AN ARCHITECTURE TO INTEGRATE DISCOVERED KNOWLEDGE IN A RULE BASED SYSTEM

Paulo Oliveira, Fátima Rodrigues
GECAD – Knowledge Engineering and Decision Support Group
Institute of Engineering – Polytechnic of Porto
Porto, Portugal
pjo@isep.ipp.pt, fr@isep.ipp.pt

Gabriel David
Faculty of Engineering – University of Porto
Porto, Portugal
gtd@fe.up.pt

The techniques and tools of Knowledge Discovery in Databases seek to transform data into knowledge in an "intelligent" and semi-automatic way. One of the possible uses to this discovered knowledge consists in its integration or fusion with the knowledge that is in the knowledge base of an Expert System. It thus complements the knowledge initially given by the expert, which is not always complete, or the most up-to-date. Using an alternative source it is possible to discover knowledge that is implicit in data, and then proceed with its fusion with the one already in the knowledge base. However, this process can result in errors appearing (for example, inconsistencies) in the knowledge base resulting from the fusion. Thus, one of the requirements to fulfil is the consistency and correction of this new knowledge base.

A generic and domain independent architecture that allows a rule based knowledge fusion, in the context above described is presented. Consistency and correction are guaranteed through the detection of errors, and by the adoption of an approach based in maximal consistent subsets of rules.

Keywords: rule based system, knowledge discovery in databases, rule bases fusion, rule bases fusion architecture, consistent knowledge base, maximal consistent subsets of rules.

I. INTRODUCTION

Considering the several technologies coming from the Artificial Intelligence area, Expert Systems (ESs) are one of the most popular. In theory, consulting an ES allows to obtain the same results that would be obtained from an expert. ESs base their operation on the specific knowledge acquired from domain experts, which is kept in a separate module designated as Knowledge Base (KB).

In spite of, commercially, being the branch of Artificial Intelligence applications that have greater acceptation and divulgation, naturally, ESs also have some limitations. Between these, we detach the fact that:

- They are limited to the knowledge that is in the KB and don't have any capability to produce conclusions about situations of which they don't possess knowledge, even if these situations are trivial. Usually, they also don't have any learning capacity that allows them to acquire new knowledge.
- They are dependent of maintenance operations (addition, change or even knowledge elimination) on its KB, to be able to maintain their efficiency and utility degree. In almost all the domains, knowledge has a dynamic nature, that is, it evolves as times goes by. The absence of knowledge maintenance operations is one of the reasons that leaded many ESs projects to failure.

In the last decades, the development of information technologies originated an enormous increase in data production and storage, in almost all human activity areas. This data abundance resulted in a situation characterized by a "data richness, but knowledge poverty", accordingly the human incapacity to analyse and understand their contents. Due the limitations of traditional data analysis methods, a new set of tools and techniques appeared. These assist, in an intelligent and semiautomatic way, the human beings to transform huge data volumes into knowledge that is useful to their owners. These tools and techniques insert themselves in the context of what was designated as Knowledge Discovery in Databases (KDDs) [1].

Progressively, the utilization of Knowledge Discovery in Databases Systems (KDDSs) is increasing. These systems have vast potentialities of extracting knowledge that is implicit in data, and may be considered as a font of new knowledge. This knowledge could be used with different purposes. One of the possible uses is to add or incorporate it in an existing ES, to actualise its KB. This knowledge acts as a complement to that supplied by the expert. The expert not always owns all the knowledge or the most up-to-date, especially in domains where it has a high evolution rate. However, part of this knowledge may be obtained from recent data. The idea is to take advantage of the knowledge produced by a technology fairly recent (KDDs), and incorporate it in the KB of another technology quite more older (ESs), helping this way to surpass the limitations mentioned previously.

The incorporation of the new (discovered) knowledge will be accomplished through the integration or fusion with the (existing) knowledge that is in the ES KB. Among others possible errors, this fusion could result in inconsistencies appearing, that is, conflicting perspectives about a same situation could appear. One of the requirements of this fusion process is to obtain a new (resulting) KB, in which the knowledge is consistent and correct.

II. Knowledge Fusion

During the last decades, the knowledge-based systems propagated considerably, providing a great variety of knowledge in different domains [2]. In presence of such richness and diversity of knowledge sources available today, one of the greatest challenges that the development of automatic reasoning systems meets, is to fuse, in an intelligent manner, the knowledge coming from those fonts with the aim to solve complex problems.

The problem of knowledge fusion occurs daily, in several tasks that an intelligent agent executes as, for instance, in belief or evidence gathering, in specifications development, in regulations fusion and in decision-making. The purpose of knowledge fusion is to form a unique perspective or point of view, a synthesis or a consensus [3].

Despite occurring frequently, knowledge fusion is an important and not trivial problem [4].

The problem is important because knowledge that results from a fusion process makes available, in a centralized way, knowledge that is spread in various knowledge sources. Besides, it permits the deduction of additional knowledge that is not possessed individually by any of the sources, but collectively by all [4]. This new knowledge is called implicit knowledge [5]. For instance, if one source has the knowledge that A is true and another that $A \rightarrow B$, the resulting knowledge permits to conclude that B is also true, although such knowledge can't be found in any of the two knowledge sources. These two aspects constitute the main advantages of knowledge fusion.

The problem is not trivial because, often, the knowledge coming from the various sources presents contentious perspectives about a same situation.

III. Approaches Used in Knowledge Bases Fusion To Create a Consistent KB

KBs fusion is defined as a process from which different KBs results only in one consistent KB. Following this line of orientation, there are approaches that give different credibility to the KBs that are involved in fusion, like belief revision [6] and belief update [7, 8]. These approaches study the problem of accommodating a new belief (something that is believed, an opinion or conviction), completely reliable, in an existing belief set, preserving the consistency and, at the same time, avoiding as much as possible knowledge loss during the process, through a minimal change strategy. The principle of minimal change can be stated like this: given a belief set K and a new belief α, the resulting belief set KR should differ from K the least as possible [9]. Both the approaches considerate the presupposition that the new belief is more reliable than those actually existing in the belief set in such a way that it should be present in the updated belief set. There are also approaches that give equal importance to the new and actual knowledge. In this case it is not possible to establish any preference. The handling of this kind of situations is the purpose of KBs arbitration [10, 11] and KBs combination [4, 12].

A. *Belief Revision*

Belief revision or theory revision is the process that permits and explains the transition between epistemic states (beliefs) in a rational agent, when confronted with new knowledge, about a static world. Revision results from a change in the description possessed about the real world although, in practice, it remains unchanged. A KB can be interpreted as a representation of an epistemic state. Belief revision studies the problem of its evolution, having consistence preservation as purpose, after a new evidence arrival. Revision assume that the new belief is always more important and reliable than those that are in the existing belief set, admitting that this may be, partially incorrect.

The new belief may express that something considered true, never was. The generality of the research made in this area is influenced by the work in philosophical logic, in particular from Gärdenfors and his team that developed the belief revision theory [6, 13, 14].

Suppose the existence of a consistent belief set symbolically represented by K. At a certain point, a new belief arrives, represented by α. Considering the presupposition in which is based belief revision that states that the newly arrived belief is entirely believable, this must be assimilated by the belief set K. The new belief may, or not, enter in conflict with this one. Belief revision becomes trivial when α is consistent with K. The difficulty appears when K and α are inconsistent, what forces the conflict resolution. To do so, it is necessary to remove some belief(s) from K, in order that α may be inserted preserving consistency. Usually, there are several ways to do it, so that the process should be executed in such a way that the knowledge loss be as least as possible.

B. *Belief Update*

Belief update is, sometimes, considered as being a variant to belief revision [15]. In spite of having similarities, these two processes are, in their essence, different [16, 17].

Belief revision, as defined previously, is the operation that allows changing existing beliefs about a static world, when a new belief becomes known. Winslett demonstrated that when knowledge is about a world susceptible to changes, the update operation properties must be different [18].

An update happens in result of a change in real world, what makes the belief set incorrect. Belief update is the process through which is possible to keep beliefs updated in relation to a mutation world [19]. Belief update assumes that the new belief is more recent than those that are in the existing belief set. Both are considered entirely believable, but referring to different time instants. Between these instants, some changes may have happened. The new belief reports that something considered true, is not anymore.

C. *Knowledge Bases Arbitration*

Revesz proposes another view in KBs fusion named KBs arbitration [10]. The idea that is in its origin is: new knowledge is not better or worst than the one already in the KB. The new knowledge is considered just as one more argument against a set of another arguments already existing.

Either belief revision or belief update presupposes that new belief is, totally, reliable in such a way that should be integrated in the new belief set. However, this presupposition should not be considered when fusing knowledge coming from different sources, referring to different time instants and having the same credibility degree. In fact, this is a very common situation. When two different knowledge sources present a contradictory view about a same situation, don't existing any reason to consider any of them doubtful, the best that can be made is to fuse the two views, possibly inconsistent, in a new one that is consistent, preserving the maximal knowledge as possible.

Such as in belief revision and belief update, in arbitration the minimal change principal is also implicit. Depending on the operator used, KBs arbitration seeks the solution that better satisfies, globally, each part involved in the conflict, or the solution that better satisfies the majority of these.

D. *Knowledge Bases Combination*

KBs combination was developed with the purpose to help ESs construction, when the knowledge comes from several experts, all with the same credibility. The knowledge provided by each of these is codified in a separate KB being, subsequently, combined in only one KB representing the group knowledge. This approach is based in determining the Maximal Consistent Subsets (MCSs) resulting from the union of the several KBs, and by selecting some of those subsets, according to a given criterion. Baral and his team present several methods (operators) to make this KBs combination [4, 12]. Some of the methods allow KBs combination without the existence of preferences between the KBs, while others allow the combination when preferences exist. However, these researchers did not develop a formal theoretical model to multiple KBs combination [20].

The problem of combining several KBs is placed this way: considering the existence of a KBs set; a set of restrictions; the presupposition that the set of restrictions is the same for all the KBs; and that each KB is consistent with the restrictions. The purpose of these methods is to combine the set of KBs, in such a way that the resulting KB be maximal, between the several correct and consistent combinations. The underlying motivation of the maximal concept is to include in the final KB as much knowledge as possible from each one of the KBs of the combination process.

IV. Proposed Architecture to Knowledge Bases Fusion

In this section a KBs fusion architecture (in the rules form) is presented in detail. According to the problem initially exposed, the purpose of this architecture is to integrate or fuse the knowledge proceeding from a KDDS with that stored in a KB of a Rule Based System (RBS).

In the previous section were briefly presented several approaches that can be used to create a consistent KB resulting from KBs fusion. Between these, some consider that the new knowledge is more believable than the actually existing (as in belief revision and belief update), while others consider that new knowledge presents equal credibility to the one already existing (as in KBs arbitration and KBs combination).

In the proposed architecture this last view is adopted, that is, the knowledge coming from the KDDS (new knowledge) has equal credibility to that in the RBS KB (existent knowledge). This seems to be an adequate approach, since it is not realist to consider that the knowledge coming from the KDDS deserves a greater credibility than the knowledge whose origin is from one or more human experts. To consider that the new knowledge, after has received some kind of filtering by the user, is put at the same level of the knowledge that is in the RBS KB seems to be the more rational approach. The term user is used with the sense of expert, or knowledge engineer, because it will compete to one of them, or both, the conduction a KBs fusion process.

The two approaches that give the same credibility to the new and actual knowledge are KBs arbitration and KBs combination. From these, the second was chosen because it is the one that better adapts to the problem of incorporating new knowledge in the KB of the RBS. KBs combination was precisely developed with the purpose of helping in ESs construction, through the union of knowledge fragments coming from several experts and by the construction of the different MCSs.

To assure consistency and correction of a KB is not sufficient to analyse separately the rules that compose it, but also the set of inference chains that it is possible to establish. Some errors just manifest along one chain (for instance, inconsistency along an inference chain) or more inference chains (for instance, inconsistency along multiple inference chains). The adoption of an approach that makes an analysis also at these levels is essential. The knowledge verification through formal methods is a solid and adequate way of doing it.

The present architecture has as basis an extremely simple idea: to assemble all the existing knowledge from the original KBs only in one KB, proceed to its formal verification, to see if errors exist, and, if they really exist, present to the user all the possible solutions (MCSs of rules) ordered by some criterion, that produce a consistent KB, to help him in his decision.

Even considering that each KB has no errors their union could result in a KB that is affected by a set of errors. Detect and solve these errors is a requirement that has to be respected, so that the new KB can be used in the RBS. To detect the errors a knowledge verification tool is used in the architecture.

In presence of the detected errors, the architecture helps the user to select a MCS of rules, where those errors do not occur. Since equal credibility is given to knowledge that comes from the two sources, all the possible consistent and correct ways of accommodating the two kinds of knowledge (under the formal point of view) are identified. The two adjectives (consistent and correct) are used not only in the sense of inconsistencies, but also circularities and ambivalences inexistence (terms used in knowledge formal verification).

Figure 1 presents an overview of the architecture to fuse knowledge from a KDDS with knowledge that is in the KB of a RBS. Each module that constitutes the architecture will be explained in detail in the following subsections.

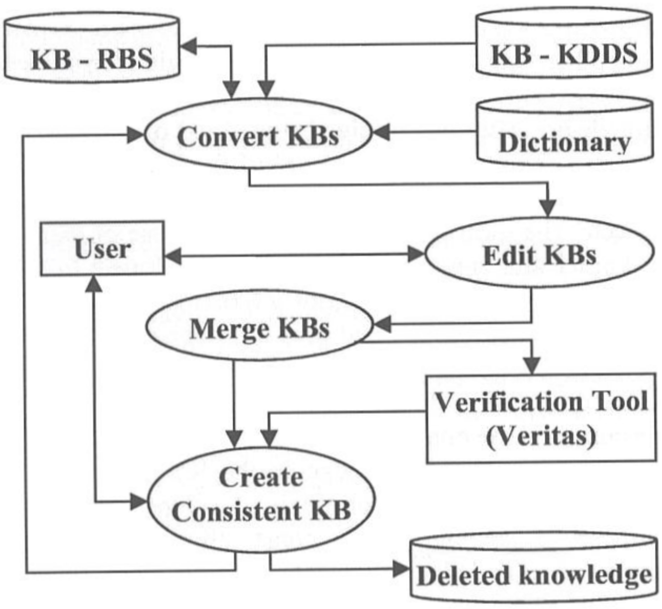

Figure 1- Architecture of the KBs Fusion System

A. Knowledge Bases Conversion Module

KBs conversion (figure 2) is the first step to be made in the presented architecture. Despite the knowledge be expressed in the same representation formalism (rules), it proceeds from two heterogeneous knowledge sources (a RBS and a KDDS), which means that almost certainly is represented under different syntaxes.

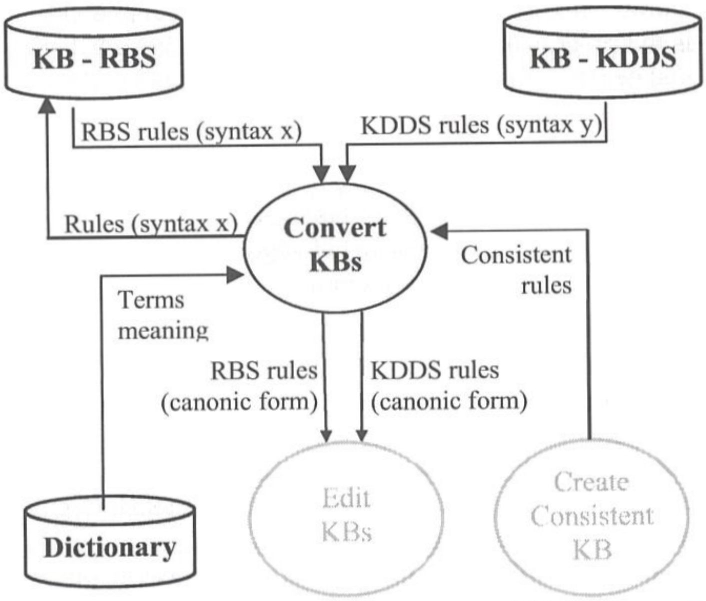

Figure 2- KBs Conversion Module

Not being limited to a single syntax of each of the KBs is one potentiality of this architecture. The application field would be too much limited if the architecture was conceived to a specific syntax. In the RBS KB case this approach even could be admissible, that is, to create an architecture having in mind a specific syntax. In the case of the KB coming from the KDDS, this approach is quite distant from being functional, since depending the KDDS, operation, technique or data-mining algorithm used, KBs with very different syntaxes may appear. Not only having in mind this reality but also to make the architecture flexible to any RBS KB syntax, a module to convert the KBs is considered. The purpose of this module is to put the rules coming from the two KBs, under generic representation syntaxes (named canonic form), recognized by the remaining modules of the architecture.

This module includes a term dictionary that is used to convert the KBs rules that were produced by the KDDS to the canonic form. The rules of this KB may include terms that, although different, have the same meaning to those existing in the RBS KB. Even if the terms are syntactically different, semantically they could represent the same concept. This means that the terms are synonyms. Usually these situations result from discrepancies between the terms used by the knowledge engineer who conceived the RBS KB (for instance: car) and the way that those terms are codified in the databases used as a source for data mining (for instance: automobile). In order to surpass this problem, a dictionary is considered to allow translating the terms used in the KDDS KB to those used in the RBS KB. The terms translation was made in this direction, but it also could have been done in the opposite direction, although that would mean some efficiency loss. The architecture has as final purpose the substitution the RBS KB by the final KB (correct and consistent) resulting from the fusion. However, to make that possible it is necessary to convert the canonic form of the final KB to the original syntax used in the RBS KB, that is, to execute the opposite operation. As a matter of fact this is precisely the other underlying purpose of this module.

B. Knowledge Bases Edition Module

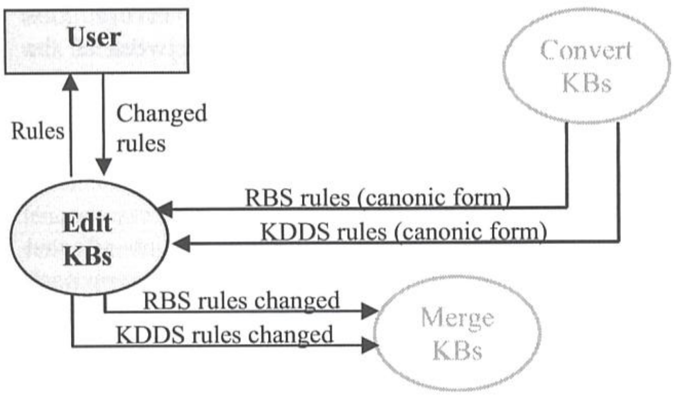

Figure 3- KBs Edition Module

In the architecture, a module that allows accomplishing edition operations over any of the KBs is also included (figure 3). Before KBs union the user has the opportunity to consult each of them, in an easy and comfortable way. Changes, or even rules eliminations, can be made if necessary. Usually, carrying out these edition operations is much more important in the KB produced by the KDDS than in the RBS KB. Through this module the user can visualize the discovered rules and introduce changes in those rules, to make them more generic or specific. Eventually, the user may even want to eliminate one or more rules, as a consequence of having a low degree of confidence or support (depending the data-mining algorithm used this information may, or not, be available), which means that he doesn't give any importance to those rules.

This kind of edition operations can be understood as a filtering and consolidation process of the knowledge that comes from the KDDS, in which the user is the principal intervenient. In relation to the RBS KB, the same edition operations are also available. However, in this case they don't seem so important, since the generality of the knowledge (if not all) is quite solid and accepted by the user. The need of some kind of edition operation is quite unlikely, but can't be excluded at all.

C. Knowledge Bases Union Module

After all the edition rules operations had been executed, the user can unchain their union (figure 4). As the name suggests this module is responsible for carry out the union of the rules coming from the two KBs (already in their canonic forms), which results in the creation of a new KB, named of merged KB. This KB joins the knowledge that proceeds from the KDDS with the one that is in the RBS KB.

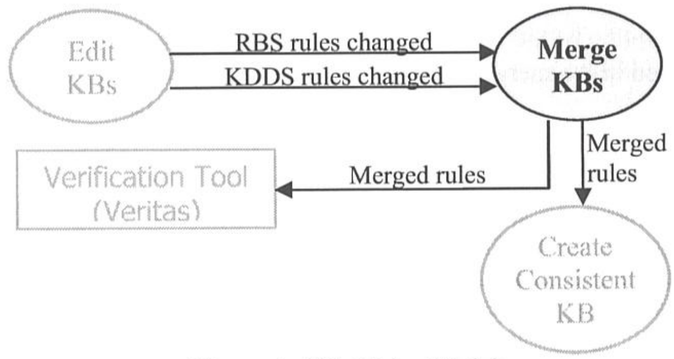

Figure 4- KBs Union Module

D. Knowledge Formal Verification Module

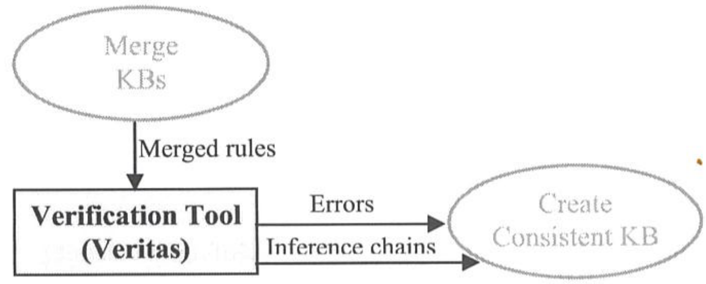

Figure 5- Knowledge Formal Verification Module

Knowledge union from the two sources may result in the appearing of a set of errors like circularities, inconsistencies and ambivalences. The detection of those errors that could exist in the merged KB is the purpose of this module (figure 5). In error detection, knowledge verification techniques based in formal methods are used.

This module has a different characteristic from the others, since it involves integrating in the architecture a verification tool named Veritas [21] to detect errors that could exist in the merged KB. This tool was applied in several domains (for instance, medicine, power systems) and obtained appreciable results. In errors detection, Veritas develops all the plausible inference chains. The theory that is underlying to this technique is the ATMS (*Assumptions-based Truth Maintenance System*) [22], which sustains that the rule antecedent is equivalent to (can be replaced by) the inference chain that originated it. Having as basis a progressive process of substitutions, Veritas produces a set with all the possible developments.

From the set of errors that can be found in the merged KB and that Veritas allows to detect, only those that are of the redundancy type were excluded, since they don't receive any treatment in the actual state of the architecture. Anyway, the existence of this kind of errors is harmless to the KB, that's why less importance was given to them. The existence of redundancies doesn't affect the establishment of any reasoning (inference) process. In some situations, the knowledge engineer intentionally introduces some redundancies in a RBS KB to increase the inference engine efficiency and performance.

In the followings subsections it is made a brief description of the errors that can be identified and manipulated in the merged KB:

(1) Indirect Circularity

A KB contains indirect circularity if and only if has several rules that result in a situation of infinite cycle when of one of those rules is *fired.*

(2) Ambivalence

Ambivalence results from a semantic restriction violation, that depends on the knowledge domain represented in the KB. A semantic restriction constitutes a set of values whose simultaneous existence has no meaning in that knowledge domain (for instance: state(x, moving) $\wedge$ state(x, stopped)). A KB is ambivalent if and only if for a given set of values that don't violate any semantic restriction is possible to infer a set of conclusions that infringe a semantic restriction.

(3) Inconsistency

Inconsistency is a particular case of ambivalence, since is delimitated to logic restrictions violations of the type $\{p, \neg p\}$. On the contrary of semantic restrictions, this kind of restriction is independent of the knowledge domain represented in the KB. A KB is inconsistent if and only if for a given set of values that don't violate the presented logic restriction is possible to infer a set of conclusions that infringe it.

E. Consistent Knowledge Base Creation Module

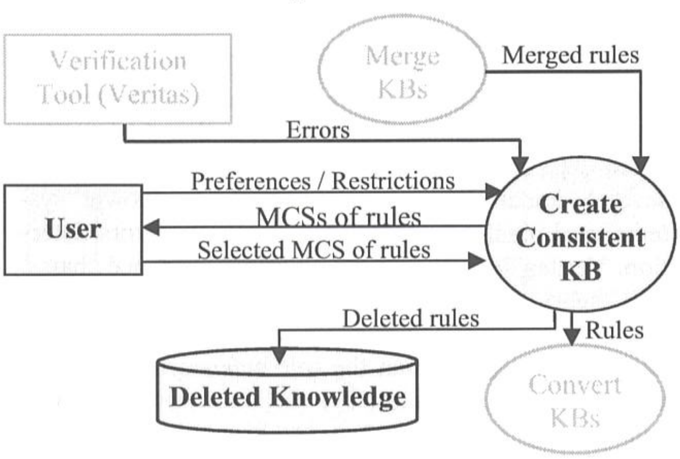

Figure 6- Consistent KB Creation Module

At last, the module that allows the creation of a consistent KB is explained (figure 6). The purpose of this module is to help the user to select a MCS of rules from the merged KB that will be transformed in the new KB of the RBS.

Having as basis the results produced by Veritas, that is, all the inference chains developed and the errors detected on them, it is possible to create the following two disjoined subsets of inference chains:

- **Inference chains affected by errors**. This subset includes all the inference chains involved in some kind of error. From these it is not possible to establish a valid reasoning process. Having as purpose the correction and consistency of the merged KB, it is necessary to prevent that these inference chains can be established. To do so, in each of these, one or more rules must be deleted.
- **Inference chains not affected by errors**. This subset results from the difference between all the inference chains produced and the ones responsible for errors. It represents all the inference chains that are not affected by any kind of error and that allow valid reasoning mechanisms.

This separation between inference chains affected and not affected by errors is inspired in an approach proposed by Lin [23]. In presence of an inconsistent KB, a separation between consistent beliefs (those that are not involved in any inconsistency, which means that they are meaningfully) and inconsistent beliefs (those that are meaningless as a result of being involved in inconsistencies) is made.

To achieve correctness and consistency when errors exist in a KB, some knowledge (rules) in the inference chains affected by errors must be eliminated. Usually there are different possibilities of knowledge elimination, all resulting in a correct and consistent KB. The idea is, precisely, to present to the user all the possible combinations of knowledge elimination that produce a MCS of rules. These possible combinations must appear ordered in function of the loss or impact that the knowledge elimination causes to the inference chains not affected by errors. Naturally that one combination may be more or less harmful to these inference chains than another. Basically, the combinations should be ordered from the one that best allows knowledge preservation (minimal change) until the one that implies the worst knowledge loss (maximal change), according to a given method selected by the user.

The set of combinations can be subject of preferences or restrictions introduced by the user, limiting or imposing the rules to consider to elimination.

The purpose is to help the user to take decisions about the solution to adopt. However, the final decision is a user responsibility. Choosing one of the combinations of knowledge elimination means that this knowledge will be eliminated from the KB, which guarantees that the detected errors will be removed, resulting in a correct and consistent KB. The eliminated knowledge is stored in a repository of deleted knowledge.

V. Conclusions and Future Work

An architecture that permits the fusion of a KB (under the rules form) proceeding from a KDDS with that one of a RBS is presented.

As the knowledge fusion problem in matter is defined generically, that is, it is not for a specific syntax of any of the two KBs (that would be too much restrictive) an architecture that is open and independent of the KBs syntaxes is proposed. Since the architecture is modular, it is easy to incorporate the proper module to effectuate the conversion of a KB syntax to that used in the architecture. Not only independence in relation to knowledge representation syntaxes is achieved, but also in relation to knowledge domain that is modeled in the KB.

The correctness and consistency of the knowledge resulting from the fusion process is a requirement that was initially placed to the architecture. To assure it, a knowledge verification tool based in formal methods is used in the resulting KB. An originality of this work is the concern with other kind of errors beyond the inconsistencies, as is the indirect circularities and ambivalences case. Solving the inconsistencies problem has centralized all the investigation efforts, as can be seen in the literature. Although this kind of error be the most difficult to solve, there are other errors that could affect seriously a KB, like the ones previously mentioned. Since the resulting KB from the fusion is going to substitute the RBS KB, it is necessary to assure not only consistency but also knowledge correction. This means that errors like circularities and ambivalence can't be ignored, since they threaten the establishment of inference processes, degrading in this way KB quality. Although the architecture had been proposed for the fusion of knowledge coming from a KDDS with that of a RBS, their characteristics allow it to be applied in any generic situation of addition or incorporation of new knowledge in the KB of a RBS.

Several aspects related to this architecture will be object of future developments. Between these, the followings deserve a special word.

- To specify the methods that will allow ordering the knowledge elimination combinations that result in MCS of rules. These methods should order the combinations from the one that best allows knowledge preservation until that which implies the worst loss.
- To consider, in some way, the deleted knowledge in the successive fusion processes, giving a meaning to the deleted knowledge repository that is considered in the architecture. Some of this knowledge may even be valid, but was eliminated to allow a MCS of rules.
- To implement the proposed architecture in a computational tool.
- To apply the computational tool to real cases of KBs fusion in order to validate the proposed architecture.

VI. References

[1] U. Fayyad, G. Piatetsky-Shapiro, P. Smyth and R. Uturusamy – *Advances in Knowledge Discovery and Data Mining*. Cambridge: AAAI/MIT Press, 1996.

[2] F. Koriche – Modelling Knowledge Merging Systems. In R. Wagner (Ed.) – *Proceedings of the 8th International Workshop on Database and Expert Systems Applications*, Toulouse. IEEE Computer Society Press, 1997, p. 562-567.

[3] L. Cholvy and A. Hunter – Information Fusion in Logic: A Brief Overview. In D. Gabbay, R. Kruse, A. Nonnengart and H. Ohlbach (Eds.) – *Qualitative and Quantitative Practical Reasoning, Proceedings of the 1st International Joint Conference on Qualitative and Quantitative Practical Reasoning*, Bad Honnef. *Lecture Notes in Computer Science*. Springer, 1997, vol. 1244, p. 86-95.

[4] C. Baral, S. Kraus, J. Minker and V. Subrahmanian – Combining Knowledge Bases Consisting of First-Order Theories. *Computational Intelligence*, 8:1 (1992) p. 45-71.

[5] J. Halpern and Y. Moses – A Guide to the Modal Logics of Knowledge and Belief. In *Proceedings of* the *9th International Joint Conference on Artificial Intelligence*, Los Angeles. Morgan Kaufmann Publishers, 1985, p. 480-490.

[6] C. Alchourrón, P. Gärdenfors and D. Makinson – On the Logic of Theory Change: Partial Meet Functions for Contraction and Revision. 50 (1985) p. 513-530.

[7] R. Fagin, G. Kuper, J. Ullman and M. Vardi – Updating Logical Databases. *Advances in Computing Research*, 3 (1986) p. 1-18.

[8] H. Katsuno and A. Mendelzon – A Unified View of Propositional Knowledgebase Updates. In *Proceedings of* the *11th International Joint Conference on Artificial Intelligence*, Detroit. Morgan Kaufmann Publishers, 1989, p. 1413-1419.

[9] M. Dalal – Investigations into a Theory of Knowledge Base Revision: Preliminary Report. In *Proceedings of the 7th National Conference of the American Association for Artificial Intelligence*, Saint Paul. MIT Press, 1988, p. 475-479.

[10] P. Revesz – On the Semantics of Theory Change: Arbitration Between Old and New Information. In Proceedings *of the 12th ACM SIGACT-SIGMOD-SIGART Symposium on Principles of Databases*, Washington. ACM Press, 1993, p. 71-92.

[11] J. Lin. and A. Mendelzon – Knowledge Base Merging By Majority. In R. Pareschi and B. Fronhoefer (Eds.) – *Dynamic Worlds: From the Frame Problem to* Knowledge *Management*. Kluwer Academic Publishers, 1999.

[12] C. Baral, S. Kraus and J. Minker – Combining Multiple Knowledge Bases. *IEEE Transactions on Knowledge and Data Engineering*. 3:2 (1991) p. 208-220.

[13] P. Gärdenfors – Knowledge in Flux: Modelling the Dynamics of Epistemic States. MIT Press, 1988.

[14] P. Gärdenfors (Ed.) – *Belief Revision. Cambridge Tracts in Theoretical Computer Science*. Cambridge: Cambridge University Press, 1992, vol. 29.

[15] P. Liberatore and M. Schaerf – The Complexity of Model Checking for Belief Revision and Update. In Proceedings of the 30th National Conference on Ar-

tificial Intelligence and Eighth Innovative Applications of Artificial Intelligence Conference, Portland. AAAI Press/The MIT Press, 1996, vol. 1, p. 556-561.

[16] P. Peppas, A. Nayak, M. Pagnucco, N. Foo, R. Kwok and M. Prokopenko – Revision vs. Update: Taking a Closer Look. In W. Wahlster (Ed.) – *Proceedings of the 12th European Conference on Artificial Intelligence*, Budapest. John Wiley and Sons, 1996, p. 95-99.

[17] H. Katsuno and A. Mendelzon – On the Difference Between Updating a Knowledge Base and Revising It. In P. Gärdenfors (Ed.) – *Belief Revision*. Cambridge: Cambridge University Press, 1992, p. 183-203.

[18] M. Winslett – *Updating Logical Databases*. Cambridge: Cambridge University Press, 1990.

[19] H. Katsuno and A. Mendelzon – Propositional Knowledge Base Revision and Minimal Change. *Artificial Intelligence*. 52 (1991) p. 263-294.

[20] V. Subrahmanian – Amalgamating Knowledge Bases. *ACM Transactions on Database Systems*. 19:2 (1994) p. 291-331.

[21] J.Santos, C.Ramos, Z.Vale, A.Marques – Veritas, An Application for Knowledge Verification. In *Proceedings of 11th International Conference on Tools with Artificial Intelligence*, Eds. IEEE Press, Chicago, Illinois, 1999, p.441-444.

[22]J. Kleer – An Assumption-Based TMS. *Artificial Intelligence*. 28 (1986) p. 127-162.

[23] J. Lin – Consistent Belief Reasoning in the Presence of Inconsistency. In *Proceedings of 5th Conference on Theoretical Aspects of Reasoning about Knowledge*, Pacific Grove. Morgan Kaufmann Publishers, 1994, p. 80-94.