Foundational Challenges in Automated Data and Ontology Cleaning in the Semantic Web

J. Antonio Alonso-Jiménez, Joaquín Borrego–Díaz*,
Antonia M. Chávez-González, F. Jesús Martín–Mateos
Departamento de Ciencias de la Computación e Inteligencia Artificial
E. T. S. Ingeniería Informática, Universidad de Sevilla
Avda. de Reina Mercedes s.n., 41012–Sevilla. Spain.
Keywords: Data Cleaning, Semantic Web, Certified Reasoning

Abstract: *The application of automated reasoning systems to data cleaning in the Semantic Web raises many challenges on the foundational basis of cleaning agent design. The authors discuss some of them. They finally argue that logic trust in the Semantic Web can only be achieved if it is based on certified reasoning.*

Nowadays data management in the WWW needs tools to ensure secure and trustworthy performance. The utopian future shows a *Semantic Web* (SW) providing frameworks where some actual problems about data are solved by the logical trust that SW establishes. Though for the present time there are some problems to solve. Realistic inmediate future raises several challenges, some of them dealing with fundational issues of Semantic Web, the abstract (ontological) definition of data, and the work with incomplete or provisional ontologies through the evolution from actual WWW to SW (as well as the evolution of the ontologies itself). In any case, the pair *data-ontology* will be an indissoluble marriage and represents the knowledge bases (or *Knowledge DataBase*, KDB), aim of study in the SW framework. We want to analyse the role that an *Automated Reasoning System* (ARS) may play as assistant for cleaning KDB's.

The reader is warned that the challenges (rather questions that imply challenges) on which we are interested are intentionally oriented to problems about logical reasoning and its robustness for data cleaning and preprocessing, focusing them on qualitative KDB. Of course, there are other fields where looking for answers. Moreover, we emphasize the problems found in the first phase of data cleaning, namely *data analysis*.

Some of the ideas we discuss are suggested by experiments about data cleaning with the assistance of an automated theorem prover (ATP) [1] (see figure 1) that serve us to show the problems as well as to suggest solutions. The experiments carried out on a KDB in a paradigmatic field, where ontology design is a challenge itself: the Qualitative Spatial Reasoning (QSR).

In these experiments, we use the First Order Logic (FOL) as language, focusing in this way in foundational problems, and forgetting representational issues for a second stage. A powerfull ATP based on resolution, OTTER[1] was used as assistant.

The long term goal for data cleaning in SW might be the design of general purpose *cleaning agents*: intelligent agents capable to find and repair anomalies in KDB, both in the ontology and in the data. To achieve this goal, we need to analyze first the challenges that it raises. For the present, the ARS only plays the role of an assistant in an ARS-aided methodology to clean anomalies in a KDB [2], as a first step towards the design of cleaning agents. The overall question that it adresses is whether it is possible to design trustworthy data cleaning systems to certify both the KDB and the self reasoning on it, as SW promises. This challenge emphasizes the current needing of an explanation of the reasoning behind cleaning programs [17].

What is the logical complexity of the problem?

Cleaning a KDB in a dynamic environment as SW is a quite hard task. Some of the reasons are the following:

- We can not suppose that the KDB is finished (because the user would add new facts in the future). Thus, even if we have a good KDB, the dificulties will begin again with the future introduction of data, persisting in this way the existence of anomalies.

- Usually the *intensional theory* of the KDB (i.e. the ontology) is nonclausal. Thus, it is highly possible that classical axiomatization of database theory becomes inconsistent. Nevertheless, the self database represents a real model.

- The KDB does not contain facts about all the relations in the language. It is usual that only an almost complete information about primary relations and concepts appears, those that one could consider as the *primary ones*. Other complex notions have to be deduced.

*Corresponding author. E-mail: jborrego@us.es

[1] `http://www-unix.mcs.anl.gov/AR/otter/`

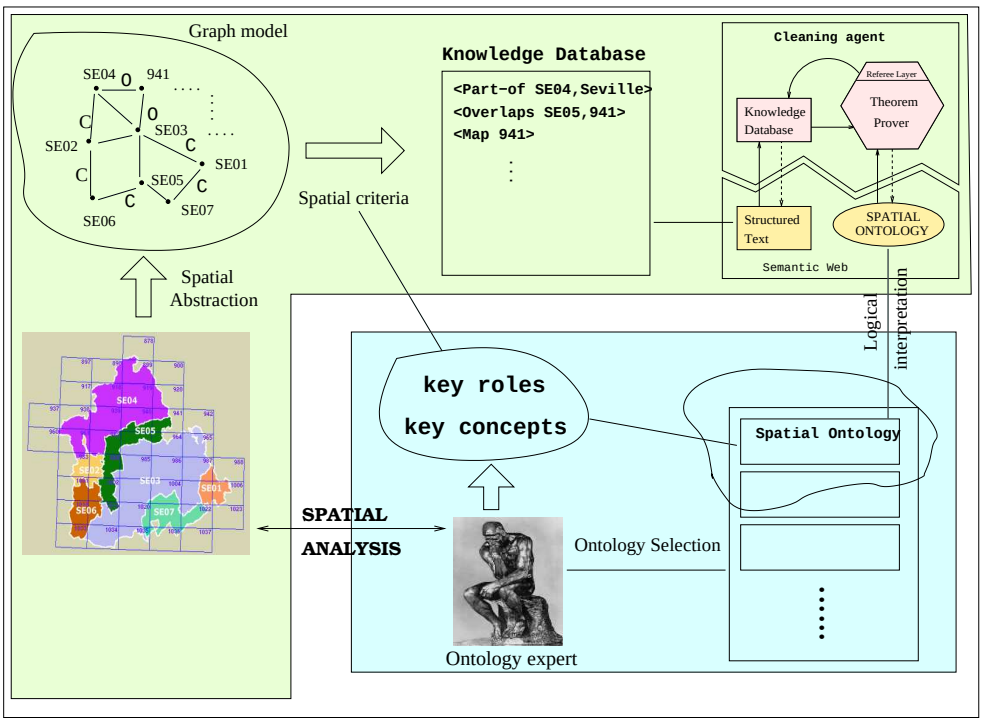

Figure 1: Ideal scenario for a specific-pourpose cleaning agent

This list of reasons must be accomplished by a necessary revision of the verification and validation of Knowledge Based Systems problem [4].

On the other hand, the anomalies we can find come from several sources. A first classification is the following:

- The set of data may be inconsistent with the ontology, due to formal inconsistencies produced by wrong data, or the absence of some knowledge.
- The database does not give us complete information about primary predicates (the user could introduce more data later). For example, the theorem prover may deduce, from the database, the existence of objects without name.
- Unexpected disjunctive answers to standard queries (a logical deficiency).
- Inconsistency in the Ontology.

Also, it is necessary bear in mind, when we work with many information sources, that for focusing query plans on selected sources we need to estimate the quality of the answer, task that can become a critical issue [38].

A characteristic of the problem is that it forces us to keep the balance between real-time processing and complex reasoning. This dilemma is, in essence, the same as reactivity versus proactivity attitudes in agents. Complex logical features could need more autonomy from ARS assistant to work with. So it is not advisable to tightly bound ARS's work time. Nevertheless, it is not clear how much autonomy degree must it have. The reason is that an ATP can produce an *overspill* of new knowledge, not all of it interesting for the problem. Thus it needs a strict *referee* layer to manage the workflow. This is not a new idea in automated deduction (see [12]). These are the main drawbacks for the integration of an ARS in agent's architecture. However, the loss of real-time requirements might not be important: the system could work as a *night cleaning service*, debugging metadata implemented by the users during the idle-time of the computer.

Another interesting aspect is that some languages to specify ontologies as KIF[2] allow us to design sintactly complex theories. In this case the formal paradigm can be hard to manage. Intuitively complex axioms in ontology description often come from an incomplete or inconsistent set of concepts and relationships, or a deficient set of explicit relationships. A paradigmatic case is the following: in a company several programmers are concurrently developing the ontology and associated tools (inducing them from its self practical experience), and others concurrently introduce the data. The cooperative work may produce inconsistencies (due, for example, to programmer's wrong interpretations of some concepts in the ontology) or complex axioms that makes the ontology messy (a designer does not know any concepts which can simplify definitions). Other example is data warehouses. Finally, in some cases there exists a lack of mechanisms to soundly evaluate the ontology engineering task. This is an obstacle to the use in companies [18].

On the other hand, providing metadata is a dificult task that the users tend to avoid: to save costs, the

[2] http://logic.stanford.edu/kif/kif.html

initial ontology may have been built by an ontology learning system, but then we have to debug it. Even it is necessary an ontological analysis of the intended meaning of the elements of the ontology (as [21]). In any case, it is necesary to know how to work with poor (or provisional) ontologies.

How to work with poor ontologies?

The problem is almost -but not wholly- one of logical reasoning with weak theories. We assume that the content on SW is (explicitily or implicitily) in the form of logical formulas. From this viewpoint, we think that the use of ontologies in data management implies a logical reasoning deeper than the one that so far we believe as advisable. An ontology is not only a logical theory, but something with aditional features as backward compatibility [22]. However, one must accept that logical trust provides a first security criterion.

In a logical setting, data are ground terms constrained by the ontology language, and, more important, whose behaviour is specified by the ontology. For example, a partially built ontology, constraint us to reason with a poor language or a deficient set of axioms.

The reasons to the persistance of unstable ontologies are several. Most of what the research do is related to the development of ontology representation paradigms and it has not a similar counterpart in other features associated with the evolution as persistence, transactions or scalability [30]. As widely constated, the effort to build a *robust* ontology, including the large body of information of a company in the universe of metadata, is expensive. So the investment must be promoted in early stages, impeding significant changes later.

In ATP-aided cleaning tasks, an interesting type of reason for *bottom-up change generation in ontologies* [30], is in this framework the *Skolem noise*, due to the analysis of track interaction among KDB, the automated theorem prover and the user (figure 2), when we work with provisional ontologies.

Can we wait until the ontology becomes stable? The answer is no. Not at least it is clear the meaning of stable, or better, *robust*.

When is robust an ontology?

The ontologies are presumably designed to protect us from wrong management of information. But can we be sure of the current ontology? Is it possible to predict that only minor changes will be applied? Ontologies need to be maintained just like other parts of a system (for example, by refinement [3] or versioning [27]) The definition of robustness for ontologies has several perspectives, all of them necessary but none sufficent.

From SW perspective, the desiderata for a *robustness* notion come from some of the requirements to exploit SW ontologies, mainly in two of them [22]:

- *SW ontology should be able to extend other ontologies with new terms and definitions.*
- *The revision of an ontology should not change the well-foundness of resources that commit to an earlier version of the ontology.*

Nevertheless, even being able to be extended, it is evident that the *core* of the ontology should be stable. By *core* we understand a portion of the source of the ontology that we consider as the sound representation of a theory with well known properties, accepted as the best for the concepts contained in the ontology. Usually, it is advisable that the top of the ontology (general concepts) is included in the core, preferably choosing a standard proposal for it, as could be SUMO[3].

From a logical point of view, *robust ontology* should mean *complete logical theory*, and this definition might be applied in the context of OWL[4] full language, or in any case, it may be useful to some coherent parts of an ontology. However, this is not a local notion: minor changes commit logical completeness in a dramatic way. Other logical notions, as *categoricity*, crash with natural logical principles for reasoning in databases as Closed World Assumption or Unique Names Principles.

Therefore, robustness should be a combination of both perspectives, jointly with a notion of *clear* ontology, as opposite to messy ontology. A messy ontology may be, in the future, aim of a cleaning process, suggested by the diary management. A definition of robustness may be the following:

> *An ontology is robust if its core is clear, stable (except for extensions), every model of its core exhibit similar properties w.r.t. the core language, and it is capable to admit (minor) changes made out of the core without commiting core consistency.*

By *similar properties* we understand a serie of metalogical properties that bring us to the conclusion that all these models are, in essence, the same: they can not be distinguished by means of amenable properties.

But the robustness can not be understood without considering dynamic aspects. Mostly, we think the robustness as a measure: ontology evolution extends the core to a big portion of the ontology leaving out almost only data. This evolution schema allows us to locate the possible inconsistencies in the data, approximating in this way the problem to the classical integrity constraint checking.

In the example of figure 2, Skolem noise phenomenon suggest us to add to the ontology (even to the core theory used in [1], which is the *Region Connection Calculus*, RCC) a geometrical interpretation of `$f1` as the intersection of regions (when they intersect).

The dynamic of ontology evolution brings us to adapt other ontologies to extend ours, or generally, to design systems for the management of multiple and distributed ontologies (as [31]). This question leads another one: how to design sound ontology mappings for automated reasoning?

[3] `http://ontology.teknowledge.com/`
[4] http://www.w3.org/TR/owl-features/

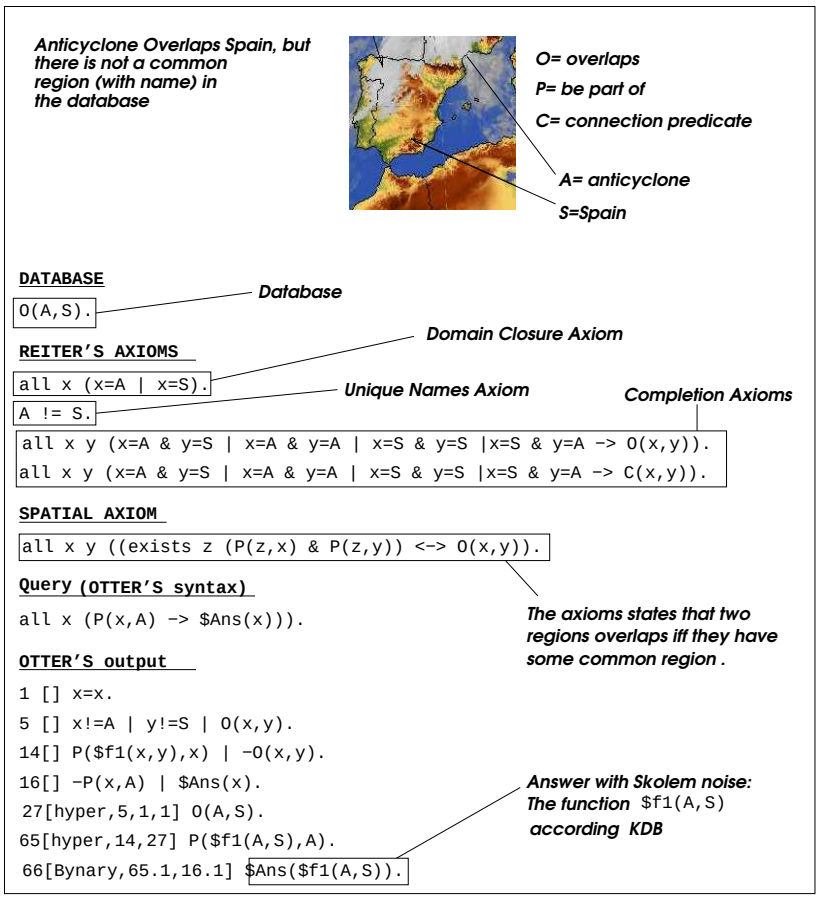

Figure 2: A naif example of Skolem noise obtained by the system OTTER, reasoning with a poor ontology

Ontology mapping means interpretation among logical theories?

Semantic heterogeneity of ontologies is a major barrier to cleaning data. Ontology mapping will become a key tool in heterogeneous and competitive scenarios as e-commerce and the knowledge management in large organisations. This technique will play -in the SW- a key role in the usual Extraction-Tranformation-Loading (ETL) process (see [39]) for data cleaning. However, it is not clear which are the logical and cognitive consequences that an ontology mapping has for automated reasoning. Logical perspective gives us again a rigid concept, the *interpretation among theories*, that is the most suitable for logical automated reasoning, but this approach has limited applications. In practice, a solution can be the use of contexts (or microtheories), as in CyC[5] ontology. There also exist proposals that may be useful to control the expansion of anomalies to the complete ontology, as the *contextual* OWL [8] (see also [32]).

There exists a limit for ontology mapping. Intelligent KDB analysis aided by ATP will need in some moment a logical interpretation of basic data as integers (by example, when we accept that *number of parts* is a positive integer). It is not easy to build an ontology on numeric data and their properties. Even if we have such one, the ontology mapping can be understood as a logical interpretation of an arithmetic theory (recall that, in OWL, mappings are not part of the language). Since interpretation means, up to a point, to incorporate a logical theory about such data to the target ontology, logical incompleteness is not only assumed, but unavoidable (recall also that, though OWL integrates data types, there is nothin about integer arithmetic, for example). This phenomenon can be tamed by syntactical restriction of queries, but the solution will be hard in any case: extensions of OWL (thinkig it as a logical theory) will add features for numerical data representation and reasoning. Moreover, undecidability will be intrinsic to any language for rules with powerful features, so this also happens for more complex tools. This is a definitive barrier to ontology language design. We have to find a way to escape of that in practice.

Model Theory for Semantic Web?

At the top of the SW cake proposed by Tim Berns-Lee, *Logic* and *Proof* appear as the bases of *Web trust*. Logical trust, in a broad sense, is based in logical semantics, and logical semantics deals with models and the definition of truth. In the ontology design task, model theoretic analysis is often forgotten, because ontology designer has in mind a particular model (the real or

[5] http://www.cyc.com/

intended model). A well known principle in Knowledge Representation states that no language nor KB exists to represent faithfully the intended world where we want to work; that is, unintended models exist (see figure 3). In QSR, this principle turns into the *poverty conjecture*: there is no purely qualitative, general purpose kinematics. Therefore no categorical ontology exists for this field.

One of the goals of logical Model Theory is to study all the models of a theory, including nonintended models. Model Theory is not only a semantic basis for good specification [15]. It has impact on practical proving: the existence of unintended models is consequence of incompleteness and vice-versa. For example, in the figure 3 the existence of a model where $\mathtt{A} = \mathtt{S}$ implies that the theory RCC $+\{O(A, S)\}$ does not entail that the region affected by the anticyclone and Spain are different.

Again, Model Theory may be combined with other features, especially of linguistic nature. In this way the designer may specify non-logical information. However the amount of linguistic and lexicographic problems that this option supposes warn us to explore carefully the combination of logic with (mathematical) linguistic. On the other hand, it is not strange to face up with inconsistent information in a KDB in the Web, and in this case classical model theory has nothing to do: there is no models.

How to reason with inconsistent information?

As larger is a KDB, smaller will be the possibility of it being consistent. We may say that logical inconsistency, one of the main sources of untrust, is frequent whenever the KDB has big amount of hand-made information (and whenever it is a relatively interesting part of WWW). Big KDB escapes for any model search (consistency checking) system. Thus, consistency analysis will be weakly solved by a semidecision procedure: if a refutation is founded, KDB will be inconsistent, and thus inconsistency turns out to be the main anomaly (see the figure 4).

If we accept to work with inconsistency, the goal is to design logical formalisms that limit those that can be inferred from inconsistent KDB. An approach to the acquisition of trustworthy information consists in to *argue* the information extracted from data. Argumentation is a succesfully strategy in this scenario.

An argument according to a KDB is simply a pair $< \Gamma, fact >$ where Γ is a subset of the KDB that proves $fact$. An argument with unacceptable conclusion may be considered as a report about an anomaly, and the cleaner must find the reason of it. Any anomaly warns us of a mistake, but the expert decides which type of arguments have to be analysed. However, the ARS offers more arguments than human s analysis can study. Though a referee layer for ARS's output could partially solve the overspill of arguments produced by the system, we have to design very efficent criteria to discard arguments. For the spatial example, the absence of Skolem noise in an argument makes it consistent [1], thus a quick look at the content of the argument reveals its consistency. But it is not easy for other anomalies. There is an interesting hierarchy of arguments that estimates -at least theoretically- their robustness [14]. This hierarchy is based on FOL notions as consistency and entailment among others.

Is FOL the universal provider for formal semantics?

For tasks as verification of KDB, FOL provides a formal framework where some anomalies in specifications can be defined. Ontology languages as DAML+OIL [6] are actually verified and revised by translating specifications into FOL and then applying an ATP [16]. Reasoning services for such class of ontology languages, based on its relationship with Description logic (a subset of FOL) are also investigated [24]. FOL is selected as translation target because in this way we have strong reasoning methods for several sublogics expanding the ontology language, as well as we have an ideal framework where we can expand the expressivity of the language itself. Extensions of classical predicate logic have been proposed, as *F-logic* [41] that naturally solves problems as reification, a feature of RDF language.

Modal logics for beliefs lift the reasonning to mental attitudes. Mental attitudes does not seem adequate to be add to ontology reasoning, but there exist other intensional operators as *Provable from Ontology O*, used in logic programming for meta-reasoning, which may be interesting to be considered. But how? For the present, it is often to think in the ontologies as often thought as *static theories*, that is, a set of axioms, separated in this way of reasonning (or the set of rules).

This option reduces the problem to investigate which rule language is the best for each propose [23]. But it does not seem adequate for verification of KDB. It is advisable this limitation is solved for KDB cleaning tasks, and not only by action of *external* reasoners: an ontology should *ontologically define its reasoning framework*, or better, the ontology should be a *potential reasoner* itself. We will return later to the question to succinctly explore this solution.

Is there an ontology for untrustworthly or fraudulent information?

We have commented that the existence of an argument allows us to locate and repair an anomaly -if it is the witness of such one- by means of expert analysis of its content. A deep analysis of the arguments reported by

[6] http://www.w3.org/TR/daml+oil-reference

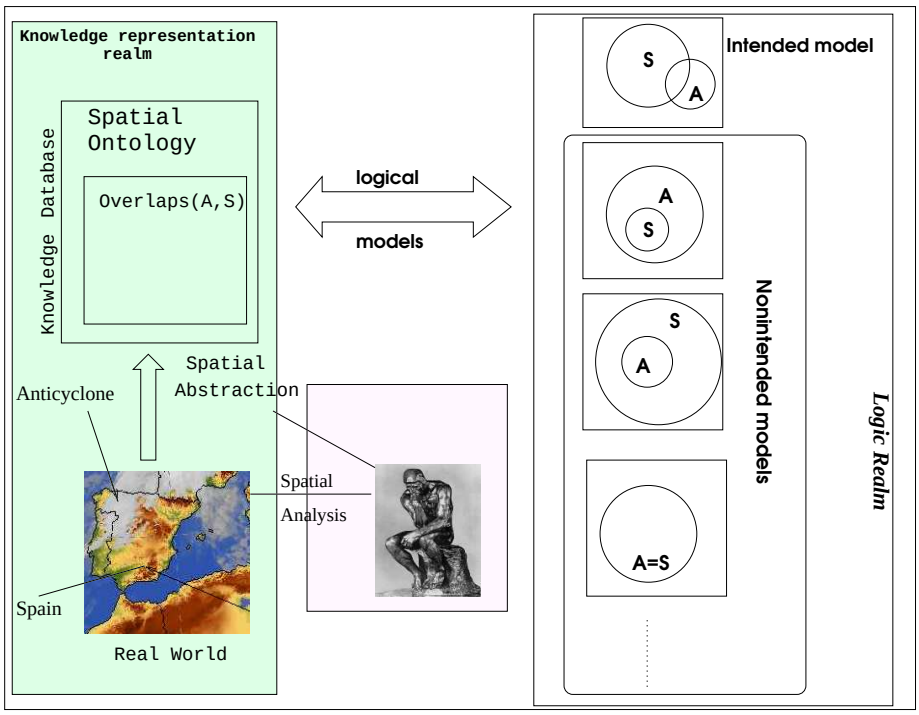

Figure 3: Unintended models for a poor KDB (with the natural notion of `Overlaps` role)

an ARS classifies them according to its trustworthyness. So, an argument hierarchy is a first step towards an ontology of trustness. But we also need to clarify operational features of the ARS assistant. The autonomous behaviour that we expect of the automated argument searcher can be slanted. Much work about this topic could be accomplished the background from automated reasoning field.

More arduous will be the task of recognizing fraudulent information; it represents a significantly different problem. If the ontology represents a fraudulent world (a unintented model of the user's knowledge particularly dangerous for his/her pourposes) the arguments reporting fraudulent information do not show anomalies. To make matters worse, fraudulent ontologies may use linguistic features to hide information. This leads the question out of our interest: fraudulent, by definition, can not have logical legitimacy. A solution may be the combination of user's trustworthiness on others users (see [40]). One can believe, too, that the problem can defeated with a fine analysis of the problem which deal with mental attitudes as *beliefs, desires or intentions* to specify the phenomenom for logical data-mining agents.

Is it advisable to add mental attitudes in data mining/cleaning?

In an scenario with inconsistent information and/or several cleaning agents working, it is necessary to add an intentional level to the logical representation? We have already commented that it not advisable to add mental attitudes to the ontology, but in a multiagent system setting, the specification of knowledge extracted by agents from inconsistent information is unavoidable. In fact, standards for agent communication languages as FIPA-ACL[7] use mental attitudes in the specifications. A *multiagent data-mining* system should manage this kind of information.

A way to avoid the adoption of mental attitudes when one works with simple information items (as structured news reports) might be the implementation of a set of fusion rules to solve agent's disappoinments based on the nature of information (see the survey [6]). This option may need to preprocess data from heterogeneous databases. In other case, data need to be extracted and translated (by ontology mapping) during cleaning runtime, making difficult to achieve acceptable time responses, as occurs in classical data cleaning. But the preprocessing in KDB cleaning has other need: it is possible that the system fires some rules to add new data, to transform the data source of the KDB in a consistent instance of the KDB target.

Is it possible to preprocess data with respect to an ontology?

An interesting aspect of cleaning proccess appears when the KDB programmers decide which are the main roles and concepts of the ontology when they focus the cleaning (*spatial abstraction step* in figure 1). Does Skolem noise mean that we do not know the ontology implicitly or unconsciously used by KDB builder? Not, in general. It only might suggest the need to preprocess the knowledge, sometimes applying simple rules associated with the ontology. An option to preprocess full

[7] http://www.fipa.org/specs/fipa00037/SC00037J.pdf

The Main Logical Anomaly: Inconsistency

The main drawback of many of the methods for inconsistency handling is the computational complexity. On the other hand, the relevant role of ontologies in KDBs design for the Semantic Web needs of a revision both clasical semantics for databases (which are oriented to identify correct answers and logical consequences) and the self definition of consistency (basically focused on integrity constraint checking). Some of the recent proposals to inconsistency handling are the following:

Paraconsistent logics:
Paraconsistent logics attempts to control the undesirable (logical) consequences from an inconsistent database, taming the logical calculus. See [25] for a general introduction, and [19].

Non-repairing methods:
The idea behind these methods is to preserve the source information and decide the trueness by a general analysis of the retrieved knowledge. Some recent examples are:

- To establish Pre-orders on information sets (see e.g. [9], or [33], where the inference is bounded to select belief subbasis of the KDB, in the paraconsistent pradigm).
- *Argumentative hierarchies*, that classify the arguments according to robustness with respect to other subsets of the KDB (see e.g. [14]) or their relationships with other arguments (for example, the argumentative framework based in the defeating relationship among them [13]). See [37] for an application to databases.
- A classic idea in Artificial Intelligence, *contextualization*, can be used in the new framework (contextualizing ontologies [8] and data [32]).

Merging-oriented techniques:

- To establish rules that allow to consistently fuse jointly inconsistent information (see [6]).
- Solutions provided by the theory of merging databases [11].

Measuring anomalies:
A good option is to provide tools to estimate the anomaly, for example:

- Evaluating by means of a paraconsistent logic [26].
- Measuring inconsistent information by means of *measures for semantic information*. These measures estimate the information that the KDB contains on the model which it represents. There exist measures for working with inconsistencies [28].

Repairing techniques:
Mainly, there exist two drawbacks for KDB repairing: the complexity of the revision of the full KDB and the possible discarding of inconsistent data which are potentially useful. So, it seems advisable the repairing is focused on relatively small datasets. The revision of ontologies is an essentially different task, because it represents a key organization of the knowledge of the owner and, as every logical theory, minor changes may produce unexpected and dangerous anomalies. Some options are:

- Correcting records: the Fellegi-Holt method, used in the past in many government stadistic bureaux, can be reviwed as a safe logical method [7]. The method is based in the searching of *all deducible requirements* to decide which fields must be changed to correct an incorrect record.
- Once the notion of database repairing is defined in a logical form, the *entailment component* of the definition can be simulated by calculus; for example by tableaux method [5].
- Consistent querying to repair databases: Te self answer drives the reparation. For example, splitting the integrity constraints according to the character of the negation which involves each constraint, it is possible to produce consistent answers and to repair the database [20].
- Consistent enforcement of the database. The aim is to systematically modify the database to satisfy concrete integrity constraints. A promising method consists in using greatest consistent specializations, adapting the general method (which may be undecidable) [29].

Consistent answering without repairing:

- A good option may be the transformation of the self query to obtain consistent answers [10].
- Using paraconsistent inference (see e.g. [33]).

Preserving consistency:
The idea is to update the KDB preserving the consistency. The classical notion of consistency (satisfaction of integrity constraints) must be expanded due to the relevant role of the ontologies. It is also necessary to prove the consistency of the method, and some sort of completeness (see e.g. [36]).

Figure 4: A brief overview about inconsistency handling

databases is to extend them with new data produced by means of the application of simple rules, designed from the ontology [1]. For example, in the context of figure 2, it might add to a RCC-database the fact $O(A, B)$ to the database if an exploration found two facts $\mathtt{P}(\mathtt{C}, \mathtt{A})$ and $\mathtt{P}(\mathtt{C}, \mathtt{B})$, for any regions $\mathtt{A}, \mathtt{B}$.

A proposal: towards ROWL language

We do not want to finish without suggesting a solution to solve, in the future, some of the above challenges. We think that the best way might be the design of a language that could be called ROWL (a Reasonable Ontology Web Language). The idea borns from the need of attaching (specialised) certified logical inference to the ontology; that is, ontological information about how to reason with the information. To make this, we should add new features to OWL for specifying what type of reasoning and which ARS are advisable to reasoning with the OWL ontology (thinking also that the ontology is optimized to reasoning with them). We do not describe here a formal language, we can give an impression of how that language might look like in early releases of its development, and how it could satisfy some of the challenges.

A closer look to the design of ROWL

The idea is suggested by the building of *Certified Generic Frameworks* (CGF) where the design of a certified *ad hoc* reasoner is simplified to provide some key features of the ATP , that ontology must supply [34]. An ontology that accepts a reasoner that fits in this framework only has to specify simple elements: computation rules (to drive the deduction), representation rules (to normalize formulas), measure function on formulas (to deduce the halting of deduction methods) and model functions (to supply models in some basic cases). The framework has been programmed in ACL2[8], which is is both a programming language in which you can model computer systems and a tool to help you proving properties of those models. In this way the ATP obtained is sound and complete, and formally verified by ACL2. A detailed presentation of the CGF will appear in [35], where the formal verification of the framework is described. In this way the ATP obtained is sound and complete, and formally verified by ACL2. In the case of [35], the framework -which runs, because it produces a COMMON LISP program- has been designed to synthesize the most important component in many AI systems based in automated reasoning, the SAT provers.

In fact, ROWL language should only have features to show links where those elements explicitly appear, allowing the refinement (or even the change) of the deduction method by changing slightly the KDB. In this way,

$$\text{ROWL} = \text{CGF} + \text{OWL}$$

A generalization of this framework to others ARS will be an interesting task. Of course, previous to decide what are the key features of ROWL, it is necessary to design an ontology about automated reasoning. There exist some projects to build such an ontology, for example, within *MathBroker*[9] and *MONET*[10] projects. At the moment, the performance of the synthesized SAT provers is far from the results that can be obtained using any sate-of-art SAT-provers. But it is a next aim to obtain better provers using more efficent data structures.

What advantages would have ROWL?

The definitive advantage of a language as ROWL consists in that it allows to design general-purpose cleaning agents, as opposed to specific purpose cleaning agents based on wrapper technology. Such agent only would adopt, as *thinking component*, the ATP synthetised from the information of the ROWL ontology (across the links), and the behaviour scheme of the agent (see figure 5), where modules to repair anomalies can be implemented (for example, the translation into logic of methods to correct anomalies as Fellegi-Holt's one [7]). This proposal is only the first step, a lot of research remains to be done. The first at all, a complete analysis of the relationships between ACL2 logic (the logic of the theorem prover used to build the CGF; a computational logic closer to a quantifier free FOL accepting some amount of induction) and Description Logics. But with this sketch, how does this proposal solve some of the challenges above? Basically, giving standards to represent aditional information not explicit in data associated with the ontology:

- If we assume that we are working with a poor ontology (second challenge), other explicit features can be desired (specified in the language ROWL) with links to explicit information about some kinds of anomalies that the ARS would find, the noninteresting ones to instantly discard.
- Ontology mapping (fourth challenge) can be certified up to a point by the ARS associated with the ontology target: the trust in the mapping is augmented if the ARS certifies the translation of essencials relationships and properties on concepts of ontology source. On the other hand, in the concrete example of ACL2, it is extremately interesting to exploit the *certified arithmetic reasoning* embedded in the self system ACL2. In this way we could add arithmetic reasoning about data to OWL (by means logical interpretation in ACL2 logic).
- A witness of the existence of nondesired unintended models (fifth challenge) may be the unprobability with the recommended ARS of basic theorems on concepts of the ontology.

[8] http://www.cs.utexas.edu/users/moore/acl2/

[9] http://www.risc.uni-linz.ac.at/projects/basic/mathbroker/

[10] http://monet.nag.co.uk/cocoon/monet/index.html

- If Ontology designer thinks that it is highly probable that inconsistencies appear, she/he can associate, in the earlier stages of ontology evolution, a reasoning model based in argumentation. After that, a standard ARS could be associated, changing only the corresponding links in the ROWL ontology. In this way, the ontology will benefit from the use of state-of-art certified reasoning models.

It is important to notice the importance of that the framework will be certified. This way, the ATP embedded in the cleaning agent will be correct and thus it will not have an anomalous behaviour.

Conclusions

The challenges that we have shown, about the application of ARS for KDB cleaning, represent a partial view. Nevertheless. they affect to every phase in usual data cleaning [39]: *data analysis*, *definition of mapping rules, verification, transformation and backflow of cleaning data*. From these fundational issues, many problems arrive to real-life cleaning scenarios.

One might think that some of the aspects of the challenges does not have importance: we cannot wait SW replaces to the current WWW in such way that, really, KDB cleaning becomes a computational logic enterprise. Therefore, the marriage data/ontology may not be celebrated in many cases, and some kind of classical data cleaning, combined with automated reasoning engineering, will remain as the sole option. To sum up, cleaning agents for SW raises a serie of challenges that overcomes logic-based agents and classical datacleaning techniques if they are not combined.

Acknowledgements

This work is partially supported by the TIC-137 of Plan Andaluz de Investigación.

References

[1] J. Alonso-Jiménez, J. Borrego-Díaz, A. M. Chávez González , M. A. Gutiérrez-Naranjo and J. David Navrro-Marín, "Towards a Practical Argumentative Reasoning with Qualitative Spatial Databases". *Proc. of 16th Int. Conf. on Industrial and Engineering Applications of Artificial Intelligence and Expert Systems IEA/AIE 2003*, LNAI 2718, Springer-Verlag, 2003, pp. 789-798.

[2] J.A. Alonso-Jiménez, J. Borrego-Díaz, A. M. Chávez-González, M. A. Gutiérrez-Naranjo, J. D. Navarro-Marín, " A Methodology for the Computer–Aided Cleaning of Complex Knowledge Databases," *Proc. of IEEE Int. Conf. on Industrial Electronics, Control and Instrumentation IECON 2002,* IEEE Press, 2003, pp. 1806-1812.

[3] G. Antoniou, A. Kehagias. "On the Refinement of Ontologies". International Journal of Intelligent Systems, vol. 15 2000, pp. 623-632.

[4] T. J. M. Bench-Capon, "The role of ontologies in the verification and validation of knowledge-based systems" Int. J. Intell. Syst. 16(3): 377-390 (2001).

[5] L. E. Bertossi, C. Schwind, "Database Repairs and Analytic Tableaux", Annals of Mathematics and Artificial Intelligence 40(1-2): 5-35 (2004)

[6] I. Bloch et al. "Fusion: General concepts and characteristics", International Journal of Intelligent Systems vol. 16, 2001, pp. 1107-1134.

[7] A. Boskovitz, R. Goré, M. Hegland, "A Logical Formalisation of the Fellegi-Holt Method of Data Cleaning", Proceedings of 5th International Symposium on Intelligent Data Analysis, IDA 2003, LNCS vol. 2810, Springer-Verlag, 2003, pp. 554-565

[8] P. Bouquet, F. Giunchiglia, F. van Harmelen, L. Serafini, and H. Stuckenschmidt, "C-OWL: Contextualizing ontologies," *Proc. of the 2nd Int. Semantic Web Conf ISWC2003*, LNCS 2870, Springer-Verlag, 2003, pp. 164-179.

[9] J. Cantwell, "Resolving Conflicting Information", Journal of Logic, Language and Information vol. 7, 1998, pp. 191-220.

[10] A. Celle, L. Bertossi, "Consistent Data Retrieval", Information Systems vol. 19, 1994, pp. 1193-1221.

[11] L. Cholvy, S. Moral, "Merging databases: Problems and examples", International Journal of Intelligent Systems vol. 16, 2001, pp. 1193-1221.

[12] J. Denzinger, I. Dahn, "Cooperating Theorem Provers", in W. Bibel, P.H. Schmitt (eds.): Automated Deduction. A basis for applications. Vol. II: Systems and Implementation Techniques, Kluwer Academic Publishers, 1998, pp. 383-416.

[13] P. M. Dung, "On the Acceptability of Arguments and its Fundamental Role in Nonmonotonic Reasoning, Logic Programming and n-Person Games", Artificial Intelligence vol. 77, 1995, pp. 321-358.

[14] M. Elvang-Goransson, A. Hunter, "Argumentative Logics: Reasoning with Classically Inconsistent Information", Data and Knowledge Engineering vol. 16, 1995, pp. 125-145.

[15] J. Farrugia, "Model-theoretic semantics for the Web", Proc. of World Wide Web 2003 Congress, pp. 29-38.

[16] R. Fikes, D. L. McGuinness, R. Waldinger, "A First-Order Logic Semantics for Semantic Web Markup Languages", Report n. KSL-02-01 of Knowledge Systems Laboratory, Stanford University, 2002.

[17] H. Galhardas, D. Florescu, D. Shasha, E. Simon, C. Saita, "Declarative Data Cleaning: Language, Model, and Algorithms" Proc. of 27fh Very Large Databases Conference VLDB 2001, Morgan-Kauffman, pp. 371-380.

[18] A. Gómez-Pérez, "Evaluation of Ontologies" International Journal of Intelligent Systems vol. 16, 2001, pp. 391-409.

[19] J. Grant, V.S. Subrahmanian, "Applications of Paraconsistency in Data and Knowledge Bases", Synthese, Vol. 125, 2000, pp. 121-132.

[20] S. Greco, E. Zumpano, "Querying Inconsistent Databases", 7th International Conference on Logic for Programming and Automated Reasoning LPAR 2000, LNCS vol. 1955, Springer-Verlag, 2000, pp. 308-325.

[21] N. Guarino, C. A. Welty, "Evaluating ontological decisions with OntoClean", Communications of ACM vol. 45, 2002, pp. 61-65.

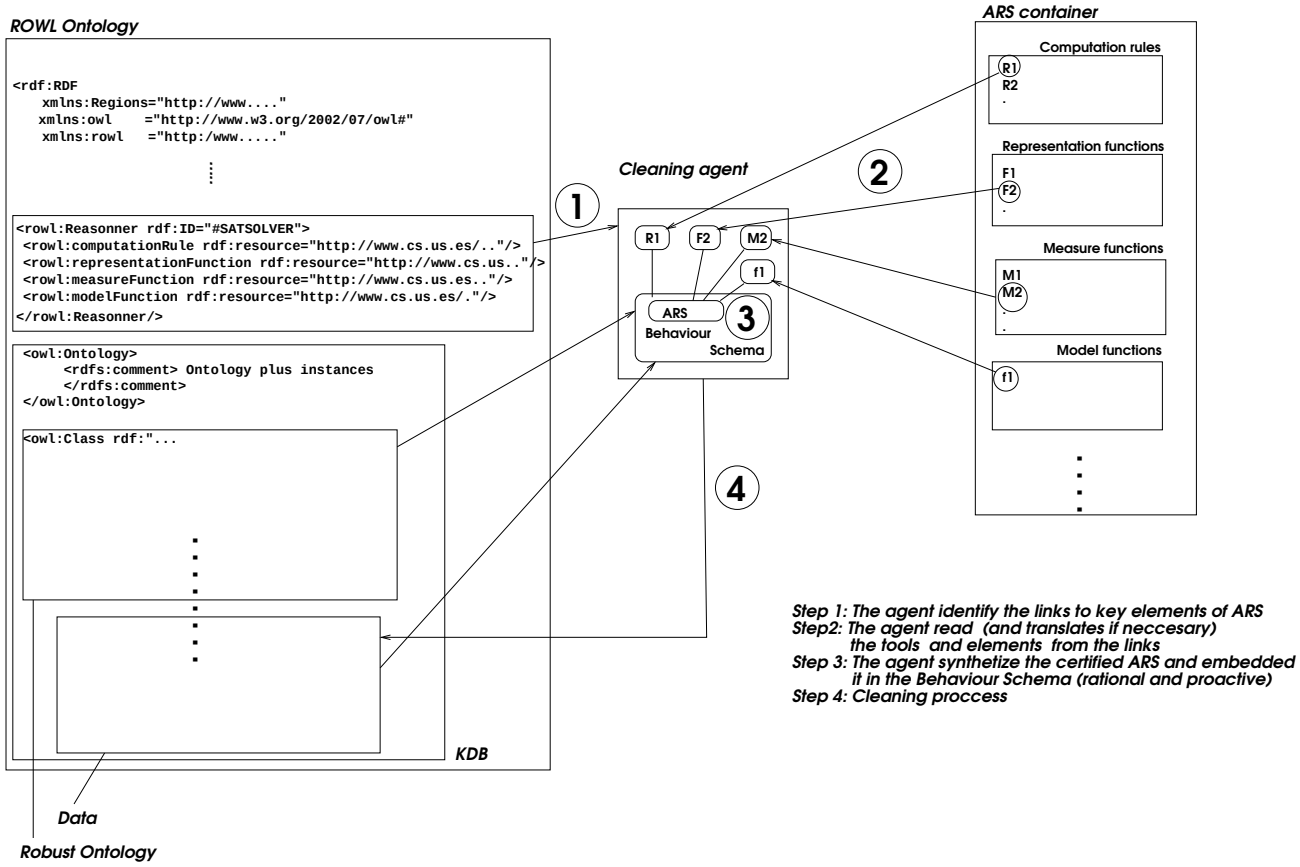

Figure 5: The dream of a general purpose logic-based cleaning agent

[22] J. Heflin, "Towards the Semantic Web: Knowledge Representation in a Dynamic, Distributed Environment," Ph.D. Thesis, University of Maryland, College Park. 2001

[23] I. Horrocks, J. Angele, S. Decker, M. Kifer, B. Grosof, G. Wagner, "Where are the rules?," *IEEE Intelligent Systems* vol. 18, no. 5, Sept./Oct. 2003, pp. 76–83

[24] I. Horrocks and P.F. Patel-Schneider, "Reducing OWL Entailment to Description logics satisfiability," *Proc. of 2nd Int. Semantic Web Conf. (ISWC2003)* LNCS 2870, Springer-Verlag, 2003, pp. 17–29.

[25] A. Hunter, "Paraconsistent Logics", in D: Gabbay and Ph. Smets (eds.) Handbook of Defeasible Reasoning and Uncertain Information, Kluwer, 1998, 13-44.

[26] A. Hunter, "Evaluating the significance of inconsistencies", in Proceedings of the International Joint Conference on AI (IJCAI'03), Morgan Kaufmann, 2003, pp. 468-473, .

[27] M. C. A. Klein, D. Fensel, "Ontology versioning on the Semantic Web", Proceedings of the 1st Semantic Web Symposium SWWS 2001, pp. 75-91.

[28] K. M. Knight, "Two Information Measures for Inconsistent Sets", Journal of Logic, Language and Information vol. 12, 2003, pp. 227-248.

[29] S. Link, "Consistency Enforcement in Databases", Proc. of 2nd Int. Workshop on Semantics in Databases 2001, LNCS vo. 2582, Springer-Verlag, 2003, pp. 139-159

[30] A. Maedche, B. Motik, L. Stojanovic, R. Studer and R. Volz, "Ontologies for Enterprise Knowledge Management," *IEEE Intelligent Systems* vol. 18, no. 2, March/April 2003, pp. 26–33.

[31] A. Maedche, B. Motik, L. Stojanovic, "Managing multiple and distributed ontologies on the Semantic Web", The Very Large Database Journal vol. 12, 2003, pp. 286-302.

[32] R. MacGregor, I.-Y. Ko, "Representing Contextualized Data using Semantic Web Tools", Proceedings of the First International Workshop on Practical and Scalable Semantic Systems Services.

[33] P. Marquis, N. Porquet, "Resource-Bounded Paraconsistent Inference", Annals of Mathematics and Artificial Intelligence vol. 39, 2003, pp. 349-384.

[34] F.J. Martín-Mateos, J.A. Alonso-Jiménez, M.J. Hidalgo-Doblado, J.L. Ruiz-Reina, "Verification in ACL2 of a Generic Framework to Synthesize SAT-Provers," *Proc. of 12th Int. Workshop on Logic Based Program Development and Transformation LOPSTR '02*, LNCS 2664, Springer-Verlag, 2003, pp. 182-198.

[35] F.J. Martín-Mateos, J.A. Alonso-Jiménez, M.J. Hidalgo-Doblado, J.L. Ruiz-Reina, "Formal verification of a generic framework to synthesize SAT-provers", Journal of Automated Reasoning, to appear (2004).

[36] E. Mayol, E. Teniente, "Consistency preserving updates in deductive databases", Data and Knowledge Engineering vol. 47, 2003, pp. 61-103.

[37] S. Pradhan, "Connecting Databases with Argumentation", 14th Int. Conf. on Applications of Prolog INAP 2001, LNCS vol, 2543, Springer-Verlag, 2001, pp. 170-185.

[38] M. Peim, E. Franconi, N. W. Paton, "Estimating the quality of answers when querying over description

logic ontologies", Data and Knowledge Engineering vol. 47, 2003, pp. 105-129.

[39] E. Rahm, H-H Do, "Data Cleaning: Problems and Current Approaches", IEEE Data Engineering Bulleting vol. 23, 2000, pp. 3-13.

[40] M. Richardson, R. Agrawal, P. Domingos, "Trust Management for the Semantic Web", Proc. of 2nd Int. Semantic Web Conference ISWC 2003, LNCS vol. 2870, Springer-Verlag, 2003, pp. 351–368.

[41] G. Yang, M. Kifer, "Reasoning about Anonymous Resources and Meta Statements on the Semantic Web", Journal of Data Semantics vol. 1 (LNCS vol. 2800), 2003, pp. 69-97.

Appendix

We have remarked that the CGF only synthesize, at the moment, SAT provers. Let us see how would to run a cleaning session of a ROWL ontology. Consider us the little inconsistent ROWL ontology:

```
<rowl:Reasoner rdf:ID=''#Davis-Puntnam-SATSOLVER''>
 <rowl:ComputationRule
     rdf:resource=''#DP-comp-rule''/>
 <rowl:RepresentationFunction
     rdf:resource=''#DP-repr''/>
 <rowl:MeasureFunction
     rdf:resource=''#DP-measure''/>
 <rowl:ModelFunction
     rdf:resource=''#DP-models''/>
</rowl:Reasoner>

<rowl:Asummption
  rdf:resource=''#Uniques-names-principle''/>
<rowl:Asummption
  rdf:resource=''#Domain-closure-principle''/>

<owl:Class rdf:ID="Herbivore">
<rdfs:Comment> herbivores are exactly those
        animals that eat some plant</rdfs:Comment>
  <owl:intersectionOf rdf:parsetype="Collection">
    <owl:Class rdf:about="animal">
    <owl:Restriction>
      <owl:onProperty rdf:resource=#eats"/>
      <owl:someValuesFrom rdf:resource=#plant"/>
    </owl:Restriction>
  </owlintersectionOf>
</owl:Class>

<owl:Class rdf:ID="Carnivore">
  <rdfs:Comment> carnivores are animals  which
           are not herbivores and
           they eat also animals</rdfs:Comment>
  <owl:disjointWith rdf:resource="#Herbivore">
  <owl:intersectionOf rdf:parsetype="Collection">
    <owl:Class rdf:about="animal">
    <owl:Restriction>
      <owl:onProperty rdf:resource=#eats"/>
      <owl:someValuesFrom rdf:resource="#animals">
    </owl:Restriction>
  </owl:intersectionOf>
</owl:Class>

<Carnivore rdf:ID="Tarzan">
   <eats rdf:resource="Pumbaa">
   <eats rdf:resource="potato">
</Carnivore>

<Animal rdf:ID="Pumbaa">
   <eats rdf:resource="potato">
</Animal>

<Plant rdf:ID="potato">
```

The first block shows the four ingredients to synthesize the SAT solver (in this it describes the Davis-Putnam procedure). The second block claims two classic assumptions on databases (*unique names principle* and *domain closure axiom*) that allows to transform the ontology into a propositional set of formulas. For the above example, the set is:

```
(<-> CARNIVORE-POTATO
  (& (& (- HERBIVORE-POTATO) ANIMAL-POTATO)
     (/ (& EATS-POTATO-POTATO ANIMAL-POTATO)
        (/ (& EATS-POTATO-PUMBAA ANIMAL-PUMBAA)
           (& EATS-POTATO-TARZAN ANIMAL-TARZAN)))))

(<-> CARNIVORE-PUMBAA
    (& (& (- HERBIVORE-PUMBAA) ANIMAL-PUMBAA)
       (/ (& EATS-PUMBAA-POTATO ANIMAL-POTATO)
          (/ (& EATS-PUMBAA-PUMBAA ANIMAL-PUMBAA)
             (& EATS-PUMBAA-TARZAN ANIMAL-TARZAN)))))

(<-> CARNIVORE-TARZAN
     (& (& (- HERBIVORE-TARZAN) ANIMAL-TARZAN)
        (/ (& EATS-TARZAN-POTATO ANIMAL-POTATO)
           (/ (& EATS-TARZAN-PUMBAA ANIMAL-PUMBAA)
              (& EATS-TARZAN-TARZAN ANIMAL-TARZAN)))))

(<-> HERBIVORE-POTATO
     (& ANIMAL-POTATO
        (/ (& EATS-POTATO-POTATO PLANT-POTATO)
           (/ (& EATS-POTATO-PUMBAA PLANT-PUMBAA)
              (& EATS-POTATO-TARZAN PLANT-TARZAN)))))

(<-> HERBIVORE-PUMBAA
     (& ANIMAL-PUMBAA
        (/ (& EATS-PUMBAA-POTATO PLANT-POTATO)
           (/ (& EATS-PUMBAA-PUMBAA PLANT-PUMBAA)
              (& EATS-PUMBAA-TARZAN PLANT-TARZAN)))))

(<-> HERBIVORE-TARZAN
     (& ANIMAL-TARZAN
        (/ (& EATS-TARZAN-POTATO PLANT-POTATO)
           (/ (& EATS-TARZAN-PUMBAA PLANT-PUMBAA)
              (& EATS-TARZAN-TARZAN PLANT-TARZAN)))))

PLANT-POTATO
EATS-PUMBAA-POTATO
EATS-TARZAN-PUMBAA
EATS-TARZAN-POTATO
HERBIVORE-PUMBAA
CARNIVORE-TARZAN
```

With respect to the algorithm synthesis and verification of the SAT solver by the CGF, the input of the ACL2 program essentially takes the elements defined in the first block;

```
(definstance-*generic-sat*
 ((gen-object-p      DP-object-p)
  (gen-object-list-p DP-object-list-p)
  (gen-repr          DP-repr)
  (gen-dist-val      DP-dist-val)
  (gen-dist-val-lst  DP-dist-val-lst)
  (gen-comp-rule     DP-comp-rule)
  (gen-select        DP-select)
  (gen-measure       DP-measure)
  (gen-model         DP-models))
 "-davisputnam")
```

The running time of the CGF for the synthesis and verification of the SAT solver is 0.28 seconds;

```
Summary
Form: (ENCAPSULATE NIL
            (DEFUN GEN-MEASURE-LST-DAVISPUTNAM ..)..)
Rules: NIL
Warnings: None
Time:  0.28 seconds (prove: 0.11,
                     print: 0.00,
                     other: 0.17)
T
```

Finally, the cleaning agent only must to check the satisfiability of this knowledge base;

```
 (generic-sat-davisputnam
       (build-rowl-example1
          '(potato pumbaa tarzan)))

; real time : 0.020 secs
; run time  : 0.020 secs
; NIL
```


José Antonio Alonso-Jiménez is an associate professor at Computer Science and Artificial Intelligence Department at University of Seville. Head of Computational Logic Group of the department (http://www.cs.us.es/glc/), his research interest include computational logic, formal methods and verification on intelligent systems. Contact him at E.T.S. Ingeniería Informática, Universidad de Sevilla, Avda. Reina Mercedes s/n, 41012, Sevilla, Spain; jalonso@us.es.

Joaquín Borrego-Díaz is an associate professor at Computer Science and Artificial Intelligence Department at University of Seville. His research interest includes computational logic, automated reasoning and their applications to knowledge representation and reasoning, with special interest in the intersection of automated reasoning systems and Semantic Web. Contact him at Departamento de Ciencias de la Computación e Inteligencia Artificial, ETS Ingeniería Informática, Avda. de Reina Mercedes s.n., 41012 Sevilla, Spain; jborrego@us.es.

Antonia M. Chávez-González is an assitant professor in the Computer Science and Artificial Intelligence Department at Universidad de Sevilla, Spain. Her research interest includes automated theorem proving and intelligent data cleaning. Contact her at Departamento de Ciencias de la Computación e Inteligencia Artificial, ETS Ingeniería Informática, Avda. de Reina Mercedes s.n., 41012 Sevilla, Spain; tchavez@us.es.

Francisco J. Martín-Mateos is an assistant professor in the Computer Science and Artificial Intelligence Department at University of Sevilla, Spain. His research interest includes formal methods applied to verification and automated deduction. He received a BA in Mathematics from the Universidad Complutense de Madrid and a PhD in mathematics from the Universidad de Sevilla. Contact him at E.T.S. Ingeniería Informática, Universidad de Sevilla, Avda. Reina Mercedes s/n, 41012, Sevilla, Spain; fjesus@us.es.