Making Inferences on Tangled Hierarchies: A Paraconsistent Approach

Bráulio Coelho Ávila
Department of Informatics
Pontifical Catholic University of Paraná
Curitiba, PR, Brazil

Fernanda Hembecker
Department of Informatics
Pontifical Catholic University of Paraná
Curitiba, PR, Brazil

Abstract *Frames are a knowledge representation scheme allowing the description of complex objects. However, a gap exists between the knowledge represented by frame systems and knowledge in the real world. Few frame systems are concerned with bridging this knowledge gap; they do not handle adequately issues such as exceptions and the inconsistency phenomenon. One of the shortcomings found in such systems is the lack of a formal semantics to treat exceptions and inconsistencies. In this paper a paraconsistent inheritance reasoner in Paralog_e is implemented, representing knowledge by means of paraconsistent frames and making inferences on tangled hierarchies, based on the degree of inconsistency/under-determinedness. Furthermore, its main characteristic is not eliminating contradictions* ab initio*. The inheritance reasoner implemented allows more adequate treatment of exceptions and inconsistent information in multiple inheritance frame systems.*

Keywords: frame systems, multiple inheritance, exceptions, inconsistency, paraconsistent inheritance reasoner.

1 Introduction

A frame is a representation of a complex object [8]. It is identified by a *name* and consist in a set of slots. Each frame possesses at least one hierarchically superior frame, thus providing the basis for the inheritance[1] mechanism. A special frame is the root of this inheritance hierarchy. The two main types of existing inheritance systems are: those not admitting exceptions to inherited properties and those admitting exceptions to inherited properties.

It is easy to describe the semantics of the first inheritance system in first order Classical Logic, in which frames may be interpreted as unitary predicates and slots may be interpreted as binary predicates. The description of the semantics of the second inheritance system in first order Classical Logic is much more difficult, because exceptions introduce nonmonotonicity.

Several nonmonotonic formalisms have been proposed since the late 70's. Among the most widespread are: Clark's *predicate completion*, Reiter's *default logic*, McDermott and Doyle's *nonmonotonic logic I*, McCarthy's *circumscription technique*, McDermott's *nonmonotonic logic II*, and Moore's *autoepistemic logic*. None of these formalisms, however, adequately handles issues such as the inconsistency phenomenon.

To implement frame systems dealing with the inconsistency phenomenon one must take into account the difficulty caused by the lack of a formal semantics both for paraconsistent Frame Systems and for inheritance reasoners dealing with exceptions and inconsistencies in multiple inheritance frame systems.

The first attempts to treat exceptions systematically later proved to be incorrect in the presence of redundant links and incon-

[1] Inheritance is a specialized inference technique using a type of syllogistic reasoning.

sistencies. The first comprehensive definition to solve such problems was Touretzky's [11]. Since then, many other equally correct systems have been proposed. Nevertheless, despite a decade of studies, with the increasing subtle examples and counter-examples being considered, a consensus concerning multiple inheritance treatment with exceptions in such hierarchies is still to emerge [9][2]. It must also be stressed that none of these schemes handle adequately issues such as the inconsistency phenomenon, despite such phenomena being increasingly common in computing environments.

2 Paraconsistent Multiple Inheritance Reasoner

Inconsistency is a natural phenomenon arising from the description of real world aspects. Despite such phenomena being increasingly common in programming environments — particularly in those having a certain degree of distribution — they cannot be treated, at least directly, by means of Classical Logic, on which most current programming languages are based. Therefore, one has to resort to alternatives to classical logic to study such inconsistencies directly, it being thus necessary to look for programming languages based on such alternative logic. The work by vila *et al.* [3] proposes an extension of the ParaLog logic programming language, called ParaLog_*e*, allowing inconsistency to be handled directly.

The development of the ParaLog_*e* language is based on an *annotated paraconsistent logic* [1], infinitely valued, where the truth-values are members of $\{x \in \Re \mid 0 \leq x \leq 1\} \times \{x \in \Re \mid 0 \leq x \leq 1\}$. According to Subrahmanian [10], a two-evidence annotation may be thought of as: an *favorable evidence* to p and an *contrary evidence* to p. No restriction is placed on these evidences, except that they be within the $\{x \in \Re \mid 0 \leq x \leq 1\}$ interval. The use of both these evidences increases the expression power of the information represented. Thus it may be seen that by using the Paralog_*e* language, the representation of information in frame systems may be performed more naturally. Languages such as ParaLog_*e*, capable of merging Classical Logic Programming concepts with those of inconsistency, widen the scope of Logic Programming applications in environments presenting conflicting beliefs and contradictory information.

The implementation of a paraconsistent inheritance reasoner in ParaLog_*e* is described below, allowing handling exceptions and inconsistencies in multiple inheritance frame systems.

2.1 Description of the Paraconsistent Inheritance Reasoner

The development of an inheritance reasoner for frames requires three major decisions to be drawn, concerning the way of representing: the inheritance network, the frame knowledge, and the reasoner inference-making.

2.1.1 Inheritance Network Representation

In the first decision, one can observe that almost all the inheritance systems found in literature represent knowledge through semantic nets [11] [7]. This paper represents knowledge by means of paraconsistent frames. Frames are a more complex knowledge representation scheme than semantic nets.

The paraconsistent frame inheritance network — frame network — of the paraconsistent reasoner is a *tangled hierarchy*[3]. That is, it allows frame data to possess multiple ascending frames, and thereby multiple inheritance. In this paper, ascending links between two frames may be:

- strict; and

[2] As there is no formal semantics for inheritance reasoning with exceptions so far, the integrity of a scheme is currently determined by checking its behavior by means of a set of inheritance network examples.

[3] The basic terminology adopted in this paper is the same as Touretzky's [12]

- defeasible.

Strict links are represented by relationships *ako* with perfectly defined evidences $[1.0, 0.0]$ and $[0.0, 1.0]$. That is, evidence $[1.0, 0.0]$ represents that a frame A *ako* frame B. On the other hand, evidence $[0.0, 1.0]$ represents that a frame A *not-ako* frame B. Strict links ako and not-ako are also known as *positive links* and *negative links*, respectively, in *bipolar inheritance systems*.

Defeasible links are represented by ako relationships with evidences belonging to the $\mathcal{T} = \{x \in \Re \mid 0 \leq x \leq 1\} \times \{x \in \Re \mid 0 \leq x \leq 1\}$ lattice, excepting $[1.0, 0.0]$ and $[0.0, 1.0]$. That is, for instance, a $[0.9, 0.3]$ evidence represents that a frame A, generally, ako frame B, but with a 20% inconsistency/under-determinedness degree[4].

This frame network also admits the existence of redundant ascending links between two frames, because the network and the reasoner are based on ParaLog_*e*, and after the closure transformation, the resulting link is the *supreme* of redundant link.

2.1.2 Knowledge Representation in Frames

In the second decision, knowledge, as said before, is represented by Paraconsistent Frames. The way of representing such frames is similar to the standard frame representation. However, the information represented in paraconsistent frames has greater expression power, since it is associated with evidences. In ParaLog_*e* a frame may be represented as a fact, that is, one fact for each slot facet, as shown next.

$$< frame > (< slot >, < facet >, < value >) : [< e_f >, < e_c >].$$

For instance,

$$c(color, value, blue) : [0.9, 0.3].$$

specifies that a frame c possesses a *color* slot with *blue* value — 90% *favorable evidence* and 30% *contrary evidence* — with an inconsistency/under-determinedness degree of 20%.

This paper admits two types of facets:

1. value; and

2. exception.

The first facet specifies the *value* assumed by the slot. The value facets may be:

- strict; and

- defeasible.

Strict value facets have perfectly defined evidences $[1.0, 0.0]$ and $[0.0, 1.0]$. For instance,

$$a(color, value, blue) : [0.0, 1.0].$$

specifies that a frame a has a *color* slot of *non-blue* value.

Defeasible value facets possess evidences pertaining to lattice $\mathcal{T} = \{x \in \Re \mid 0 \leq x \leq 1\} \times \{x \in \Re \mid 0 \leq x \leq 1\}$, excepting $[1.0, 0.0]$ and $[0.0, 1.0]$. For instance,

$$a(color, value, blue) : [0.3, 0.6].$$

specifies that a frame a has a *color* slot of *blue* value, but with a 10% degree of inconsistency/under-determinedness.

Value facets also admit the existence of redundant value facets. After the closure transformation, the resulting value facet is the supreme of redundant value facet.

The last facet specifies the *exceptions* assumed by the slot. There might be one or more exception facets for each slot. Exception facets may be:

- strict; and

- defeasible.

Strict exception facets have perfectly defined evidences $[1.0, 0.0]$ and $[0.0, 1.0]$. For instance,

$$c(color, exception, red) : [1.0, 0.0].$$
$$c(color, exception, black) : [1.0, 0.0].$$

[4] The inconsistency/under-determinedness degree is obtained according to [10]

specify that a frame c possesses a *color* slot with *red* and *black* exceptions.

Defeasible exception facets possess evidences pertaining to lattice $\mathcal{T} = \{x \in \Re \mid 0 \leq x \leq 1\} \times \{x \in \Re \mid 0 \leq x \leq 1\}$, excepting $[1.0, 0.0]$ and $[0.0, 1.0]$. For instance,

$$c(color, exception, red) : [0.2, 0.5].$$
$$c(color, exception, black) : [0.1, 0.3].$$

specify that a frame c possesses a *color* slot with *red* and *black* exceptions, but with a 30% and 60% degree of inconsistency/under-determinedness, respectively.

Exception facets also admit the existence of redundant exception facets. After the closure transformation, the resulting exception facet is the supreme of redundant exception facet.

2.1.3 Reasoner Inference-Making

Finally, the last decision concerns how the inheritance reasoner draw inferences on frames. This decision is related to the foregoing ones, that is, it depends both on the inheritance network representation form and on the knowledge representation form in frames.

This paper, according to Touretzky's [12] terminology, may be classified as complex, since it is a system possessing the following characteristics: multiple inheritance, bipolarity, heterogeneity and nonmonotonicity, that is, it allows multiple ascending links, positive and negative links, strict and defeasible links, and exceptions to property inheritance.

The implemented inheritance reasoner for frames, in addition to meeting all these characteristics allows, being an encompassing reasoner, all other less complex inheritance forms, such as unipolar monotonic homogeneous inheritance, etc., to be carried out.

It is also worth highlighting that its main feature is not eliminating contradictions, *ab initio*, since in several cases of real world description the existence of contradictions is important in itself. Therefore, since the inheritance reasoner is implemented in a paraconsistent logic programming language, ParaLog_*e*, it handles inconsistent information adequately.

Paraconsistent Inheritance Reasoner Algorithm

Input: an acyclic paraconsistent frame system Γ and a query $q(f, s, v)$:$[q_f, q_c]$
Output: a set of conclusions C_n with its respective evidences $[C_{f_n}, C_{c_n}]$.

```
begin
  perform the Γ negation elimination procedure
  perform the Γ closure procedure
  Exceptions:= {}
  Equal_n:= {elem(f:[e_f, e_c],Exceptions)}
  Parent:= {}
  Sol:= 0
  ex_equal(q(f, s, v):[q_f, q_c],Γ,C_n:[C_fn, C_cn],Equal_n,
  Parent,Parent_ret,Sol,Sol_ret)
  if Sol_ret = 0
    then
      if Parent_ret ≠ {}
        then
          continue(q(f, s, v):[q_f, q_c],Γ,C_n:[C_fn, C_cn],
          Parent_ret)
        else
          C:[C_f, C_c] := {}:[0.0, 0.0]
      end if
  end if
end
```

Figure 1: Paraconsistent Inheritance Reasoner Algorithm to Handle Exceptions in Multiple Inheritance Frame Systems

2.2 Algorithm

The paraconsistent inheritance reasoner is implemented in ParaLog_*e* on the basis of the algorithm detailed in Figure 1. In this algorithm, regularization procedures are performed first. They are followed by the initialization of variables and the performance of the **ex_equal** procedure. At last, on the basis of the output values of such procedure, the **continue** procedure is performed, or an empty value is attributed to conclusion C.

The **ex_equal** procedure tries to find solutions to frames possessing equal inconsistency / under-determinedness degrees. When such solutions are found, a procedure combining the evidences of the solutions is performed. Otherwise, a procedure is performed trying to find parent frames for paths that did not produce solutions. This is necessary for the inheritance reasoner to search ascendants in Γ.

The **continue** procedure allows the search for ascendants in Γ, that it, since no solutions were found in the frames searched, the search must continue upward to the parent frames belonging to Γ. For this search to be performed, the parent frames found in the paths that did not produce solutions must first be ordered. Such ordering must be performed on the basis of the inconsistency/under-determinedness degree accrued from each path. Afterwards, the ordered elements are separated into two groups: elements with equal inconsistency/under-determinedness degrees and elements with higher inconsistency/under-determinedness degrees. Then, the **ex_equal** procedure is again performed based on these two groups of elements. This procedure must be repeated until no more parent frames exist or a solution is found.

3 Conclusions

The main objective of this work was to show that by using Paraconsistent Logic [4], the frame systems may reason adequately in the presence of exceptions and inconsistent information. To achieve such objective, a paraconsistent inheritance reasoner was implemented in ParaLog_*e*. The fact that a formal semantics does not exist both for paraconsistent frame systems and for inheritance reasoners handling exceptions and inconsistencies in multiple inheritance frame systems presented a problem, but also a motivation for the implementation of this type of reasoner.

The paraconsistent inheritance reasoner developed in this paper, according to Touretzky's [12] terminology, may be classified as complex, since it is a system possessing the following characteristics: multiple inheritance, bipolarity, heterogeneity and nonmonotonicity, that is, it allows multiple ascending links, positive and negative links, strict and defeasible links, and exceptions to property inheritance. Despite all of that, it leads to appropriate conclusions. This reasoner, in addition to meeting all these characteristics allows, being an encompassing reasoner, all other less complex inheritance forms, such as unipolar monotonic homogeneous inheritance, etc., to be carried out.

The implemented inheritance reasoner represents knowledge through paraconsistent frames and performs inferences on tangled hierarchies, based on the inconsistency/under-determinedness degree. Furthermore, its main feature is not eliminating contradictions, *ab initio*. Thus, it may be concluded that the implemented paraconsistent inheritance reasoner allows better handling of the exceptions and inconsistent information in multiple inheritance frame systems.

Paraconsistent Logic, despite having been initially developed from the purely theoretical standpoint, found in recent years extremely fruitful applications in Computing Science [2] [5] [6], thus solving the problem of justifying such logic from the practical standpoint.

References

[1] Abe, J.M., *Fundamentos da Lógica Anotada*, (Foundations of Annotated Logics), Ph.D. Thesis, Faculdade de Filosofia, Letras e Ciências Humanas, Universidade de São Paulo, São Paulo, Brazil, 1992. (in Portuguese)

[2] Abe, J.M.; Ávila, B.C.; Prado, J.P.A., *Multi-Agents and Inconsistence*, ICCIMA, p.131-42, Traralgon, Australia, 1998.

[3] Ávila, B.C.; Abe, J.M.; Prado, J.P.A., *Paralog_e: A Paraconsistent Logic Programming Language*, ICCIMA, p.143-148, Traralgon, Australia, 1998.

[4] da Costa, N.C.A., *Calculs Propositionnels pour les Systèmes Formels Inconsistants*, Compte Rendu Acad. des Sciences (Paris), 257, pp. 3790-3792, 1963.

[5] Enembreck, F.; Ávila, B.C.; Sabourin, R., *Decision Tree-Based Paraconsistent Learning*, XIX International Conference of the Chilean Computer Science Society, IEEE Computer Society Press, pp. 32-44, Talca, Chile, 1999.

[6] Enembreck, F.; Ávila, B.C.; Sabourin, R., *Using Paraconsistent Logic in Automatic*

Signature Verification, Argentine Symposium on Artificial Intelligence, pp. 87-99, Tandil, Argentina, 2000.

[7] Krishnaprasad, T.; Kifer, M., *An Evidence-Based Framework for a Theory of Inheritance*, Proc. 11^{th} IJCAI-89, pp. 1093-1098, Detroit, 1989.

[8] Minsky, M., *A Framework for Representing Knowledge*, MIT AI Memo 306, Cambridge, MA, June, 1974.

[9] Selman, B.; Levesque, H.J., *The Tractability of Path Based Inheritance*, Proc. 11^{th} IJCAI-89, pp. 1140-1145, Detroit, 1989.

[10] Subrahmanian, V.S., *Towards a Theory of Evidential Reasoning in Logic Programming*, Logic Colloquium '87, The European Summer Meeting of the Association for Symbolic Logic, Granada, Spain, July, 1987.

[11] Touretzky, D.S., *The Mathematics of Inheritance Systems*, Research Notes in Artificial Intelligence, Pitman, London, 1986.

[12] Touretzky, D.S.; Horty, J.F.; Thomason, R.H., *A Clash of Intuitions: The Current State of Nonmonotonic Multiple Inheritance Systems*, 10^{th} IJCAI-87, Milan, pp. 476-482, 1987.