Consensus based validation in knowledge base systems

1. Introduction

In all the processes (e.g., knowledge acquisition (Torra et al., 1995) and validation (Meseguer et al., 1993)) related with the definition of a Knowledge Based System (that is, the construction of a model) is more and more frequent the participation of several experts. In this case the combination of information is used so that the knowledge embedded in the KBS is more reliable and with less errors than when a single expert is considered. This presentation is focused on synthesis of information for KBS validation.

Validation of Knowledge based systems (KBS), like validation of conventional programming is to check if the system satisfies the needs of the user. However, software engineering techniques have a limited use for KBS builders because both frameworks are not equivalent. Some of the reasons that make the difference are pointed out in (Bellman, 1990, Nazareth et al., 1993, O'Keefe et al., 1993, Meseguer et al., 1993, Krause et al., 1993):

- KBSs have deceptively simple procedures. Most of the program flow and control is hidden in the knowledge base. In fact, it is difficult to divorce between the software and the model.

- Little method or experience is nowadays available for identifying and correcting the errors that may occur.

- Most KBS are not "stand-alone" systems; to the contrary, most KBSs are only one (often small) part of a larger system.

- Usually, the requirements are ill-defined. Because of that, the specification is often incomplete and not adequate for validation. In fact, a complete specification would suggest the use of a traditional programming language.

Several tools have been developed for KBS validation to overcome these difficulties. They can be classified in three groups, as the validation of a KBS encompasses three different analysis (Meseguer, 1992, Hoppe, Meseguer, 1993): verification, evaluation and test. On one hand, verification and evaluation consist of checking whether the system satisfies, respectively, completely formalizable requirements and partially formalizable ones (inconsistencies, circularity rules,... e.g., Kinkiélé, Ayel, 94, O'Keefe, O'Leary, 93). On the other hand, a test is an exam of the behaviour of the system with respect to a given sample case set (e.g., in Hernandez, et al., 1995).

These tools usually need, besides of the knowledge embedded in the KBS, some extra knowledge that should be supplied by domain experts. This fact, together with the one that validation can be performed at different stages, makes validation a process related with KBS definition instead
of a separated one. In fact, these days some tools are built to integrate validation in KBS developing systems. See for example (Gaines et al., 1993) and (O'Keefe et al., 1993).

As it has been said, one of the analysis to be performed on a KBS is the study of its level of expertise (to test the KBS). This study can be done (O'Keefe et al., 1993) considering the individual components of the system or the system as a whole. In this later case, the system is treated as a black box simply to determine if it is making the right decisions. On the other hand, in the former case, the KBS is opened up to determine if the line of reasoning is correct, i.e., if it is making the right decisions for the right reasons.

When a system is tested as a whole, we need a sample case set. The selection of cases in this set has to follow some guidelines. This are, according to (O'Keefe et al., 1993):

- The sample case set has to reflect the cases that the system will encounter.
- The number of cases has to be enough to elicit the range of parameters necessary to test the system and to be able to establish some statistical measures of significance.
- The characteristics of the cases should be established from the study of the nature of the problem.
- In some problems the decisions taken by the expert precipitate the actual outcome. Then, the cases cannot be taken into account.

Sometimes the sample case set is not available. In this cases some synthetic ones can be produced. However, when this is done by hand, at is pointed out by (O'Keefe et al., 1993) it is dangerous, and demands considerable objectivity. In some cases the sample case set is created automatically (see Ayel, Lauren 1991, Ayel, Vignollet, 1993).

When a KBS is tested, together with the sample case set we need the optimum solution for each case. This solution is called the golden standard. However, the golden standard is often not known, instead, we usually have available the diagnosis given by an expert or by a set of experts. In this later case, the golden standard can be approximated by a synthesised diagnosis obtained from the diagnosis of the experts through a consensus function. However, it is not always possible to combine the diagnosis of all experts because the values of the experts might be non comparable (O'Leary, Pincus, 1993) or because there is not an intuitively clear consensus function.

When a set of experts is available, Turing tests are appropriate as a validation method (O'Keefe, O'Leary, 1993). Although there are some drawbacks in their application (see Rousset, 1994, for a review of fundamental problems) they have been used in KBS validation.

2. Previous results

The main intention of our work was a formal definition of an approximation of the golden standard, and its application to test the expert system Pneumon-ia (Verdaguer, 1989). In the first
approximation, due to the fact that in a diagnosis (i.e., a mapping from possible solutions to linguistic labels) a linguistic label in isolation has no clear meaning we dealt with relations between pairs of possible solutions. In fact, the meaning of a label depends on the whole set of linguistic labels used and also on the expert that uses it. Therefore, we consider more meaningful, and more comparable, the relation among pairs of values than the values themselves. According to this, we built a relation from each diagnosis.

In a first step, instead of approximating the golden standard relation by a synthesis of the relations we introduced a consensus interval where the golden standard relation should belong. We introduced both a crisp and a fuzzy consensus interval, stated some properties of both intervals and analysed the results of the expert system Pneumon-ia in relation with them (by means of a Turing Test).

In a second step we studied a synthesising function for fuzzy relations. We built a function when it is supposed to be a quasi-arithmetic mean. The function was built, after stating some natural conditions, through functional equations (Aczél, 1987). The synthesised relation was again used to evaluate the behaviour of the expert system Pneumon-ia.

In a second approximation we considered whether the synthesised relation could be built not only as the synthesis of the fuzzy relations but also from a synthesised diagnosis. We have proven that there are several diagnosis that satisfies this condition, and studied a synthesis function for diagnosis that lead to the synthesised fuzzy relation (Torra, 1995a).

3. Future work

1) The results achieved suppose a fuzzy relation definition function of the form
\[ \rho(x,y) = \phi(\phi^{-1}(x) - \phi^{-1}(y)) \]. This family of functions can be further studied.

2) The test of the expert system has been done defining a distance between fuzzy relations. As the distance used has some drawbacks, a new distance should be studied.

3) Although the diagnosis of the experts are defined as mappings from possible solutions to linguistic labels, all the results suppose a monotonic mapping from the linguistic labels to the unit interval. An alternative analysis within the framework of linguistic labels is possible. This approach would use the operations and synthesis functions defined over linguistic labels introduced in (Delgado et al., 1993, Herrera et al., 1995).

4. References


