The Role of Ontologies in the Verification and Validation of Knowledge-Based Systems

Trevor J. M. Bench-Capon
Department of Computer Science, The University of Liverpool, Liverpool, United Kingdom

In this paper, the author gives some preliminary examination of the ways in which an ontology—an explicit specification of the conceptualization of the domain—can support the verification and validation of a knowledge-based system. The discussion is focused on a simple, well-known, example relating to the identification of animals. Key elements of the support provided by the ontology relate to attempting to give coherence to the domain conceptualization; making the role of experts in verification and validation more structured and less at the mercy of interpretation; constraining the number of test cases required to give good coverage of the possible cases; and structuring the testing to give better assurance of its efficacy, and a possible basis for greater automation of the testing process. Finally, the author makes some concluding remarks. © 2001 John Wiley & Sons, Inc.

I. INTRODUCTION

When expert and knowledge-based systems first appeared on the scene, they appeared to offer two main attractions. One was related to their functionality: they gave the promise of allowing applications not susceptible to conventional techniques to be provided with computer support. The other attraction was the benefits that were supposed to arise from the separation of control and problem solving methods from a declarative representation of domain knowledge. This would, it was argued make systems easier to build, easier to modify in the light of changes to the domain, and easier to assess for accuracy and correctness. In fact, these latter goals proved rather more difficult to realize than was anticipated. The aspiration of a knowledge base to be a declarative expression of domain knowledge proved to be unattainable, as distortions deriving from task, problem solving method, representation method, control issues, and the ways in which the knowledge was acquired all compromised the declarativeness of the knowledge in the knowledge base. See, for example, Visser Chap 2,[1] for a discussion of these problems. As a result, the hopes originally pinned to knowledge bases have now become centered on what are

INTERNATIONAL JOURNAL OF INTELLIGENT SYSTEMS, VOL. 16, 377–390 (2001)

known as "ontologies." Ontologies, best characterized as "the explicit specifications of the conceptualization of a domain,"[2] have, in recent years, made a significant impact on thinking about the design and development of knowledge-based systems (KBS), in an effort to restore many of the features originally ascribed to a declarative knowledge base. Typically a number of advantages are said to result from the use of ontologies including:

- Facilitating sharing of knowledge between systems;
- Facilitating reuse of knowledge in new systems;
- Aiding knowledge acquisition; and,
- Improving the verification and validation of knowledge-based systems.

Much has been written about the first three of these topics, but as yet there has been little detailed discussion of the fourth topic. In this paper, the author attempts to outline some of the things that ontologies can do for verification and validation.

The author does this by considering an example. The example the author uses is ZOOKEEPER, a very simple rule base described in Winston's AI textbook (Winston 1992).[3] There are several reasons for choosing this example: it is small enough that the complete system can be given in a short paper; it deals with a domain with which everyone has some basic familiarity; the example itself is very well known; and, because it appears in a widely used textbook, it is for many people their first encounter with a rule-based system. Section II recapitulates this rule base. In Section III, the author discusses how the rule base as it stands might be verified and validated. In Section IV, the author reconstructs an ontology for the domain from the rule base, and in Section V, the author shows some additional possibilities for verification and validation that this allows. In Section VI, the author gives some concluding remarks.

II. A TOY ANIMAL CLASSIFICATION SYSTEM

The rule base for ZOOKEEPER is given in Winston, pp 121–124.[3] It is explicitly limited to the identification of seven animals: a cheetah, tiger, zebra, giraffe, ostrich, penguin, and an albatross. It has 15 rules, enabling identification of these seven animals, often in several ways, to allow for some observations being unobtainable.

The rules (expressed here in Prolog form) are:

```
Z1: mammal(X):-hair(X).
Z2: mammal(X):-givesMilk(X).
Z3: bird(X):-feathers(X).
Z4: bird(X):-flies(X),
          laysEggs(X).
Z5: carnivore(X):-mammal(X),
                eats(X,meat).
```

```
Z6: carnivore(X):-mammal(X),
                  teeth(X,pointed),
                  has(X,claws),
               eyes(X,forwardPointing).
Z7: ungulate(X):-mammal(X),
              has(X,hoofs).
Z8: ungulate(X):-mammal(X),
                 chewsCud(X).
Z9: cheetah(X):-carnivore(X),
             color(X,tawny),
             spots(X,dark).
Z10: tiger(X):-carnivore(X),
            color(X,tawny),
            stripes(X,black).
Z11: giraffe(X):-ungulate(X),
                  legs(X,long),
                neck(X,long),
                color(X,tawny),
              spots(X,dark).
Z12: zebra(X):-ungulate(X),
            color(X,white),
            stripes(X,black).
Z13: ostrich(X):-bird(X),
              not flies(X),
                legs(X,long),
                  neck(X,long),
          color(X,blackandwhite).
Z14: penguin(X):-bird(X),
              swims(X),
              not flies(X),
          color(X,blackandwhite).
Z15: albatross(X):-bird(X),
                flies(X,well).
```

These rules can be used either to identify an animal given a set of observations, or to test a hypothesis that an animal is of a particular species. Now, let us consider how we might go about verifying and validating ZOOKEEPER.

III. VERIFYING AND VALIDATING ZOOKEEPER

Here, the author uses Boehm's well-known distinction between verification and validation[4]:

- Verification "Are we building the product right?"
- Validation: "Are we building the right product?"

As usually interpreted in the context of knowledge-based systems, verification would include checking that the rule base is structurally sound (free of subsumed rules, contradictions, dead-end rules, and the like), while validation would be effected by supplying sets of typical observations and by checking that the identifications produced by the system were possible and correct. Additionally, we might present the rules to an expert and we might ask for confirmation of their correctness.

It is probable that if we were to run such tests, ZOOKEEPER would pass them quite well. There appear to be no structural problems, each rule looks plausible enough to accept on inspection, and appropriate sets of facts will lead to the correct answers.

The only problems would arise through an inability to respond to certain sets of observations: there are certain sets of observations which would not give an answer (such as a white carnivore); and there is a need to give more information than is strictly necessary (such as the orientation of eyes as well the pointedness of the teeth). Both of these possible defects may, however, be acceptable: since we are limited to seven animals, no white carnivores will be observed, and Winston argues for the inclusion of extra information on the grounds that "there is no need for information in rules to be minimal." Moreover, antecedents that are superfluous now may become essential later as new rules are added to deal with other animals" (Winston, p 123).[3]

So shall we conclude that the rule base is entirely satisfactory? The author does not think we should, particularly if we are going to take seriously the possibility of extending the system to cater for, possibly a good many, more animals.

What the author is suggesting is that "building the system right" needs to encompass more than a simple absence of structural defects. What we want, in addition, for the representation to have some kind of *conceptual coherence*, for it to be expressed within some well-defined conceptualization of the domain. This is lacking in ZOOKEEPER. In reaching the final rule-base distinctions were proliferated as and when they were needed to discriminate among the seven particular animals, and without much regard for distinctions that had already been made. If a system is to be built correctly, it should make principled distinctions, and make them in a justifiable manner. For example, spots are "dark" and stripes are "black." Do we want a distinction between dark and black? What other varieties of spots and stripes might there be? Is there really a good difference between being white in color with black stripes, black in color with white stripes, and white and black in color? Without a clear conceptualization to serve as a reference point, it is futile to ask an expert to say whether rules are correct or not. For example, it might be that certain markings resemble rosettes. While one might be prepared to call them "spots" in the absence of an option to call them "rosettes," assent to a rule using spots depends on an assumption as to whether the finer grain distinction is available or not.

We also need the observations to be relatively easy to obtain, if the system is to be able to come with answers consistently. Some of those required by this rule base need judgement to be applied. A particular example of this is the

requirement be that an albatross flies "well." This might well raise differences of opinion and interpretation. Others are rather hard to obtain: "lays eggs" is an occasional thing which might be hard to observe (and not observable at all in the case of a male of the species). At the very least, we need to be aware of what information is likely to be available so that we resort to the information which is harder to obtain only when it is essential. In the rule base above, it is essential that a giraffe or a zebra be first classed as an ungulate. Both of the observations required to classify an animal as an ungulate are, however, not always available. If the designer is unaware of this, practical problems may arise in that giraffes and zebras may not be identified, even though sufficient information is available, whereas if the designer is aware of this problem, rules identifying zebras and giraffes in terms of more readily available observations can be supplied.

Much of the problem derives from the failure initially to conceptualize the domain in a coherent fashion. The strategy is first to classify an animal as a mammal or a bird, then to subdivide mammals into carnivores and ungulates, and then to discriminate members of these categories in terms of some observable features which are indicative of the particular animals in the collection. The higher level distinctions are theory driven, and the rules are determined by theory: for example Z4 is justified on the grounds that "some mammals fly and some reptiles lay eggs, but no mammal or reptile does both" (Winston, p 122).[3] However, in the context of use of the system, Z4 is applicable only to the albatross, since the other two birds are flightless, and if it can fly it is an albatross, so its oviparity is neither here nor there. On the other hand, if we were to take the notion of extensibility seriously Z9 would be inadequate since it describes leopards and jaguars as well as cheetahs. As it stands here, the rules are defective, with respect to the standards of a well-constructed system, because they derive from conflicting conceptualizations of the domain, and conflicting ideas of how the system will be used. Separation of the animals into mammals and birds, and mammals into carnivores and ungulates, is obviously useful for a zoological taxonomy, but is of little practical importance in performing the identifications the system is supposed to supply.

The problems above derive in part from the lack of a clear specification of what the system will be used for as a starting point. Viewed simply from the standpoint of its real use, as an example rule base to illustrate forward and backward chaining, it is adequate. It is only when we project it into standing as a real application that we would need to specify whether it was supposed to identify only seven or an indefinite range of animals; whether it is meant to incorporate known theory about animal classification, or to restrict itself to what can be seen; what kind of judgements the user of the system can be expected to make, and the like. As they stand, the rules represent more the unstructured outpouring of various facts about the animals, rather than a well thought out plan for identification.

In the next section, the author provides a crude ontology for a system which intended to identify animals on the basis of observations made by a nonexpert, and which is intended to cover the seven animals given, but also to be extensible to other animals.

IV. AN ONTOLOGY FOR ZOOKEEPER

A minimal requirement of the ontology is that it should permit the expression of all the facts and rules about animals found in the original rules. The author therefore uses as the foundation for the ontology the observations that can be identified from the ZOOKEEPER rules. We can identify the following predicates which the user is expected to be able to answer questions about. These are additional to the predicates relating to whether the animal is a mammal, is a bird, is a carnivore, and is an ungulate which are internal to the system; the user neither is asked questions about these predicates, nor sees any information about them.

(a) has hair,
(b) gives milk,
(c) has feathers,
(d) flies,
(e) lays eggs,
(f) eats meat,
(g) long legs,
(h) long neck,
(i) tawny color,
(j) dark spots,
(k) white color,
(l) black stripes,
(m) black and white color,
(n) swims,
(o) flies well,
(p) pointed teeth,
(q) claws,
(r) eyes point forward,
(s) hoofs,
(t) chews cud.

Some of these seem to present alternatives, so we can group them accordingly.

(1) skin covering {hair, feathers} [a, c]
(2) color {white, tawny, black and white] [i, k, m]
(3) markings {spots, stripes} [j, l]
(4) movesBy {swims, flies} [n, o]
(5) feet {hoofs, claws} [q, s]

In other cases, there seem to be implicit alternatives, only one option is used in the rule base because the alternatives are not needed by the rules used to identify the target group of animals:

(1) teeth {pointed, ?},
(2) eats {meat, ?},
(3) legs {long, ?},
(4) neck {long, ?},
(5) stripes {black, ?},
(6) spots {dark, ?},
(7) flies {well, ?},
(8) eyes {point forward, ?}.

In other cases we have only a true or false decision:

(1) gives milk,
(2) lays eggs,
(3) chews cud.

We now need to perform some rationalization on this; for example, flying and swimming are not exclusive (consider ducks and swans), so these predicates must be separated. Moreover, flying appears to be a qualitative thing rather than a simple Boolean: we could ask whether the same should apply to swimming as well, and indeed whether we want to include some kind of land motion such as running. We can make the markings and color situation more coherent by saying that an animal has a basic color, and markings, which may be lighter or darker than the basic color. Where we have gaps, because the options do not occur explicitly, these need to be filled.

We could now arrive at the situation where we can identify a set of attributes, and the possible values they can take, shown in Table I. This will provide us with a well defined vocabulary with which to construct a set of rules.

To complete the ontology we need to add some axioms, stating combinations which are impossible. For example:

(A1) Not (eats meat and chews cud).
(A2) Not (Material feathers and chews cud).
(A3) Not (Pattern none and shade not n/a).

In fact, we could probably supply many more such axioms, but at this stage we need not attempt to be exhaustive.

Table I. Attributes and values for ZOOKEEPER.

Coat:
Material {hair, feathers}
Color {white, tawny, black}
Markings:
Pattern{spots, stripes, irregular, none}
Shade {light, dark, n/a}
Facial features
Eyes {forward, sideways}
Teeth{pointed, rounded}
Feet {claws, hoofs}
Flies{no, poorly, well}
Eats{meat, plants, everything}
Size:
Neck{long, normal}
Legs{long, normal}
Gives milk {true, false}
Lays eggs {true, false}
Chews cud {true, false}
Swims {true, false}

We could now construct a table of our example animals, and we could fill in the values for the predicates where they have been given. Where there are gaps, we need to find the information required to complete them. This process is likely to identify additional possibilities for some of the attributes (for example, birds do not have teeth, and penguins eat fish), which will force us to extend the ontology accordingly. Table II is such a table. (Note that some of the answers are conjectural—the author is not an expert, and is unsure what ostriches in fact eat, for example.)

V. USING THE ONTOLOGY IN VALIDATION AND VERIFICATION

We now have an ontology which we can use to verify and validate a knowledge base built on it. First, however, we need to check the quality of the ontology itself. This is where the expert comes in: the expert should not be shown the encoded rules, but rather the ontology. This changes the role of the expert significantly. The expert no longer examines rules, but instead the vocabulary, and the table of attributes. With respect to the vocabulary the expert should check:

- that the attributes represent sensible distinctions,
- that the values are exclusive,
- that the values are exhaustive.

The point about values can be addressed from two standpoints: either from the point of view of the existing collection, or from the point of view of a potentially extended collection. The first indicates what is needed to test the rule base against its current operation, and the other provides an indication of its extensibility.

Table II. Attributes of animals in ZOOKEEPER.

Predicate	Cheetah	Tiger	Zebra	Giraffe	Ostrich	Penguin	Albatross
Material	Hair	Hair	Hair	Hair	Feathers	Feathers	Feathers
Color	Tawny	Tawny	White	Tawny	Black	Black	White
Pattern	Spots	Stripes	Stripes	Spots	Irregular	Irregular	None
Shade	Dark	Dark	Dark	Dark	Light	Light	N/a
Eyes	Forward	Forward	Sideways	Sideways	Sideways	Forward	Sideways
Teeth	Pointed	Pointed	Rounded	Rounded	None	None	None
Feet	Claws	Claws	Hoofs	Hoofs	Toes	Toes	Toes
Neck	Normal	Normal	Normal	Long	Long	Normal	Normal
Legs	Normal	Normal	Normal	Long	Long	Short	Normal
Gives milk	True	True	True	True	False	False	False
Flies	No	No	No	No	No	No	Well
Eats	Meat	Meat	Plants	Plants	Both	Fish	Fish
Lays eggs	No	No	No	No	True	True	True
Chews cud	No	No	True	True	No	No	No
Swims	No	Yes	Yes	No	No	Yes	No

Table III. Validated attributes and values for ZOOKEEPER.

Coat:
  Material {**hair**, **feathers**, scales}
  Color {**white**, **tawny**, **black**, gray, russet}
  Markings:
    Pattern{**spots**, **stripes**, **irregular**, **none**}
    Shade {light, **dark**, **n / a**}
Facial features
  Eyes {**forward**, **sideways**}
  Teeth{**pointed**, **rounded**, **none**}
Feet {**claws**, **hoofs**, **toes**}
  [*Comment*: *feet are hard to observe* (*Winston*)]
Flies{**no**, poorly, **well**}
Eats{**meat**, **plants**, both}
  [*Common*: *meat includes fish*]
Size:
  Neck{**long**, **normal**}
  Legs{**long**, **normal**}
Gives milk {**true**, **false**}
Lays eggs {**true**, **false**}
Chews cud {**true**, **false**}
Swims {**true**, **false**}
[*Comment*: *could have* {*no*, *poorly*, *well*}]

Also, to facilitate testing the expert should indicate whether observations are always available, or only sometimes available.

Following this process, we might modify Table I to give Table III. Here, always observable attributes are indicated in bold, as are values required by the current seven animals.

The expert should also examine the table of attributes (Table II), to confirm that these entries are correct. The table can be further verified by ensuring that it does not conflict with any of the axioms. By concentrating on the ontology rather than the rules, the role of the expert becomes much more well defined, and more systematic so that there is less possibility of interpretation allowing errors to go unnoticed.

Once we are satisfied with the ontology, we can now proceed to examine the rule base. The first thing to do is to map the predicates found in the rule base onto the terms in the ontology. We can do this as a set of Prolog rules. A possible set of mappings is as

```
M1: mammal(X):-cost_material(X,hair).
M2: hair(X):-coat_material(X,hair).
M3: givesMilk(X):-gives_milk(X,true).
M4: bird(X):-coat_material(X,feathers).
M5: feathers(X):-coat_material(X,feathers).
M6: flies(X):-not flies(X,no).
M7: laysEggs(X):-lays_eggs(X,true).
```

```
M8: carnivore(X):-eats(X,meat).
M9: has(X,Y):-feet(X,Y).
M11: ungulate(X):-chews_cud(X,true).
M12: spots(X,Y):-markings_pattern(spots),
                 markings_shade(Y).
M13: stripes(X,Y):-markings_pattern(stripes),
                   markings_shade(Y).
M18: swims(X):-swims(X,true).
M19: flies(X,Y):-flies(X,Y).
M20: chews_cud(X):-chews_cud(X,true).
```

No mapping rules are given for `teeth/2`, `eats/2`, `eyes/2`, `color/2`, `legs/2`, `neck/2`, or `flies/2` since those appear both in the rule base and the ontology.

We can now use the ontology to construct suitable test data to test the rule base. One major problem that has always existed in testing rule bases of a substantial size (and in this context even ZOOKEEPER can be considered substantial), is the combinatorial explosion that combining the predicates in test cases gives rise to. In the original ZOOKEEPER, there were 20 predicates each of which appeared capable of being true or false independently, giving more than a million possible combinations. On this basis, exhaustive testing can be considered impossible, and so test data was selected by using some selected plausible combinations. There was, however, no systematic way of generating these, and so coverage of the important cases was not only not ensured, but there was not even any reliable way of estimating the coverage provided by the test data. The existence of the ontology allows us to improve on this. First, the grouping together of attributes in the ontology identifies predicates that are not independent. If we allow for five colors coverage of these as Booleans would require 64 cases. By considering them as they are in the ontology, however, there are only five cases. If we confine ourselves to testing only the attributes which the expert identified as always available as observations, and only the values actually used by our current collection, we have only 1152 combinations. The useful test data moreover, contains only those cases which conform to the constraints imposed by the axioms. This enables a substantial further pruning. If we are able to identify a good set of axioms, then exhaustive testing becomes a possibility. Finally, since we have in Table II identified the cases which should return particular examples, we can see that the ontology provides the essential input for an automated test harness.

Testing against these cases may identify cases where the system produces:

(1) an incorrect answer;
(2) multiple answers;
(3) no answer.

Case (1) requires amendment to the, offending rule, or an additional axiom to exclude the case as impossible. For example, the rule bases as it stands will

identify swimming animals as giraffes, and animals with claws as giraffes, supposing the conditions in Z11 to be satisfied. The second case requires an extra axiom to rule out cases where we have both `chews_cud(X,true)` and `feet(X,claws)`. The first case cannot be dealt with by an axiom since the conditions are not impossible, it is just that giraffes cannot swim, although some other long necked spotted tawny ungulate might be able to. We therefore need to deal with this by adding `swims(X,false)` as an extra condition in Z11.

Case (2) indicates either that some rule must be made more specific (possibly using not always available features), or—potentially—that the current ontology is inadequate and requires another predicate to discriminate the cases. Consider, for example, rules Z13 and Z14. On the basis of the *guaranteed observable* predicates, which, recall, do not include swimming and flying, cases will allow both these rules to fire, identifying the animal as both a penguin and an ostrich. Examination of these cases will reveal that ostrich is the right answer, since the cases contain `legs(X,long)` and `neck(X,long)`. So, we could rectify the situation without recourse to the intermittently observable predicates by adding `legs(X,normal)` as an extra condition to Z14. Discriminating between a cheetah and a leopard would, however, require an extension to the ontology, since in terms of what is currently in the ontology the two are identical. Perhaps therefore we would need to extend the ontology to allow for a condition relating to tree climbing ability, or retractable claws.

Case (3) can arise in several situations. It might be that:

(a) The case represents an impossible combination, so that there should be an axiom in the ontology excluding such cases;
(b) The case is possible and that animals exist satisfying this set of attributes, but these animals are not in the collection. In which case the rule base is correct, and the combination should not be observed in practice. We can therefore either add a new rule, extending the coverage to animals outside the collection, or simply disregard it;
(c) The case is possible, but identification is reliant on some not always available feature. For example, the identification of the albatross turned on its flying power, which might not be observable. This third case is most problematic. If removing the offending antecedent creates case (1) or case (2) problems, then we have to reconcile ourselves to a certain incompleteness, or find some always available discriminating observation.

Case (2) might lead us to *introduce* antecedents relating to intermittently observable features, whereas case (3) may motivate us to *remove* them.

The testing process could be represented as a diagram shown in Figure 1.

Note that process 1, *generate case*, uses the ontology; the decision *violates axioms*?, uses the output from process 1 together with the axioms from the ontology; process 2, *execute cases* uses the rule base and the mappings from the ontology predicates into the rule-base predicates; *answer correct*? and *animal in collection*? requires an extensional description of actual animals such as is provided by Table II; processes 3 and 5, *add axiom and modify ontology* modify the ontology; processes 4 and 6, *modify rules* and *add rule*, modify the rule base; and *case possible*? *and modifications possible*? require input from the expert.

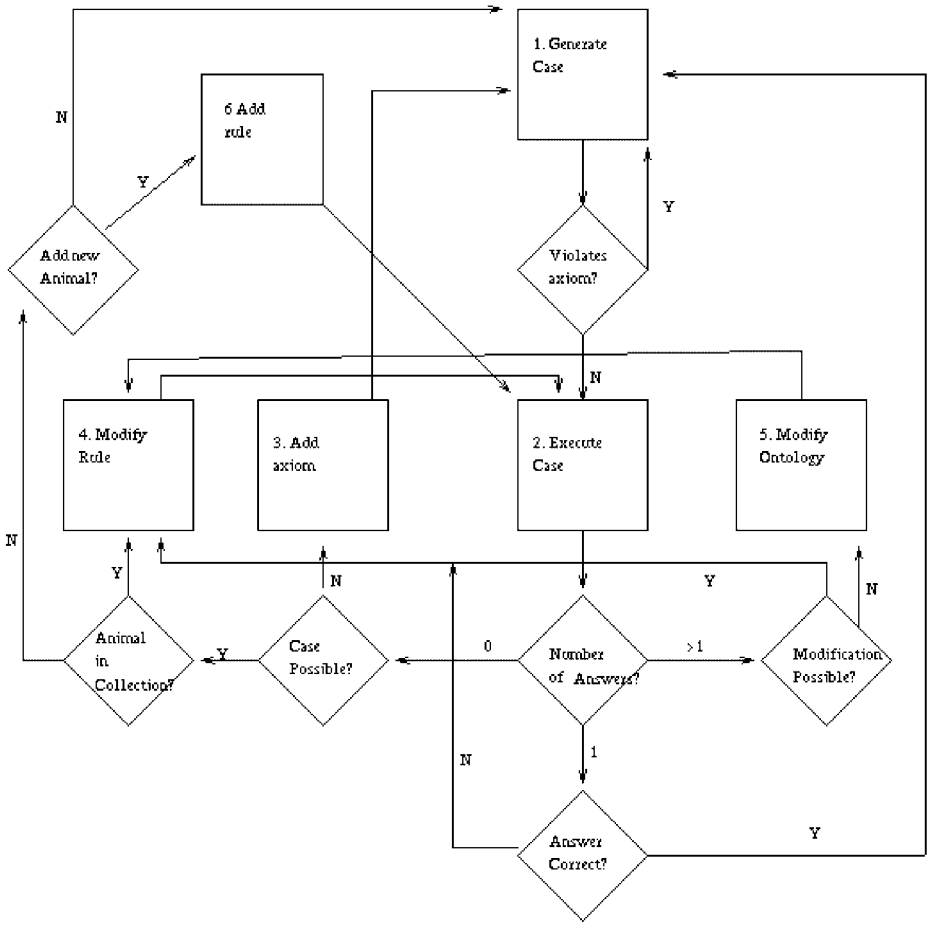

Figure 1. Schematic of testing process.

Adding axioms will prune the cases subsequently generated and additional and modified rules are tested before a new case is generated. A prototype of this test harness has been implemented in Prolog.

The ontology cannot, of course, work magic: the testing effort required even with this test harness is substantial and nontrivial, requiring as it does considerable expert input. It does, however, supply the discipline and structure necessary for testing a system, and in any event testing is *always* for any system an important and length task, typically consuming anything between 25 and 30% of the development time of a software project. Moreover, the time spent in getting the initial ontology right—particularly with respect to a complete specification of the necessary axioms, is handsomely repaid by savings in required testing.

In addition to these possibilities for verification and validation against the ontology, normal structural checks should, of course, be applied. The quasirandom testing for validation is, however, unnecessary given the more structured approach permitted by the ontology.

VI. CONCLUSIONS

In this paper, the author has used a simple example to indicate how the availability of an ontology can aid verification and validation. For verification, we enable the expert to check the vocabulary in a structured manner, and in a way which discriminates between currently needed information and information which will permit some straightforward extensions to the collection of animals. For validation, we have shown how the testing process can be structured using an ontology.

Since the integration of knowledge-based and database systems is becoming increasingly topical, we may also make some notes here on the relation between databases and ontologies. Table II is effectively a database recording the attributes of a test set. Table III corresponds well with a database schema with its identification of attributes and domains for those attributes. Thus, if we are founding a knowledge-based system on an existing database, much will be there —particularly with regard to the attributes we categorized as always available. Some difference between the attributes required by a database and a KBS may be motivated by the expert's conceptualization, or by the need for intermediate predicates to support problem solving, but the database schema should provide an excellent starting point. However, we should not become too enthusiastic about what we can get from the schema, since database work is also exploring the benefits of relating their schemas to ontologies (see, for example, Ref. 5), or similar entities such as the common concept approach.[6]

To summarize the thrust of this paper: ontologies can help drive verification and validation by;

- providing an explicit and coherent conceptualization of the domain;
- allowing the expert to inspect the distinctions made in the ontology rather than being forced to make judgement calls on the rules, making the role of the expert better defined and less subjective;
- providing a means of structuring testing;
- suggesting appropriate responses to flaws indicated by testing.

This paper is a revised and extended version of a paper presented at a workshop on verification, validation, and integrity issues in expert and database systems, a track in the Ninth International Workshop on Database and Expert System Applications, Vienna 1998. That earlier paper appeared in Wagner R, editor. In: Proceedings of the Ninth International Workshop on Database and Expert System Applications, IEEE Press, Los Alamitos, 1998. pp 64–69. The author thanks all those participants in the workshop whose comments on that paper have led to improvements in this version.

References

1. Visser PRS. Knowledge specification for multiple legal tasks. Dordrecht/Norwell, MA: Kluwer Academic; 1995.
2. Gruber TR. Towards principles for the design of ontologies used for knowledge sharing. Int J Human−Comput Interact 1995;43:907−928.
3. Winston PH. Artificial intelligence. 3rd ed. Reading, MA: Addison-Wesley; 1992.
4. Boehm BW. Software engineering economics. Englewood Cliffs, NJ: Prentice Hall; 1981.
5. Masood N, Eaglestone B. Semantics based schema analysis. In: Proceedings of DEXA1998, Lecture Notes in Computer Science. Berlin/New York: Springer-Verlag; 1998. Vol 1460, p 80−89.
6. Yu C, Sun W, Dao S, Keirsey D. Determining relationships among attributes for interoperability of multidatabase systems. In: Proceedings of the First International Workshop on Interoperability of Multidatabase Systems, IMS, Kyoto, 1997. p 251−257.