$, which consists in adding an exception to *ineffective-overflowing-beam AT-LEAST 1*, inherits *CTU-EM-dependent*, *CTU-EM-DS*, *serious-CTU-EM-DS* and *CTU-duplex-stopping.*

The *exception* criterion is then defined as follows:
Let $C^{ex}_{new_1} = C_{new} \sqcap \sqcap^{n}_{i=1} F^{\epsilon}_{i}$ and $C^{ex}_{new_2} = C_{new} \sqcap \sqcap^{m}_{j=1} F^{\epsilon}_{j}$ be two refined concepts of $C_{new}$ then $C^{ex}_{new_1}$ is preferred relatively to the exception criterion iff $n < m$.
In other words, a refined concept $C^{ex}_{new_1}$ is preferred to $C^{ex}_{new_2}$ iff it contains less exceptions than $C^{ex}_{new_2}$. The common-sense justification is that the greater the number of exceptions that have been added to the initial concept, the further away the refined concept is from the initial one.

The refinement operation and the exception criterion can be viewed as *partial matching operations* since the exceptions allow similar concepts to be found which are not similar to the initial description. Moreover our approach has several advantages:

- it is formalized,
- it is independent of the application and is sufficiently general to be used in other applications,
- the partial matching is *semantically bounded* by the kind of knowledge (only default knowledge can be excepted) therefore avoiding unjustified "similarity" and it can be easily explained to the end-user.

**Example 5** (continued): The two following tables summarize the incident concepts which are similar to $C1_{new}$ with results concerning the preference criteria. The first one concerns the initial description $C1_{new}$ and the second the refined concept $C1^{ex}_{new} = C1_{new}$ $\sqcap$ *ineffective-overflowing-beam AT-LEAST 1*$^{\epsilon}$. The first column is the concept name, the others are for criteria[21]:

| **concept name** | **typical** | **sure** | **specificity** |
|---|---|---|---|
| *serious-incident* | typical | sure | 1 |
| *equipment-incident* | typical | probable | 1 |

| **concept name** | **typical** | **sure** | **specificity** |
|---|---|---|---|
| *serious-incident* | typical | sure | 2 |
| *equipment-incident* | typical | probable | 1 |
| *CTU-EM-DS* | typical | sure | 2 |
| *serious-CTU-EM-DS* | typical | sure | 1 |
| *CTU-duplex-stopping* | exceptional | sure | 3 |
| *CTU-fire* | exceptional | probable | 1 |

**In our application**: Put simply, the order in which the above criteria are applied in our application corresponds to a pragmatic necessity of a suitable display for the operators. Thus it is considered that the most similar index incident concepts are those which are not refined and which are most specific (specificity criterion), typical (typical criterion), sure (sure criterion) (in this order). For the refined concepts, those which have the least exceptions (exception criterion) are chosen, then the same above order is applied. Although

21 Except the *exception* criterion since there is only one refined concept.

the first results obtained are relevant and useful in our application, it is easy to modify this order (for another application) thanks to the use of the symbolic criteria. This offers an advantage over numerical criteria, where a different order can only be obtained by modifying a complicated weighting system. Another advantage of our symbolic criteria is the ability to produce explanations to the end-user easily. Explanation such as *"This incident concept C is more similar than this other D because the new incident is a sure and typical instance of C ..."* can easily be understood and evaluated by the end-user.

Finally note that the refinement function is very important and useful in the application. Indeed it gives some probable extrapolations of the incident (a kind of reasonning with incomplete information) which can be very helpful to the operator. Indeed this information can alert the operator on an incident which seems at first sight minor but for which one extrapolation is very serious (as *CTU-fire* in the example). The operator can thus manage this incident sooner and get more priority treatment of it than if the system did not give him this information.

### 4.4 The instance level

Once the similar index concepts have been ordered as described above their action form and their stored instances are accessed. Put simply, the conceptual description of each index concept is the abstraction of a certain number of properties its instances have in common. However the instances have some discriminating properties[22]. Therefore, among these instances it is possible to distinguish some which are more similar than others to the new incident. For example an instance which is an instance of both *serious-incident* and *high-temperature* is more similar to $C1_{new}$ than an instance which is only an instance of *serious-incident*. To discover them two phases are used: a *terminology* with the abstract concepts of the old instances is created in order to find subsumption and inheritance relations between them and the properties that $c_{new}$ and the $c_{old_i}$ have in common are compared using the *least common subsumption* ($\mathcal{LCS}$) operation of description logic.

**Terminology of $C_{old_i}$:** A terminology of the abstract concepts $C_{old_i}$ of all the retrieved old incidents and $C_{new}$ is automatically[23] and temporarily created. The purpose of this creation is to discover subsumption and inheritance relations between the $C_{old_i}$ and to (be able to) apply conceptual preference (specificity, typical, sure) criteria defined in the conceptual level to select the most similar abstract instances.

However it may happen that $C_{new}$ does not inherit any abstract concepts of the old instances. One solution to discover ever so similarity relations between $c_{new}$ and the $c_{old_i}$ consists in constructing concepts whose description is the common concepts of the definitions of $C_{new}$ and each $C_{old_i}$. This operation is known as the $\mathcal{LCS}$ (least common subsumption) in description logic (Cohen and Hirsh, 1994; Ventos et al., 1995b). Then these $\mathcal{LCS}$ concepts are classified in the terminology of the abstract concepts $C_{old_i}$ and the (specificity, typical, sure) criteria can be applied to select the most similar instances.

22 These discriminating properties are indications for the learning process. The less these properties are discriminating (i.e. the greater the number of stored instances which have these properties in common) the more a learning process is necessary (this learning process is beyond the scope of this paper).

23 Remember that thanks to the classification process this terminology is really created *automatically.*

**Least common subsumption operation**:

$$\mathcal{LCS} : \mathcal{ALN}_{\delta\epsilon} \times \mathcal{ALN}_{\delta\epsilon} \rightarrow \mathcal{ALN}_{\delta\epsilon} :$$
$$(C1, C2) \quad \mapsto D \in \mathcal{ALN}_{\delta\epsilon}, C1 \sqsubset D \textit{ and } C2 \sqsubset D \textit{ and}$$
$$\not\exists C3 \;/\; C1 \sqsubset C3 \textit{ and } C2 \sqsubset C3 \textit{ and } C3 \sqsubset D$$

Let us suppose for example, two old incidents $c1_{old}$ and $c2_{old}$, description of their abstract concept being:

$C1_{old} \equiv$ *incident* $\sqcap$ *CTU-dependent* $\sqcap$ *more-than-10-minutes* $\sqcap$ *at-least-200-people-affected* $\sqcap$ *ineffective-beam AT-LEAST 4* $\sqcap$ *total-busy-beam AT-LEAST 4* $\sqcap$ *high-temperature.*
$C2_{old} \equiv$ *incident* $\sqcap$ *CTU-dependent* $\sqcap$ *more-than-10-minutes* $\sqcap$ *at-least-200-people-affected* $\sqcap$ *ineffective-beam AT-LEAST 5.*

Neither $C1_{old}$ nor $C2_{old}$ is inherited by $C1_{new}$ and there is no inheritance relation between $C1_{old}$ and $C2_{old}$. However the $\mathcal{LCS}$ of $C1_{old}$ and $C1_{new}$ is

*incident* $\sqcap$ *CTU-dependent* $\sqcap$ *more-than-10-minutes* $\sqcap$ *at-least-200-people-affected* $\sqcap$ *ineffective-beam AT-LEAST 3* $\sqcap$ *total-busy-beam AT-LEAST 2* $\sqcap$ *high-temperature*

and the $\mathcal{LCS}$ of $C2_{old}$ and $C1_{new}$ is

*incident* $\sqcap$ *CTU-dependent* $\sqcap$ *more-than-10-minutes* $\sqcap$ *at-least-200-people-affected* $\sqcap$ *ineffective-beam AT-LEAST 3.*

As the $\mathcal{LCS}(C1_{old},C1_{new})$ inherits $\mathcal{LCS}(C2_{old},C1_{new})$, the instance $c1_{old}$ is preferred to $c2_{old}$ (specificity criterion). In other words the incidents $c1_{old}$ and $c1_{new}$ have more properties in common than the incidents $c2_{old}$ and $c1_{new}$.

**In our application**: To be efficient the process is divided into two steps. Firstly the terminology of $C_{old_i}$ is created then the $\mathcal{LCS}$ operation is applied if no $C_{old_i}$ is inherited by $C_{new}$. For these two steps, the same strategy for (specificity, typical, sure) criteria application as in the conceptual level is used in order to display first of all the most similar instance forms to the operators. As in the conceptual level, thanks to our symbolic criteria the explanation process is easy to justify and can easily be evaluated by the operators. Note that the terminology of abstract old instances is destroyed at the end of each retrieval process to avoid space explosion in the conceptual base (i.e. the index base).

### 4.5 Efficiency

Subsumption, inheritance and abstraction in $\mathcal{ALN}_{\delta\epsilon}$ are computed in polynomial time. The search for similar concepts to a concept $C$ in a concept base, sure and typical criterion are therefore polynomial in time. The refinement function between a concept $C$ and a concept $D$ is computed as follows:

1. We verify that $C$ is subsumed by the strict part of the definition of $D$ because only default properties can be excepted (polynomial in time).

2. We add to the definition of $C$ the exceptions to the concepts which $D$ p-inherits and $C$ does not (polynomial in time).

The search in a (finite) concept base of the concepts which can be used to refine $C$ is therefore done in polynomial time. The $\mathcal{MS}$ function is in $O(n^2)$ ($n$ is the number of concepts of the set), the sequence $\mathcal{U}$ is obviously computed in a polynomial time as $\mathcal{H}$ therefore the exception criterion is polynomial. Finally, the $\mathcal{LCS}$ operation between two concepts have been shown polynomial in time (Ventos et al., 1995b).

In fact this retrieval process is bounded by the classification of the abstract concept $C_{new}$ in the index base which is in $O(l^4)$, l being the number of symbols occurring in the definition of $C_{new}$ (Coupey and Fouqueré, 1993).

## 5. Related works

Over the last few years, different research has been undertaken to use the formalism and (subsumption and classification) inferences of description logics for CBR systems. However none of these approaches enabled a partial matching to be formalized using description logic. In addition to Koehler work described in the introduction several work can be cited. In (Napoli, 1992; Napoli and Lieber, 1993; Napoli and Laurenço, 1993) the authors described the YCHEM system which uses subsumption and classification to build organic synthesis plans. The knowledge representation system they used is the object oriented language YAFOOL (Ducourneau and Quinqueton, 1986). The authors defined a co-subsumption relation (subsumption based on compound-component relations (partonomy)) which is closed to the subsumption relation defined in description logics. Classification was used first to constitute the terminology of molecular structures and second to classify a new molecule to be synthesized (target molecule). In brief, the target molecule (new case) is classified and synthesis plans of the molecules which subsume it were selected to plan the synthesis of the target molecule. However the formal properties of the co-subsumption relation were not described and no results concerning complexity of algorithms were given. In (Napoli and Laurenço, 1993) the authors proposed the premisses of an extension of YCHEM towards a CBR system. The described results mainly concern the adaptation of a synthesis plan for a similar old case in order to apply it to the target molecule. However only exact matchings are considered and no solutions have been suggested concerning ways to order the similar old cases.

H.W. Beck (Beck, 1991) used CANDIDE, a terminological system derived from KANDOR (Patel-Schneider, 1984), in a CBR system. In particular the author used the subsumption relation to evaluate similarity between two instances. Two processes are involved: a deductive process which corresponds to the ideal solution where there is an exact matching between two instances (abstraction + classification) and an inductive process which generates new concepts and enables partial matchings to be found and evaluated. When there is no exact matching between two instances, the system creates a new concept, the description of which is the abstraction of the properties common to the two instances. This new concept is then classified in the terminology and the similarity is evaluated through its place in the terminology. Moreover Beck proposes a sort of exceptional instance when an instance of a concept does not respect certain properties of the concept. However the presentation is only operational and no formal description is given.

In an application for computer-assisted diagnosis and search in a document database, G. Kamp proposed a system based on the LOOM (MacGregor and Bates, 1987) terminological system. The terminological system is used to index the instances from their descriptions, and to search for similar cases. The retrieval process is classical: a concept is created from a request which corresponds to the description of the new case, this concept is classified in the terminology and the instances of the concepts which subsume it are the similar instances. An computer-assisted bicycle diagnosis and repair application is described in (Kamp, 1995). However, only exact matching is considered.

## 6. Conclusion

The goal was to find symbolic criteria to evaluate similarity in the retrieval process and to formalize them to define a clear semantics which is independent of any application. It has been shown that $\mathcal{ALN}_{\delta\epsilon}$ description logic with the two new connectives $\delta$ and $\epsilon$ is a privileged formal framework to achieve this purpose. Thanks to default and exception connectives, subsumption and inheritance relations, abstraction, classification, refinement and $\mathcal{LCS}$ operations, it is possible to formally define *specificity*, *sure*, *typical* and *exception* criteria and partial matching which are the foundations for the retrieval of similar old cases. These criteria are homogeneous, can easily be evaluated by the end-user, can be explained clearly and are useful in this computer-assisted application for the diagnosis of incidents on the French telephone network. They take advantage of the formal semantics of $\mathcal{ALN}_{\delta\epsilon}$ description logic and therefore the efficiency of the retrieval process can be evaluated. Thus it has been shown that it is polynomial in time and that C-CLASSIC$_{\delta\epsilon}$ that is an extension of $\mathcal{ALN}_{\delta\epsilon}$ (C-CLASSIC (Cohen and Hirsh, 1994) with $\delta$ and $\epsilon$ connectives) is PAC-learnable (Ventos et al., 1995b) (which is an important result for the learning process of the application). These criteria are independent of the application and sufficiently general to be used in other applications. Of course we do not claim that they are universal and that they can be used in *all* applications (for example numerical or fuzzy logic criteria can be more adapted for certain applications). $\mathcal{ALN}_{\delta\epsilon}$ has been implemented in C++ and A graet part of the CBR system has been also implemented. Part of the concept terminology has been built and validated by an expert. The study currently under way by a team from INRIA on a supervision site will make it possible to give a clear definition of the set of knowledge of the supervision domain (alarm definitions and management, new action forms, restructured incident forms, etc.). A prototype has been used for a demonstration for our industrial partners and it has been judged as very satisfactory. The final prototype will serve as the basis for the development of an operational system envisaged by CNET, our industrial partner. Our prototype will be used in another project also involving CNET and concerns the construction of a terminological knowledge base on human-computer interfaces (IHM). Classes (concepts) of IHM will be defined from criteria (e.g. syntactic, functional, ergonomic). The system must be able to classify a new IHM in the terminology and find similar IHM to this one in the knowledge base. We conjecture that default and exception knowledge will be also very useful in this project.

## Acknowledgements

The authors are very greatful to the referees for their constructive criticism.

## References

AAMODT, A. and E. PLAZA. 1994. Case-based reasoning: Foundational issues, methodological variations, and system approaches. *The European Journal on Arificial Intelligence, AICOM*, ***7***(1), 39–59.

BECK, H. W. 1991. A conceptual clustering algorithm for semantic data models. In *International Workshop on Terminological Logics, Document D-91-13*, pp. 17–22 SchloßDagstuhl, Germany.

BIÉBOW, B. , P. COUPEY, C. FOUQUERÉ, J. NOBECOURT, S. SZULMAN, and V. VENTOS. 1994. Preliminary report: creation and enrichment of a knowledge base, application to the supervision of the telephone network. Tech. rep., LIPN, Villetaneuse, France.

BORGIDA, A. and P. F. PATEL-SCHNEIDER. 1994. Complete algorithm for subsumption in the CLASSIC description logic. *Journal of Artificial Intelligence in Research*, *1*, 278–308.

BRACHMAN, R. 1985. I lied about the trees. *AI Mag.*, ***6***(3), 80–93.

BRACHMAN, R. and H. LEVESQUE. 1984. The tractability of subsumption in frame-based description languages. In *4th National Conference of the American Association for Artificial Intelligence*, pp. 34–37 Austin, Texas.

BRACHMAN, R. and J. G. SCHMOLZE. 1985. An overview of the KL-ONE knowledge representation system. *Cognitive Science*, ***9***(2), 171–216.

CAULIER, P. and L. PERRON. 1995. La supervision de réseau de télécommunications : analyse de l'activité centrée sur la réutilisation et la catégorisation d'incidents. Tech. rep., INRIA, Rocquencourt, France.

COHEN, W. W. and H. HIRSH. 1994. Learning the CLASSIC description logic: Theoretical and experimental results. In *Principles of Knowledge Representation and Reasoning, Proc. of 5th International Conference*, pp. 121–133 Bonn, Germany.

COUPEY, P. and C. FOUQUERÉ. 1993. Default subsumption in terminological languages. Tech. rep., LIPN, Villetaneuse, France.

COUPEY, P. and C. FOUQUERÉ. 1994a. Définir un concept avec des propriétés par défauts et des exceptions. In *Actes du congrès RFIA*, pp. 197–208 Paris, France.

COUPEY, P. and C. FOUQUERÉ. 1994b. Extending conceptual definitions with default knowledge. revised version in submission.

COUPEY, P. and C. FOUQUERÉ. 1995. Defaults in concept definitions: an application. In *International KRUSE Symposium Knowledge Retrieval, Use, and Storage for Efficiency, (a longer version will appear in a LNAI)*, pp. 112–121 Santa Cruz, USA.

DIONNE, R. , E. MAYS, and F. OLES. 1993a. Concept algebras, objects, and k-rep. In *13th International Joint Conference on Artificial Intelligence, Workshop on Object-Based Representation Systems*, pp. 61–66 Chambéry, France.

DIONNE, R. , E. MAYS, and F. OLES. 1993b. The equivalence of model-theoretic and structural subsumption description logics. In *13th International Joint Conference on Artificial Intelligence*, pp. 710–716 Chambéry, France.

DONINI, F. , M. LENZERINI, D. NARDI, and W. NUTT. 1991a. The complexity of concept languages. In FIKES, J. A. R. and E. SANDEWALL. (Eds.), *Principles of Knowledge Representation and Reasoning: 2nd International Conference*, pp. 151–162 Cambridge, Mass.

DONINI, F. , M. LENZERINI, D. NARDI, and W. NUTT. 1991b. Tractable concept languages. In *12th International Joint Conference on Artificial Intelligence*, pp. 458–463 Sydney, Australia.

DOYLE, J. and R. PATIL. 1991. Two theses of knowledge representation: language restrictions, taxonomic classification, and the utility of representation services. *Artificial Intelligence*, ***48***(3), 261–297.

DUCOURNEAU, R. and J. QUINQUETON. 1986. YAFOOL, encore un langage objet à base de frames. Tech. rep. 72, INRIA.

EWCBR93. 1993. *Proceedings of the 1st European Workshop on Case-Based Reasoning.* SEKI Report SR-92-12, University of Kaiserslautern, Germany.

EWCBR94. 1994. *Proceedings of the 2nd European Workshop on Case-Based Reasoning.* Chantilly, France.

GRATZER, G. 1968. *Universal Algebra.* Van Nostrand Company.

GRELET, H. , A. MARCEROU, T. PASDELOUP, J. PÉRICHON, and C. RATEL. 1992. Traitement des perturbations de trafic en Ile de France. Tech. rep., France Telecom D.R.I.F./RESEAU.

JACOBSON, N. 1989. *Basic Algebra II.* W.H. Freeman and Company.

KAMP, G. 1995. On the use of CBR in corporate service and support. In *Proceedings of the 2nd European Workshop on Case-Based Reasoning*, pp. 175–183 Paris, France.

KOEHLER, J. 1994. An application of terminological logics to case-based reasoning. In *Principles of Knowledge Representation and Reasoning, Proc. of 5th International Conference*, pp. 351–362 Bonn, Germany.

KOLODNER, J. 1993. *Case-based reasoning.* Morgan Kaufmann.

MACGREGOR, R. 1991. The evolving technology of classification-based knowledge representation systems. In MORGAN KAUFMANN, C. SAN MATEO (Ed.), *Principles of Semantic Networks*, pp. 385–400. J. Sowa.

MacGregor, R. and R. Bates. 1987. The LOOM knowledge representation language. Tech. rep. RS-87-188, Information Science Institute, University of Southern California, Maria del Rey, Cal.

Napoli, A. 1992. Subsumption and classification-based reasoning in object-based representations. In *10th European Conference on Artificial Intelligence, ECAI'92*, pp. 425–429 Vienna, Austria.

Napoli, A. and C. Laurenço. 1993. Représentations à objets et classification, conception d'un système d'aide à la planification de synthèses organiques. *Revue d'Intelligence Artificielle*, ***7***(2), 175–221.

Napoli, A. and J. Lieber. 1993. Classifying knowledge for reusability, an application to organic synthesis planning. In *Workshop of the 13th International Joint Conference on Artificial Intelligence, Reuse of designs: an interdisciplinary cognitive approach*, pp. I1–I12 Austin, Texas.

Nebel, B. 1988. Computational complexity of terminological reasoning in BACK. *Artificial Intelligence*, ***34***(3), 371–383.

Nebel, B. 1990. *Reasoning and Revision in Hybrid Representation Systems*. No. 422 in Lecture Notes in Computer Science. Springer-Verlag.

Nobecourt, J. 1995. Une expérience de création d'une base de connaissances en logique de descriptions. In *Journées Acquisition, Validation, Apprentissage, JAVA95*, pp. 255–268 Grenoble, France.

Padgham, L. and T. Zhang. 1993. A terminological logic with defaults: a definition and an application. In *13th International Joint Conference on Artificial Intelligence*, pp. 663–668 Chambéry, France.

Patel-Schneider, P. 1984. Small can be beautiful in knowledge representation. In *IEEE Workshop on Principles of Knowledge-Based Systems*, pp. 11–16.

Perron, L. 1995. La réutilisation de cas "une problématique commune entre I.A. et ergonomie cognitive ··· des points de vue différents". In *Journées Acquisition, Validation, Apprentissage, JAVA95*, pp. 199–212 Grenoble, France.

Schaerf, A. 1994. *Query Answering in Concept-Based Knowledge Representation Systems: Algorithms, Complexity and Semantics Issues*. Ph.D. thesis, Università di Roma "La Sapienza", Roma, Italy.

Schmidt-Schauβ, M. 1989. Subsumption in KL-ONE is undecidable. In *Principles of Knowledge Representation and Reasoning: 1st International Conference*, pp. 421–431. R.J. Brachman H. J. Levesque, and R. Reiter, Toronto, Ont.

Smolka, G. 1988. A feature logic with subsorts. Tech. rep. 33, LILOG, IWBS, IBM Germany Scientific Center, Stuttgart, Germany.

VENTOS, V. , P. BRÉZELLEC, P. COUPEY, and H. SOLDANO. 1995a. Aspects déductifs et inductifs d'un langage de descriptions incluant les connecteurs défaut et exception. Tech. rep., LIPN-Report, université Paris Nord, Villetaneuse, France.

VENTOS, V. , P. BRÉZELLEC, P. COUPEY, and H. SOLDANO. 1995b. C-classic$_{\delta\epsilon}$ : un langage PAC-learnable. In *Journées Acquisition, Validation, Apprentissage, JAVA95* Grenoble, France.

WCBR89. 1989. *Proceedings of the international Workshop on Case-Based Reasoning.* Morgan Kauffmann, Pensacola Beach, Florida.

WCBR91. 1991. *Proceedings of the international Workshop on Case-Based Reasoning.* Morgan Kauffmann, Washington D.C.

WOODS, W. 1991. Understanding subsumption and taxonomy: a framework for progress. In MORGAN-KAUFMANN (Ed.), *Principles of Semantic Networks*, pp. 45–94. J. Sowa.

## 7. Appendix

This section is devoted to present the prototype developed in C++ for our CBR application. Concepts used in this presentation have already been defined in section 4.2. Figure 11 shows a tool implemented to graphically describe knowledge bases. This tool allows to construct/delete/move all concepts and roles of a knowledge base. The terminological knowledge base in figure 11 is relative to our example in section 4.2. Heavy arrows are strict links, thin arrows are default links, dashed arrows are exception links, circles are concepts and squares are roles. Different kind of concepts (primitive, partially defined and fully defined) are distinguished in the tool by different colors which unhappily do not appear in this black and white figure. This tool allows also to translate a graphic into an ASCII text which is the knowledge base in the description language of $\mathcal{ALN}_{\delta\epsilon}$. The text below is the translation of the knowledge base of the figure 11: *defprimc* is the command to define a new primitive concept, *defpconcept* defines a new partially defined concept, *deffconcept* is to define a new fully defined concept and *defprimr* is to define a new primitive role. The other commands are quite easy to understand (*at-least, at-most, all, and*).

**The knowledge base in our description language**:

```
(defprimc incident);
(defprimc more-than-10-minutes);
(defprimc at-least-200-people-affected);
(defprimc fire-alarm);
(defprimc equipment-dependent);
(defprimc service);
(defprimc number-overcall);
(defprimr number-affected);
(defprimr total-busy-beam);
(defprimr ineffective-beam);
(defprimr ineffective-overflowing-beam);
(deffconcept serious-incident
        (and incident
```

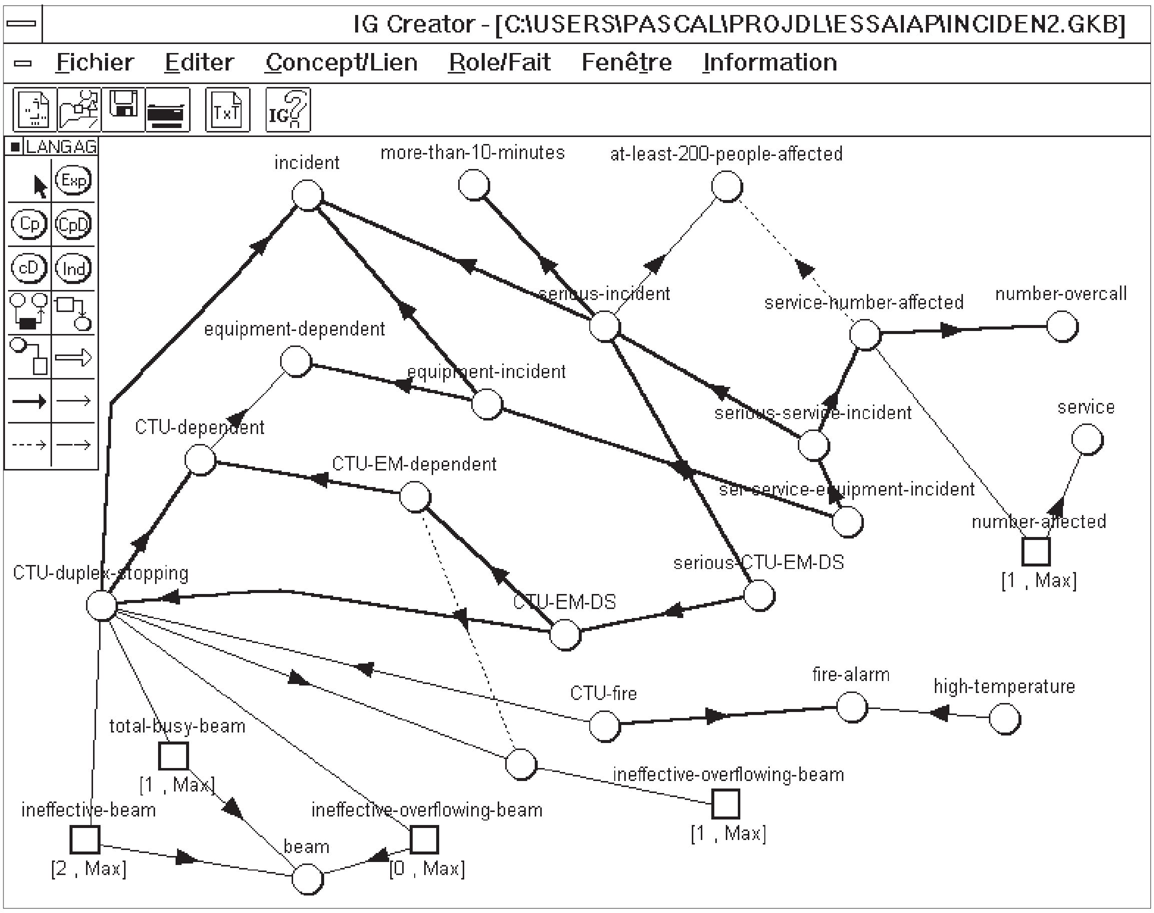


Figure 11: Graphic description of the knowledge base.

```
                 more-than-10-minutes
                 (default at-least-200-people-affected)));
(deffconcept equipment-incident
           (and equipment-dependent
                 incident));
(defpconcept CTU-dependent
           (default equipment-dependent));
(deffconcept CTU-duplex-stopping
           (and incident
                 (at-least 1 total-busy-beam)
                 (at-least 2 ineffective-beam)
                 CTU-dependent
                 (default (at-least 1 ineffective-overflowing-beam))
                 (all ineffective-beam beam)
                 (all total-busy-beam beam)
                 (all ineffective-overflowing-beam beam)));
(deffconcept CTU-EM-dependent
           (and CTU-dependent
                 (exception (at-least 1 ineffective-overflowing-beam))));
(deffconcept CTU-EM-DS
```

```
          (and CTU-duplex-stopping
               CTU-EM-dependent));
(deffconcept serious-CTU-EM-DS
          (and CTU-EM-DS
               serious-incident));
(deffconcept CTU-fire
          (and fire-alarm
               (default CTU-duplex-stopping)));
(defpconcept high-temperature
          (default fire-alarm));
(defpconcept service-number-affected
          (and number-overcall
               (all number-affected service)
               (at-least 1 number-affected)
               (exception at-least-200-people-affected)));
(deffconcept serious-service-incident
          (and serious-incident
               service-number-affected));
(deffconcept ser-service-equipment-incident
          (and serious-service-incident
               equipment-incident));
```

Figure 13 is the CBR interface (conceptual level) of our prototype. This interface allows to load a knowledge base (the base loaded in this figure is the knowledge base shown in figure 11) and to retrieve similar concepts to a source concept relatively to a target concept. The retrieval is achieved thanks to the inference services (subsumption, classification, inheritance) proposed by the core of the whole system, the description logics $\mathcal{ALN}_{\delta\epsilon}$. In the example, all concepts similar to $C_{new}$ and which are incidents (target concept is *incident*) have been retrieved. The list of similar concepts is displayed in the left of the figure. The description of the new incident is given in section 4.2 ($c1_{new}$). $C_{new}$ is the abstract concept of this new incident, a graphic representation of which is given in figure 12.

The center bottom of this interface displays the set of properties of a selected similar concept in the list. In figure 13, the current selected concept is *serious-incident*: its specificity value is 1,"sûr" and "typique" mean that the new incident is a sure and typical incident of *serious-incident* and $exception = 0$ means that no exception has been added to the description of the new incident to find *serious-incident* is similar to it (it is not a refined concept). The order of the display of the list of similar concepts can be changed by varying priority of preference criteria (in the right bottom of figure 13). The highest priority is 4, the lowest is 1, and 0 means that the criteria is not taken into account to order the similar concepts. In the example the priority of the specificity criterion is 4: it means that this criterion is used at first to order the concepts in the list. Thus the first concepts displayed in the list have their specificity value equal to 1 then the following ones have their specificity value equal to 2 and so on. The priority of the sure criterion (3) means that inside a set of concepts with the same specificity value, those for which the new incident is a sure instance are displayed before those for which the new incident is a probable instance. Finally in the example the display of similar concepts in the list is given by applying at first the specificity

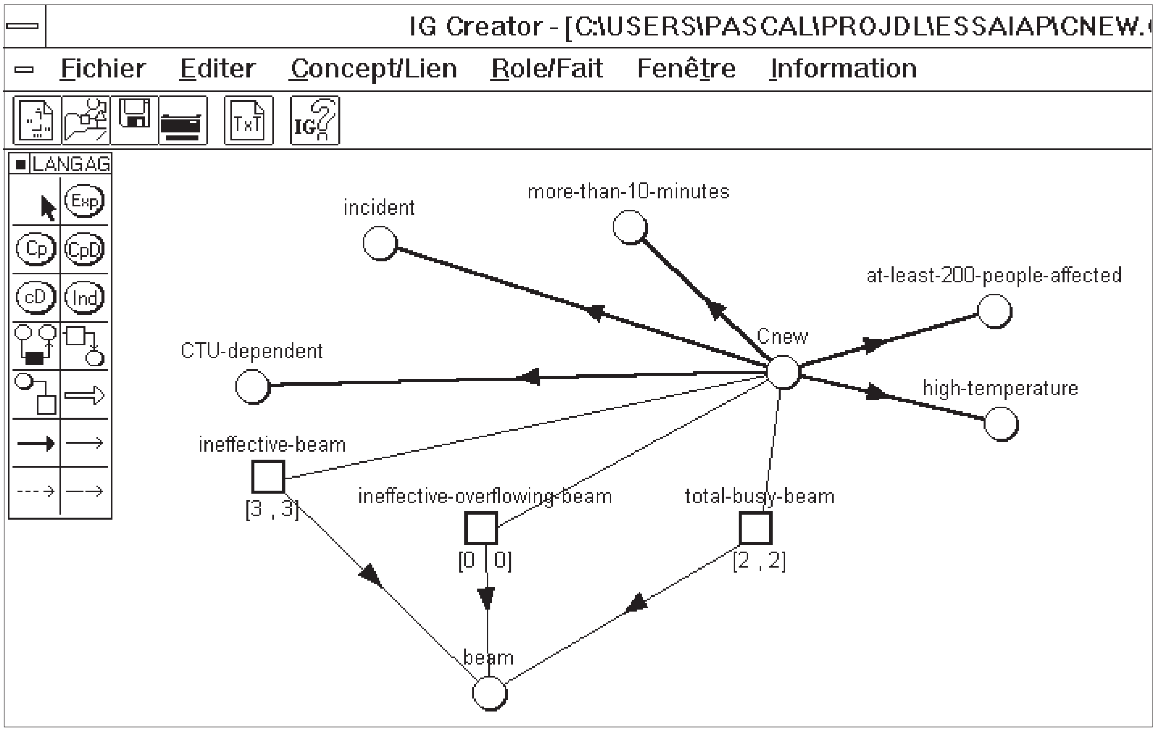


Figure 12: Graphic description of the abstract concept of the new incident.

criterion, then the sure, then the typical and finally the exception criterion. The end-user can change the priority value of the criteria to get different displays.

The CBR interface allows also to display the action form of an incident concept and to access to another CBR interface (instance level) to display the list of instances of a concept and their incident forms (the implementation of this interface is in progress).

Figure 14 is the same interface (conceptual level) with the same knowledge base as above but the source concept is the refined concept of $C_{new}$, called $C1^{ex}_{new}$ in section 4.3. An exception to *at-least 1 ineffective-overflowing-beam* has been added to the description of the new incident. Thanks to the exception this refined concept inherits concepts which are not inherited by $C_{new}$. Thus in the example, the end-user can see that the new incident could be a *CTU-fire*.

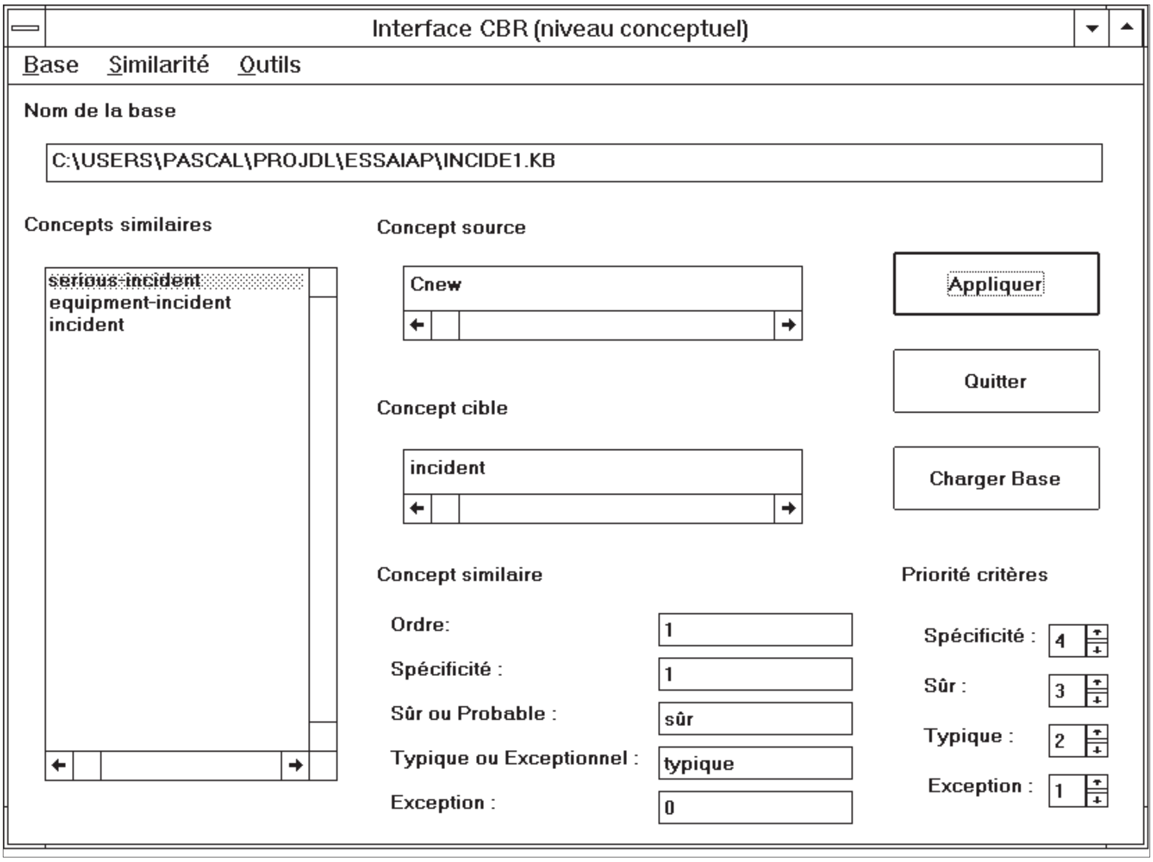


Figure 13: The CBR interface (conceptual level).

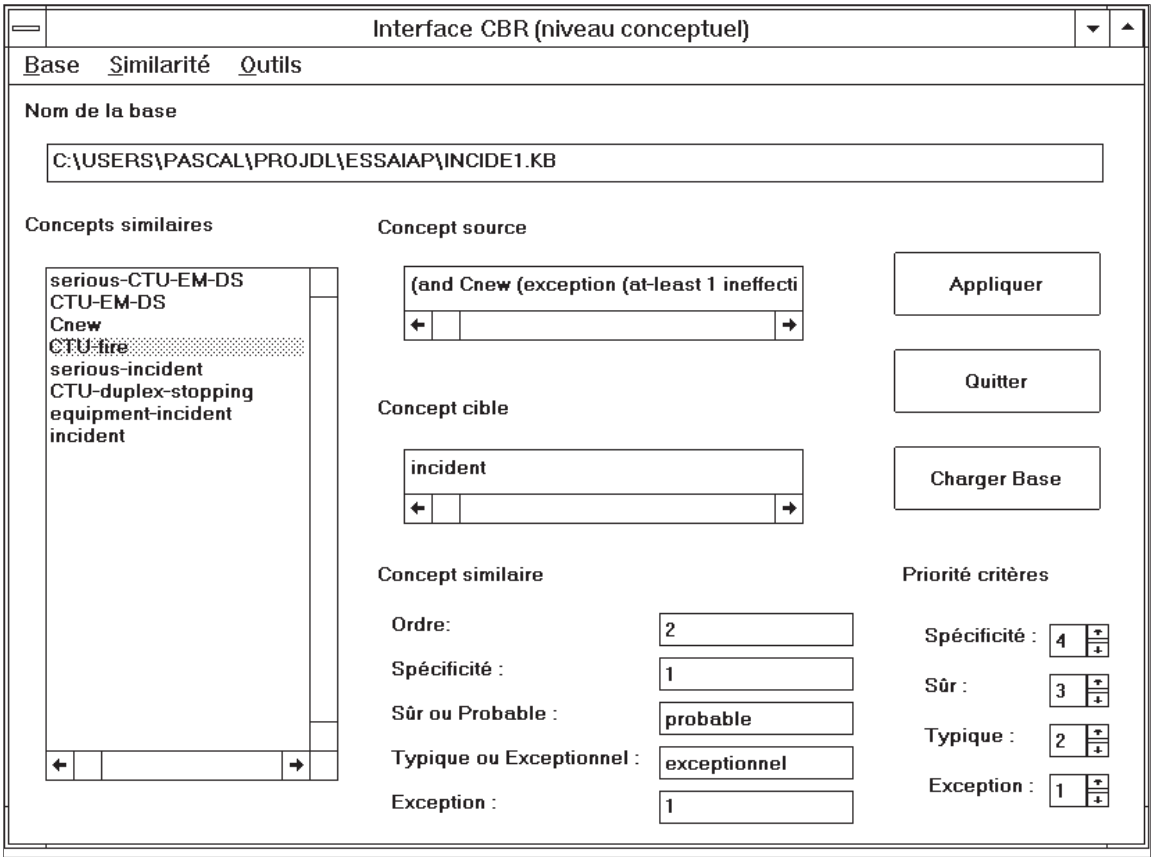


Figure 14: A refined concept of the new incident