Available online at www.sciencedirect.com

ScienceDirect

Expert Systems with Applications 36 (2009) 423–433

Expert Systems with Applications

www.elsevier.com/locate/eswa

A semantic fuzzy expert system for a fuzzy balanced scorecard

Fernando Bobillo [a,*], Miguel Delgado [a], Juan Gómez-Romero [a], Enrique López [b]

[a] Department of Computer Science and Artificial Intelligence, E.T.S.I. Informática, University of Granada, C. Periodista Daniel Saucedo Aranda, 18071 Granada, Spain

[b] Economy and Business Management Department, University of León, Campus de Vegazana s/n, 24071 León, Spain

Abstract

Balanced scorecard is a widely recognized tool to support decision making in business management. Unfortunately, current balanced scorecard-based systems present two drawbacks: they do not allow to define explicitly the semantics of the underlying knowledge and they are not able to deal with imprecision and vagueness. To overcome these limitations, in this paper we propose a semantic fuzzy expert system which implements a generic framework for the balanced scorecard. In our approach, knowledge about balanced scorecard variables is represented using an OWL ontology, therefore allowing reuse and sharing of the model among different companies. The ontology acts as the basis for the fuzzy expert system, which uses highly interpretable fuzzy IF–THEN rules to infer new knowledge. Results are valuable pieces of information to help managers to improve the achievement of the strategic objectives of the company. A main contribution of this work it that the system is general and can be customized to adapt to different scenarios.

Keywords: Knowledge-based systems; Expert systems; Ontologies semantic web; Applied economy; Balanced scorecard; Fuzzy logic

1. Introduction

Knowledge management plays a key role in the search for success in the current business world. Increasing specialization and complexity of companies has given raise to the necessity of an integral management of own and foreign resources, which involves and generates huge amounts of valuable data. Empresarial intelligence must be consequently more a cornerstone of the corporative strategy than simply an amalgam of disperse tools and procedures, if decision processes are expected to be faced with guarantees in order to achieve a joint and balanced global performance.

Balanced scorecard (BSC) (Kaplan & Norton, 1992) is a decision support tool at the strategic management level which improves the satisfaction of the strategic objectives. Since it was proposed in the early 1990s, it has demonstrated its suitability to assist decision making in management.

Nevertheless current balanced scorecard-based systems suffer from two problems. Firstly, variables which are to be measured have associated vagueness, being much more natural to refer to their values using a linguistic label instead a numerical value as frequently is done. Secondly, data do not have an explicit representation of their semantics; ad hoc solutions are usually implemented for each problem, making developers duplicate efforts and users cope with their specific details.

Some solutions have been proposed to the first problem. Since fuzzy set theory and fuzzy logic (Zadeh, 1965) have proved to be successful in handling imprecise and vague knowledge, they have been combined with the BSC leads to fuzzy balanced scorecard (see Section 5 for details). However, such approaches also leave room for improvements in several aspects such as interpretability, modularity and accuracy. On the other hand, to the very well of our knowledge, there has not been any effort in the other direction. Thus we have represented balanced scorecard data using an ontology, which allows to add semantics to them

[*] Corresponding author. Tel.: +34 958243194; fax: +34 958243317.
E-mail addresses: fbobillo@decsai.ugr.es (F. Bobillo), mdelgado@ugr.es (M. Delgado), jgomez@decsai.ugr.es (J. Gómez-Romero), elopez@unileon.es (E. López).

0957-4174/$ - see front matter © 2007 Elsevier Ltd. All rights reserved.
doi:10.1016/j.eswa.2007.09.020

making easier knowledge base maintenance as well as reuse of components among different organizations.

In this paper we present a new approach to a fuzzy BSC which improves the state of art by extending the number of variables and perspectives. We also present a fuzzy expert system for this fuzzy BSC. Its knowledge base relies on an ontology and its inference system derives new knowledge from fuzzy rules. The system is general and reusable, so every company can personalize it by providing their own meaning for the linguistic labels defined over the variables (e.g. what they consider a "high" value of some variable) and their own rules. The results of the expert system are highly interpretable pieces of information ready to be incorporated to managers' decision making processes.

The remainder of this paper is structured as follows. Section 2 provides some preliminaries on the fundamental theoretical aspects underlying this paper: balanced scorecard, fuzzy logic and ontologies. In Section 3 we present our fuzzy balanced scorecard, describing the variables which take part in it. Implementation details are set out in Section 4. The description of our intelligent system starts by sketching the ontology and then we show how the rule-based engine computes the value of the output variables of the system. Section 5 evaluates our proposal with regard to the related work. Finally, some conclusions and ideas for future research are drawn in Section 6.

2. Background

This section provides some basic background about the topics covered in the paper: Section 2.1 quickly overviews the original balanced scorecard, Section 2.2 refreshes the basic ideas in fuzzy sets theory and fuzzy logic, and Section 2.3 recalls the notion of ontology.

2.1. The balanced scorecard

In 1992 Robert S. Kaplan and David P. Norton proposed the balanced scorecard (BSC) Kaplan and Norton (1992), a widely recognized tool to support decision making at the strategic management level which improves the satisfaction of the strategic objectives. The name reflects the objective of maintaining a balance "between short and long-term objectives, between financial and non-financial measures, between lagging and leading indicators, and between internal and external performance perspectives" (Kaplan & Norton, 1996).

The key innovation of the BSC is, as opposite to traditional approaches which only consider the financial data, to supplement this information with additional non-monetary measures. In words of the authors, "financial measures are inadequate, however, for guiding and evaluating the journey that information age companies must make to create future value through investment in customers, suppliers, employees, processes, technology, and innovation".

In particular, these authors consider four perspectives: *Financial* perspective obviously, measuring the financial performance of the company, *customer* perspective, measuring the satisfaction of the customers preferences, *internal business process* perspective, measuring internal business results against measures from financial and customer perspectives, and *innovation and learning* perspective, measuring the ability of the company to adapt to changes. A more detailed description of the perspectives is out of the scope of this paper. On the other hand, Section 3 depicts the perspectives that we consider.

Since the apparition of the BSC it has become an important field of theory and research. Many companies have successfully applied this tool and several variations to the original proposal have been investigated (for instance, see Section 5).

2.2. Fuzzy sets and fuzzy logic

This section briefly reviews fuzzy sets theory and fuzzy logic; for more details a good reference is (Klir & Yuan, 1995). Fuzzy set theory and fuzzy logic, proposed by (Zadeh, 1965), are acknowledged as an appropriate formalism for capturing imprecise and vague knowledge. While in classical set theory elements either belong to a set or not, in fuzzy set theory elements can belong to a certain degree. More formally, let X be a set of elements. A fuzzy subset A of X, is defined by a membership function $\mu_A(x)$ which assigns any $x \in X$ to a value between 0 and 1. As in the classical case, 0 means no-membership and 1 full membership, but now a value between 0 and 1 represents the extent to which x can be considered as an element of X.

All crisp set operations are extended to fuzzy sets. The complement, intersection and union set operations are performed by a negation function, a t-norm function (typically, the minimum) and t-conorm function (typically, the maximum) respectively.

Several membership functions can be used in the definition of a fuzzy set. Some of the most used are the triangular and the trapezoidal function. A triangular function $tri_{\alpha,\beta,\gamma}(x)$ (see Fig. 1a) is defined over the set of non-negative reals $\mathbb{R}^+ \cup \{0\}$ with $\alpha \leqslant \beta \leqslant \gamma$ being real numbers. A trapezoidal function $trz_{\alpha,\beta,\gamma,\delta}(x)$ is defined over the set of non-negative reals $\mathbb{R}^+ \cup \{0\}$ as in Fig. 1b, with $\alpha \leqslant \beta \leqslant \gamma \leqslant \delta$ being real numbers. Note that a triangular function $tri_{\alpha,\beta,\gamma}$ can be represented using a trapezoidal function $trz_{\alpha,\beta,\beta,\gamma}(x)$.

One of the most important features of fuzzy logic is its ability to perform approximate reasoning (Zadeh, 1973), which involves inference rules with premises, consequences or both of them containing fuzzy propositions. Fuzzy rule-based systems have some advantages over other formalisms: they provide a natural representation for human knowledge as well as a very interpretable model (since the semantics of the rules can easily be understood even for not experts users), are simpler, cheaper and more robust than their crisp versions and, last but not least, have shown to behave very well in practical applications.

A fuzzy IF–THEN system consists of a rule base (a set of IF–THEN rules) and a reasoning algorithm performing

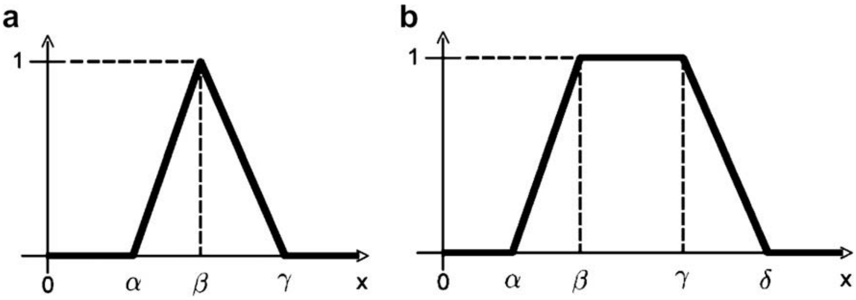


Fig. 1. (a) Triangular membership function; (b) trapezoidal membership function.

an inference mechanism. In general, the input of the system is the current value for the input variable and the output is a fuzzy set, which can be defuzzified into a single value. In a fuzzy IF–THEN rule, its antecedents, consequences or both are fuzzy. Fuzzy IF–THEN rules are fired to a degree which is a function of the degree of match between their antecedent and the input. The deduction rule is generalized modus ponens. Roughly speaking, given a rule "IF A THEN B", where A and B are fuzzy propositions, it is possible from a premise "A'" which matches A to some degree, to deduce "B'", which is similar to B.

One of the most popular IF–THEN systems is the Mamdani model (Mamdani & Assilian, 1975). In a Mamdani model, fuzzy rules have the form IF X_1 IS A_1 AND $\cdots$ AND X_n IS A_n THEN Y IS B, where for all $i = 1, \ldots, n, A_i$ and B are linguistic values defined by fuzzy sets on universes of discourse X_i and Y, respectively.

For every clause in the antecedent of the rule, the matching degree between the current value of the variable and the linguistic label in the rule is computed (typically, using the minimum or another t-conorm). If there exist several clauses, they are aggregated into a firing degree, using a fuzzy logic operator (typically, the maximum). Then, this firing degree is used for modifying the consequent of the rule using some function (typically the minimum). Sometimes this function is referred in the literature as an implication function, but this is a misleading term which should be avoided (e.g. minimum is not an implication function) (Hájek, 1998).

Rules are fired using an inference algorithm such as Rete (Forgy, 1982). The computed consequences related to the same variable are aggregated (typically, using the maximum). Then, the output variables can be defuzzified. A defuzzified number is usually represented by the centroid of gravity (COG), which can be determined using the moment of area method defined as $\mathrm{COG} = (\int_X x\mu_B(x)\mathrm{d}x / \int_X \mu_B(x)\mathrm{d}x)$, where $\mu_B(x)$ is the aggregated value of the fuzzy variable B over the universe of discourse Z.

2.3. Ontologies

An ontology is a "formal, explicit specification of a shared conceptualization" as defined by Studer, Benjamins, and Fensel (1998). They represent the concepts and the relationships in a domain promoting interrelation with other models and automatic processing. Ontologies are considered to be a proper mechanism to encode information in modern knowledge-intensive applications, so they have become one of the most used knowledge representation formalism. They allow the enrichment of data with semantics, enabling automatic verification of data consistency and making easier knowledge base maintenance as well as the reuse of components.

An example of this capacity to add semantics to data is the Semantic Web, an "extension of the current web where resources are well described using logic-based languages in order to allow automatic processing" (Berners-Lee, Hendler, & Lassila, 2001). The key to reach this Semantic Web is the use of metadata to annotate resources; thus, software agents would be able to search, locate, discover, or link documents better than today lexical-search engines. As a consequence, SW researchers are very interested in formalisms for creating metadata to be associated to web resources. Here is where ontologies play a fundamental role: they have been included as the main representation formalism in the core of the layered architecture of the Semantic Web.

Nowadays, the World Wide Web Consortium (W3C) standard language for ontology representation is OWL (McGuinness & van Harmelen, 2004). OWL consists of three sublanguages of increasing expressive power, namely OWL Lite, OWL DL and OWL Full. While OWL Lite and OWL DL are almost equivalent to $\mathcal{SHIF}(\mathcal{D})$ and $\mathcal{SHOIN}(\mathcal{D})$ Description Logics (Horrocks & Schneider, 2004) (and thus remain complete and decidable), OWL Full is incomplete and undecidable.

3. A fuzzy balanced scorecard

In the traditional formulation of the BSC, most system variables are affected by imprecision and vagueness but they are represented using numerical values. Even in the cases that there is no vagueness associated, the graduality of the variables enriches the expressivity (Dubois, Godo, Prade, & Esteva, 2005, Chapter 40). From these observations it seems natural to express BSC values using linguistic labels, which can be achieved by integrating fuzzy logic with BSC methodology.

The basic requirements for a fuzzy BSC are the following:

- *Interpretability*. It is expected that the BSC provides as a result the value of some output variables, which will guide a decision making process. This decision making process can be more profitable if the system is easily interpretable, that is, if knowledge and inference rules are represented in such a way that their meaning can be intuitively apprehended. Interpretability results in a better understanding of how the system works and hence it is easier to decide how to change input in order to accomplish a better satisfaction of the objectives.
- *Modularity*. By modularity we mean that variables belonging to a given perspective should not have an effect in the objective variables of a different perspective. Of course, this is not possible in all BSC models though it has some interesting advantages: there is no need to load simultaneously in memory the whole set of variables during the inference process and, as a consequence, the system becomes more scalable, since the addition of new variables has a smaller impact on the performance of the system.
- *Reusing*. Although a BSC is specific for a company, it is very likely that different companies can share several components. Adding precise semantics to the data facilitates the reuse of components. Hence, it is possible to build a general model for a fuzzy BSC which every company can instantiate and customize in order to fit its own needs. It also facilitates the understanding of the system as well as the maintaining, by making explicit the knowledge.

In our proposal, ontologies are used for asserting statements about BSC knowledge. In general, adding ontologies has several advantages: enables a flexible representation of the inference rules and the knowledge, reduces human intervention in maintenance, allows an automatic verification of data consistency, promotes modularity and enhances scalability and future reuse. Accordingly, they are appropriate to fulfil these requirements.

Following the guidelines of some experts in the area, we have extended the four perspectives initially proposed by Kaplan and Norton in order to achieve a better understanding of the company and to take into account measures which at that time were not as important as now, such as the environmental impact. The new perspectives include pieces of information about the environment, the quality of the products, the vendors and the company staff. Currently, we are considering 183 variables grouped in eight perspectives:

- *Customer*. This perspective aims to evaluate customers' satisfaction with the company and their perceptions of the activities.
- *Environmental*. The objective is to understand the importance of the damage caused to the environment by the actions of the company, as well as the possible consequences (administrative, commercial, legal, etc.), promoting company awareness and an improvement of the policy. In this context issues like for example pollution levels, company efforts for reducing it or consumption of natural resources become relevant.
- *Financial*. This perspective groups variables which have traditionally been used to indicate the level of satisfaction of the financial objectives, as well as the economical quantification of the decisions made. For instance, it includes risk assessment, rentability, market share and cost–benefit data.
- *Innovation*. This perspective aims to evaluate the ability of learning of the company, covering inversions in R&D, indicators of the social responsibilities or policies for employee training.
- *Business processes*. The majority of the key factors in the competitiveness lie in the company operations, such as the quality, flexibility, delivery times and corporate expenditures. This perspective pays attention to the production processes, focusing on the own process rather than on its operations.
- *Quality*. This perspective aims to make aware of the financial impact of the quality in the results of the company, helping to identify the weaknesses and so the opportunities of improvement. These variables allow to analyze if quality costs are well distributed among different areas, to reduce the quality costs caused by failures and to promote the knowledge about improvement areas as well as the evaluation of the specific actions carried out.
- *Vendor*. The objective of this perspective is to reach a high level of synchronization between the necessities of the company and the possibilities which the vendors offer, e.g. costs of storing raw materials or precision (of the vendors) in the delivery of the products.
- *Staff*. This perspective provides information about the fulfilment of the objectives related to the staff: effectiveness of the policies, structure, productivity and, in general, about the capacity of the organization to give response from scenario changes. This perspective is divided into seven sub-categories: *Composition*, *Background*, *Labour environment*, *Leadership*, *Salary*, *Security* and *Stability*.

4. Implementation: fBSC system

In this section we describe an ontology-based fuzzy expert system, our implementation of the fuzzy balanced scorecard (fBSC). Fig. 2 illustrates the architecture of the fBSC system. It is mainly formed by (i) an ontology, which acts as the basis for the maintenance of the knowledge base, and (ii) the inference system, responsible for assessing the value of the output variables.

4.1. fBSC ontology

In this section, we describe fBSCO (*Fuzzy Balanced ScoreCard Ontology*). Its aim is to represent the variables

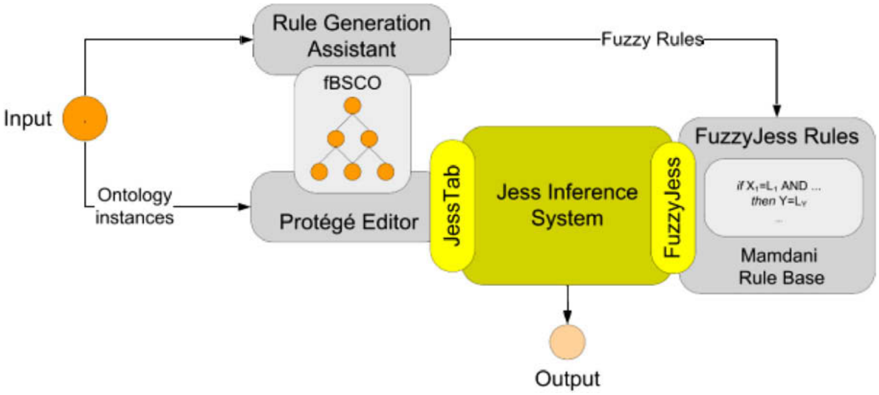


Fig. 2. Architecture of the expert system.

of our balanced scorecard described in Section 3, as well as the linguistic labels which can be associated to each of them.

The development of fBSCO has been guided by the METHONTOLOGY methodology (Gómez-Pérez, 1998) and carried out with the assistance of some experts in the fields of economy and company management.

To the very best of our knowledge, there does not exist another ontology for the BSC to be reused, so we ought have to start the development of the fBSCO from scratch. On the contrary, we have been able to reuse fuzzy knowledge representation ontology (FKRO) (Blanco, Martínez-Cruz, Marín, & Vila, 2005) for the representation of the linguistic labels. FKRO ontology represents different types of fuzzy knowledge (e.g. interval fuzzy sets, possibility distributions or trapezoidal membership functions), with the aim of enabling their reuse in an abstract way, without any need to know particular implementations details.

We have chosen to implement it in the standard language OWL (McGuinness & van Harmelen, 2004). The used editor has been Protégé enhanced with the OWL Plugin. Protégé is a graphical tool for ontology editing and knowledge acquisition developed at Stanford University (Noy et al., 2001). The OWL Plugin extends Protégé allowing to edit OWL ontologies, to access description logic reasoners and to acquire instances for semantic markup (Knublauch, Fergerson, Noy, & Musen, 2004). Protégé + OWL also provides a Java API for working with OWL and RDF models.

fBSCO comprises two main classes: `LinguisticLabel` and `SimeVariable` (see Fig. 3 for an excerpt from the ontology).

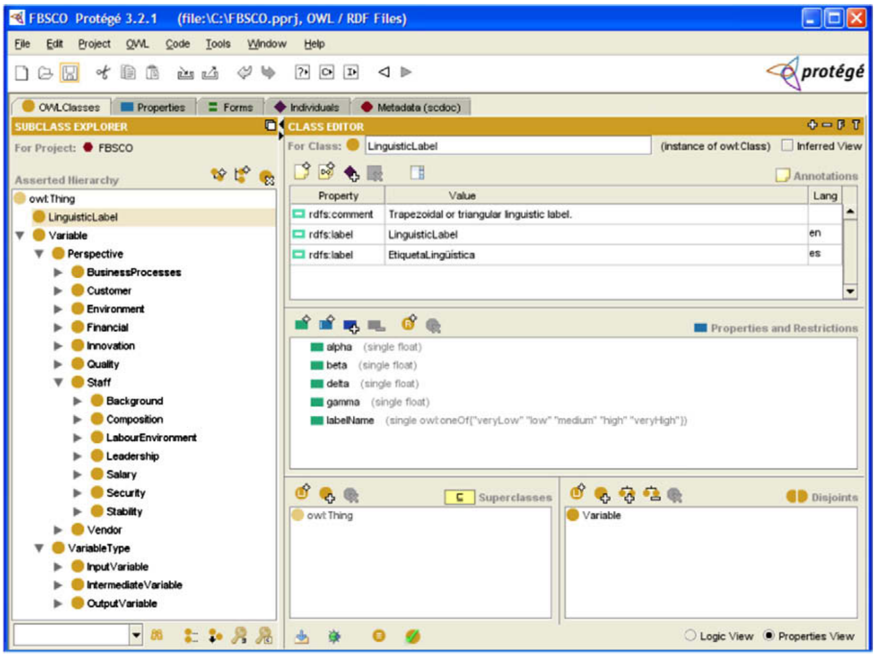
`LinguisticLabel` represents a linguistic label, a fuzzy set with a membership function specificated by a trapezoidal function. It has five datatype properties – the parameters of the membership function (`alpha`, `beta`, `gamma` and `delta`) – and a text property (`labelName`). We note that this representation corresponds to the representation for trapezoidal variables of what the authors call type 2 fuzzy variables in the FKRO ontology.

On the other hand, `SimeVariable` contains the fuzzyBSC variables, which are organized following two criteria: type of perspective and type of variable:

- `Perspective` class structures the variables in the eight disjoint perspectives described in Section 3: `CustomerPerspective`, `EnvironmentalPerspective`, `FinancialPerspective`, `InnovationPerspective`, `ProductionProcesses Perspective`, `QualityPerspective`, `StaffPerspective` and `VendorPerspective`. `StaffPerspective` comprises seven disjoint classes: `Composition`, `Background`, `LabourEnvironment`, `Leadership`, `Salary`, `Security` and `Stability`.
- `Variable` class structures the variables in three disjoint categories: `InputVa riable`, `IntermediateVariable` and `OutputVariable`, which represent (respectively) input variables which the user must provide to the system, intermediate variables internally used during the computation of the output and output variables, which are shown to the user as a response.

The definition of every system variable includes two necessary conditions: it must be a descendant of the `Perspective` class and a descendant of the `Variable` class. For example, `Doctors` (stating for the percentage of doctors in the staff) is an input variable belonging to the formation perspective. Abusing of the language, we will refer to a final descendant of the `Variable` class with the term *variable class*.

These variables have two datatypes properties: `hasValue` and `hasLabel`. The former represents the numerical value of a variable, which changes over the life cycle of the system. In the case of an input variable, this value is introduced by the user. In an output variable, it is the result of the defuzzification process. In intermediate variables, the value is used during the computation of the output variables but it is not updated in the ontology instances. Concerning the latter, every variable class is related to one or more linguistic labels using the object property `hasLabel`. This relation establishes a link among a variable and the



For Project: FBSCO

- owl:Thing
  - LinguisticLabel
  - Variable
    - Perspective
      - BusinessProcesses
      - Customer
      - Environment
      - Financial
      - Innovation
      - Quality
      - Staff
        - Background
        - Composition
        - LabourEnvironment
        - Leadership
        - Salary
        - Security
        - Stability
      - Vendor
    - VariableType
      - InputVariable
      - IntermediateVariable
      - OutputVariable

For Class: LinguisticLabel (instance of owl:Class) □ Inferred View

Property	Value	Lang
rdfs:comment	Trapezoidal or triangular linguistic label.	
rdfs:label	LinguisticLabel	en
rdfs:label	EtiquetaLingüística	es

alpha (single float)
beta (single float)
delta (single float)
gamma (single float)
labelName (single owl:oneOf{"veryLow" "low" "medium" "high" "veryHigh"})

Superclasses: owl:Thing

Disjoints: Variable

Fig. 3. An excerpt from the fBSC ontology in the Protégé editor.

definitions of all the linguistic labels which are assigned to it. These data are specific for each company and are not expected to be modified unless the system is reconfigured.

Before running the system for the first time, the user would be responsible to adapt the definition of the linguistic label class to his particular believes (e.g. what is 'High' or 'Low' for him) for every variable class. It would be desirable to relate (using the `hasLabel` property) each instance of a variable with some linguistic label classes, as the latter are not expected to change during the life of the system. Unfortunately, relating individuals and classes is not a trivial task in OWL. This issue has been considered in Noy (2005), where several recommendations are provided. The simplest solution is allowing these relations among individuals and classes but, unfortunately, this forces an ontology to be OWL Full, and thus undecidable.

In our particular case, reasoning capabilities are a crucial requirement, so we have chosen another option: to create (and relate) several `LinguisticLabel` instances for each variable instance in order to represent the labels assigned to it. Thus, instances of `InputVariable` and `IntermediateVariable` have three linguistic label instances associated (`high`, `medium` and `low`), while instances of `OutputVariable` have five (`veryHigh`, `high`, `medium`, `low` and `veryLow`). In this manner, for instance, a instance of `Doctors` would have associated a current numeric value with tt hasValue (e.g. 10) and three linguistic labels with `hasLabel`, `Low` (e.g. $trz_{0,0,0.5,1.5}$), `Medium` (e.g. $trz_{0.5,1.5,3.1,4}$) and `High` (e.g. $trz_{3.1,4,10,10}$). This option is feasible due to the fact that every company is expected to populate final variable classes with just one instance. In the example, this would correspond to having only one instance of class `Doctors` for a given company.

According to this formulation, fBSCO instances materialize the BSC for a given company, i.e. make concrete the generic metamodel described by the fBSCO classes. Before using the fBSC system, managers are expected to guide the definition of `LinguisticLabel` variables, which we have called the *static materialization*, as values of these instances are not likely to change for a company. On the other hand, instances of input variables representing current company values are as well to be created, composing the so called *transient materialization*, as these instances may change in successive evaluation processes. Intermediate and output variables should not be instantiated and, if done, these values will be ignored.

At the moment, the ontology has 888 classes and seven object and datatype properties. Of course, it also has some annotation properties and some class axioms. For example, disjointness among classes, cardinality constraints (e.g. the cardinality of `hasLabel` is three for input and intermediate variables, and five for output variables) or type of the properties (for example, the properties `alpha`, `beta` and `gamma` are mandatory for every `LinguisticLabel`, while `delta` is optional in order to allow triangular labels, which do not need it). Another important point to remark is the multilingual nature of the ontology, in English and Spanish, taking advantage of the `xml:lang` attribute in some annotation properties such as `rdfs:label` and `rdfs:comment`.

4.2. The inference system

In this section we will describe how the fBSC system works. As explained in previous section, fBSC provides a generic framework, so the first step is to create a suitable knowledge base reflecting the parameters of the company to be analyzed. That means creating instances of the fBSCO and fuzzy rules to calculate outputs. In our case, the values for the linguistic labels have been defined with the help of experts. Having populated the KB, an inference system is applied in order to compute the output of the system. We have chosen to implement it using Jess, FuzzyJess and JessTab.

Jess (Java Expert System Shell) is a popular rule-based language (Friedman-Hill, 2003) which has been successfully applied to expert systems. It is simple but powerful enough to allow its use in real applications. It can also be integrated with external Java programs through a well-defined Java API. Jess inference engine is mostly Rete-based (Forgy, 1982) forward chaining, but backward chaining is supported as well. FuzzyJess (Orchard, 2001) is an extension of Jess which enables the use of fuzzy rules. It was developed through integration of Jess and NRC FuzzyJ Toolkit, a set of Java classes that provide the capability for handling fuzzy concepts and reasoning. Finally, JessTab (Eriksson, 2003) is a plugin for Protégé which enhances its integration with Jess: it enables the Jess engine to run inside the Protégé framework and lets users build in Protégé knowledge bases with Jess programs and rules. JessTab is not directly compatible with FuzzyJess, but it is possible to slightly modify its free source code so it uses the FuzzyJess engine (Orchard, 2005) instead the Jess engine.

The rulebase contains Mamdani fuzzy rules of the form IF `Var1` IS `Label1` AND ...AND `VarN` IS `LabelN` THEN `VarZ` IS `LabelZ`, where `Var1`···`VarN` are input or intermediate variables, `VarZ` is an intermediate or output variable and `Label1`···`LabelN`,`LabelZ` are linguistic labels. These rules allow users to interpret their semantics in a very easy way, even if they are not experts. Hence, they can help the user to understand the global system functioning and consequently to be able to improve the performance of the company.

The rule generation assistant is a semi-automatic tool which helps the user during the definition of the rules. The user can navigate through the ontology and select several variables for belonging to the antecedent of a rule (they have to be either input variables or intermediate variables). Next, it is possible to select a variable for the consequent of the rule (intermediate or output depending on the antecedent). Since input and intermediate variables have three linguistic labels and there are two variables in the antecedent, there exist nine rules in order to cover all the possibilities for the variable appearing in the consequent (twenty seven in some exceptional cases). For example, Table 1 shows how the input variables `Doctors` and `Graduates` can be used to determine the value of the intermediate variable `HSLevel`. The rules in the table should be read as e.g. IF `Doctors` IS `Low` AND `NumberOf Graduates` IS `High` THEN `HSLevel` IS `Medium`. A such table is presented to the user, so he can specify abstractly, for every combination of the linguistic labels in the antecedent variables, the linguistic value of the consequent. It is also possible to discard some combination of antecedents if the user considers them to be irrelevant. Finally, a set of Jess rules –which we call *family of rules*– is generated. Using this tool the risk of forgetting one of the possible cases is avoided and there is no need to know the concrete syntax for the Jess rules.

Table 1
Rules for the HSLevel variable

Graduates	Doctors		
	Low	Medium	High
Low	Low	Low	Medium
Medium	Low	Medium	High
High	Medium	High	High

Rules in a family have some special features:

- In general, the antecedents of the rules contain two clauses, although exceptionally the number of clauses may be three.
- Every variable in the antecedent can only be used in one family of rules (i.e. they take part directly in the computation of only one variable).
- Every variable in the consequent can only be used in one family of rules (i.e. the computation of its value depends directly of only two – or exceptionally three – variables).

For every perspective, the inference systems performs the following operations (see Fig. 4):

(1) The instances of the ontology are translated into facts in the knowledge base. These can be done by using the JessTab function `mapclass`.
(2) Every input variable is fuzzified, matching their numerical value (`hasValue`) with their three associated linguistic labels (`hasLabel`) as the Jess rule in Table 2 shows. Notice that another possibility would be to provide directly a fuzzy linguistic input.

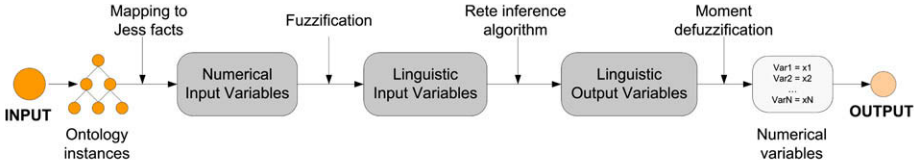


Fig. 4. Inference module.

Table 2
Example Jess rule fuzzifying the current values of the input variables

```
(defrule ruleFuzzify
   (object
      (is-a-name ?name)
      (hasValue ?value)
      (hasLabel ?label1 ?label2 ?label3)
   )

   =>
   ; label1 is the linguistic label High
   (assert
      (fact
         (class ?name)
         (value high)
         (degree
            ((new SingletonFuzzySet ?value)
            maximumOfIntersection
               (new TrapezoidFuzzySet
                       (slot-get ?label1 alpha)
                       (slot-get ?label1 beta)
                       (slot-get ?label1 delta)
                       (slot-get ?label1 gamma)
               )
            )
         )
      )
   )

   ; label2 is the linguistic label Medium
   ...
   ; label3 is the linguistic label Low
   (assert
      (fact
         (class ?name)
         (value low)
         (degree
            ((new SingletonFuzzySet ?value)
            maximumOfIntersection
               (new TrapezoidFuzzySet
                       (slot-get ?label3 alpha)
                       (slot-get ?label3 beta)
                       (slot-get ?label3 delta)
                       (slot-get ?label3 gamma)
               )
            )
         )
      )
   )
)
```

(3) The inference algorithm is run (Jess function `run`, see (Friedman-Hill, 2003) for details). The values in the previous step can fire some rules. From the input variables, intermediate variables are calculated. Then, intermediate variables are used to compute another intermediate variables and, finally, output variables. For instance, the Jess rules in Table 3 implement the fuzzy rule IF `Doctors` IS `High` AND `NumberOf Graduates` IS `High` THEN `HSLevel` IS `High`. If the consequent of the rule has not been inferred, rule *ruleHSLevel1a* creates it; whereas if it already exists rule *ruleHSLevel1b* modifies its value.

(4) Output variables are defuzzified using the centroid of area (in FuzzyJess, function `momentDefuzzify`) and then are shown to the user. We note in passing that FuzzyJess calls COA to the Center of Area, but we use this name for the Centroid of Area. These variables are expected to guide the user in a subsequent decision-making process.

It is interesting to note that the linguistic labels may overlap, so in the assessment of one of these variables more than one rule can be fired and contribute to the final value.

Despite of the number of rules is quite large, the rules involving variables from a certain perspective are independent from the rules corresponding to the other perspectives, so the value of the output variables can be assessed sequentially for each perspective – but without a need of maintaining simultaneously in memory an intractable number of facts and rules – or in independent parallel processes.

4.3. An example

We will illustrate the calculation of the output variables by showing an example. Let us consider the `Background` perspective with one output variable, `TrainingPlan`, representing the quality of the evaluation of the training plan for the employees of the company. As every final variable, it has five associated labels, in this case: `VeryLow` $= trz_{0,0,0,2}$, `Low` $= trz_{0,2,3.5,4.5}$, `Medium` $= trz_{3.5,4.5,5.5,6}$, `High` $= trz_{5.5,6,7,8}$ and `VeryHigh` $= trz_{7,8,9,10}$.

Firstly, we show how the system works with input variables. For simplicity we focus on two of them, `Doctors` and `Graduates`. Let us suppose that the percentage of doctors in the staff is 3.5 and that the percentage of graduates is 25. `Doctors` has three labels associated: `Low` $= trz_{0,0,0.5,1.5}$, `Medium` $= trz_{0.5,1.5,3.1,4}$ and `High` $= trz_{3.1,4,10,10}$. `Graduates` has three labels associated: `Low` $= trz_{0,0,3,16}$, `Medium` $= trz_{3,16,22,26}$ and `High` $= trz_{22,26,30,30}$. Hence:

$$\mu_{DoctorsIsLow} = trz_{0,0,0.5,1.5}(3.5) = 0$$
$$\mu_{GraduatesIsLow} = trz_{0,0,3,16}(25) = 0$$
$$\mu_{DoctorsIsMedium} = trz_{0.5,1.5,3.1,4}(3.5) = 0.5556$$
$$\mu_{GraduatesIsMedium} = trz_{3,16,22,26}(25) = 0.25$$
$$\mu_{DoctorsIsHigh} = trz_{3.1,4,10,10}(3.5) = 0.4444$$
$$\mu_{GraduatesIsHigh} = trz_{22,26,30,30}(25) = 0.75$$

Assuming the rules in Table 1, four rules are fired (where the variable `HSLevel` stands for the high schooling level):

(1) IF `Doctors` IS `Medium` AND `Graduates` IS `Medium` THEN `HSLevel` IS `Medium`
(2) IF `Doctors` IS `Medium` AND `Graduates` IS `High` THEN `HSLevel` IS `High`
(3) IF `Doctors` IS `High` AND `Graduates` IS `Medium` THEN `HSLevel` IS `High`
(4) IF `Doctors` IS `High` AND `Graduates` IS `High` THEN `HSLevel` IS `High`

Table 3
Example Jess rules computing the value of an intermediate variable

```
(defrule ruleHSLevella
   (fact (class ‘‘Doctors’’) (value high) (degree ?cfl))
   (fact (class ‘‘Graduates’’) (value high) (degree ?cf2))
   ?id <- (fact (class ‘‘HSLevel’’) (value high) (degree ?cf3))
   (test (< ?cf3 (min ?cfl ?cf2)))
=>
   (modify ?id (degree (min ?cfl ?cf2)))
)

(defrule ruleHSLevellb
   (fact (class ‘‘Doctors’’) (value high) (degree ?cfl))
   (fact (class ‘‘Graduates’’) (value high) (degree ?cf2))
   (not (fact (class ‘‘HSLevel’’) (value high)))
   (test (neq (min ?cfl ?cf2) 0.0))
=>
   (assert (fact (class ‘‘HSLevel’’) (value high)
      (degree (min ?cfl ?cf2))))
)
```

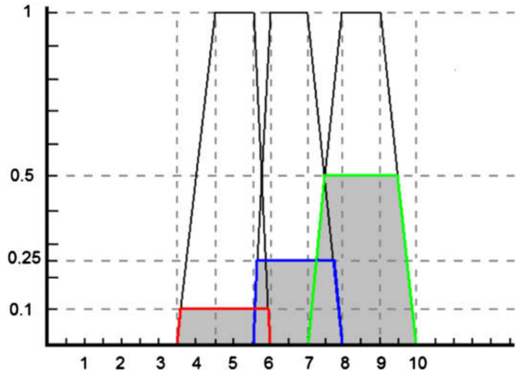


Fig. 5. Value of the TrainingPlan variable before defuzzification.

Since $\mu_{DoctorsIsMedium} = 0.5556$ and $\mu_{GraduatesIsMedium} = 0.25$, the consequence of the first rule is obtained using the minimum: $\mu_{HSLevelIsMedium} = \min\{0.5556, 0.25\} = 0.25$. Similarly, rules 2, 3 and 4 give $\mu_{HSLevelIsHigh} = 0.5556$, $\mu_{HighSchoolingLevelIsHigh} = 0.25$ and $\mu_{HSLevelIsHigh} = 0.4444$, respectively; values which are aggregated into a single value using the maximum: $\mu_{HSLevelIsHigh} = \max\{0.5556, 0.25, 0.4444\} = 0.5556$.

These intermediate values are used together with another variables to compute the value of the final variable `TrainingPlan`. As every final variable, it has five associated labels: `VeryLow` $= trz_{0,0,0,2}$, `Low` $= trz_{0,2,3.5,4.5}$, `Medium` $= trz_{3.5,4.5,5.5,6}$, `High` $= trz_{5.5,6,7,8}$ and `VeryHigh` $= trz_{7,8,9,10}$. Once fired the corresponding rules, intermediate variables infer the following consequences: $\mu_{TrainingPlanIsMedium} = 0.1$, $\mu_{TrainingPlanIsHigh} = 0.25$ and $\mu_{TrainingPlanIsVeryHigh} = 0.5$.

Next, we compute the intersection between this values and the corresponding labels. Fig. 5 shows the intersection of `Medium` (first function on the left), `High` (on the middle) and `VeryHigh` (on the right). Finally, the centroid of the dark area is computed, producing an output value $\mu_{TrainingPlan} = 7.6162$ (see Fig. 6).

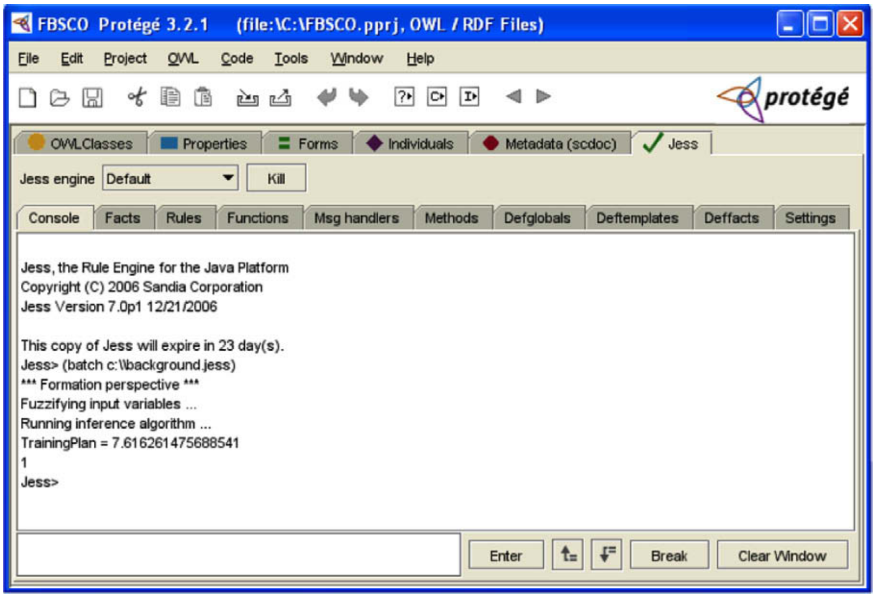
5. Related work

To the very best of our knowledge, this is the first attempt to provide a semantic framework for the balanced scorecard. However, the idea of extending BSC with fuzzy logic is not completely new. In this section we will shortly review the related work. As we will see, none of the existing proposals fully satisfy the requirements described in Section 3.

The oldest work we are aware of is due to Haase, who in 2000 proposed a fuzzy Balance Scorecard and implemented its proposal in the ActiveScoreCard system (Haase, 2000). Later on, Chou and Liang applied a fuzzy BSC to shipping companies (Chou & Liang, 2001). Together with other authors, they also worked with ports performance (Su, Liang, Liu, & Chou, 2003). More recently, Madrigal claimed for a fuzzy balanced scorecard, but no concrete methodology nor implementation were proposed though (Madrigal, 2005). Nissen also considered a fuzzy balance scorecard, analyzed the new modeling process and presented a prototypical implementation (Nissen, 2006). We also note the existence of previous works in the German literature (Nissen, 2005b; Nissen, 2005a). Finally, Pochert also considered an alternative model (Pochert, 2005).

As stated in Section 4.1, a very interesting property of our system is the modularity of the perspectives, which makes easier to increase the knowledge base with new variables. This does not hold in Chou and Liang (2001), Su et al. (2003), Madrigal (2005) and Nissen (2006), so our system provides more facilities for the scalability of the system.

Some of the existing works do not specify the number of variables they are dealing with. Haase (2000), Chou and Liang (2001) and Su et al. (2003) consider 45, 35 and 31 and variables respectively. Thus, the number of variables in our fuzzy BSC is much higher (as already said in Section 3, we have 183 variables). The system is able to represent the company's functioning with more accuracy – using


FBSCO Protégé 3.2.1 (file:\C:\FBSCO.pprj, OWL / RDF Files)
File Edit Project OWL Code Tools Window Help
protégé
OWLClasses Properties Forms Individuals Metadata (scdoc) Jess
Kill
Console Facts Rules Functions Msg handlers Methods Defglobals Deftemplates Deffacts Settings
Enter Break Clear Window

Jess engine Default

```
Jess, the Rule Engine for the Java Platform
Copyright (C) 2006 Sandia Corporation
Jess Version 7.0p1 12/21/2006

This copy of Jess will expire in 23 day(s).
Jess> (batch c:\\background.jess)
*** Formation perspective ***
Fuzzifying input variables ...
Running inference algorithm ...
TrainingPlan = 7.616261475688541
1
Jess>
```

Fig. 6. Output of the system after defuzzification.

more variables – without a significant performance lost, which would not be possible if it was not modular.

Our system is also more easily to be understood by people due to the use of linguistic variables and fuzzy IF–THEN rules, which have proved to be more interpretable tools to represent fuzzy knowledge and inference rules than fuzzy numbers and other formalisms. Some other proposals rely on more complex representations, such as Chou and Liang (2001), which uses an entropy-based method to perform decision making, Su et al. (2003), where arithmetic operations on triangular numbers are used, and Madrigal (2005), who, although does not propose a full reasoning methodology, considers as an example arithmetic operations on interval fuzzy numbers.

Regarding linguistic variables, Su et al. (2003) does not use them, but the specific value of a given variable is fuzzified into a triangular fuzzy number without any linguistic label. Neither does (Haase, 2000), which works directly with truth degrees in $[0,1]$. It is worth to note that (Chou & Liang, 2001) defines five linguistic labels (`VeryGood`, `Good`, `Medium`, `Bad` and `VeryBad`) for every variable but, however, the definition of the labels is the same for every variable. Hence, our proposal is more accurate, allowing to distinguish what we mean by "*good*" whether referred to a perspective p_1 or to a different perspective p_2.

We also note that Madrigal (2005) makes a distinction between fuzzy and crisp variables. In the same spirit, he does not fuzzify financial variables. However, all variables in our system are expressed using linguistic labels over a numerical domain. In fact, as already claimed, the graduality of the variables makes the representation more expressive and does not need to convey any idea of uncertainty.

To end, we recall that, as pointed out by Nissen (2006), Pochert (2005) does not provide enough accuracy in the results, because it follows a bottom-up strategy and assumes that only one rule is applied for every combination of input variables.

6. Conclusions and future work

This paper proposes a semantic fuzzy expert system for the balanced scorecard. We have presented a novel approach to a fuzzy balanced scorecard, extending the number of variables and perspectives with respect to previous works. We have built an OWL ontology to encourage reusing and sharing of this model. We have also developed a fuzzy expert system with a knowledge-based relying on this ontology and an inference system using fuzzy IF–THEN rules to derive new knowledge. The output of the expert system is appropriate to take part in a decision making process which improves the achievement of the strategic objectives of the company. The whole system is very general and may be used by different companies, which can customize it to make it suitable to their own needs.

An interesting direction of further research would be to use a more standard representation for fuzziness, with a more solid theoretical basis such as fuzzy concrete domains (Straccia, 2005). Fuzzy concrete domains were proposed in the context of some fuzzy extension of description logics and offer the possibility to represent some explicit membership functions for fuzzy sets such as triangular and trapezoidal. However, for the concerns this paper, it would be sufficient to have a crisp description logic with fuzzy concrete domains, as proposed in Schockaert et al. (2006).

Another interesting consideration would be the representation of the fuzzy rules using some of the recently proposed fuzzy semantic rule languages, such as fuzzy SWRL (Pan, Stoilos, Stamou, Tzouvaras, & Horrocks, 2006) or fuzzy RuleML (Damasio, Pan, Stoilos, & Straccia, 2006), as an alternative to FuzzyJess.

Acknowledgements

This work was partially supported by the Spanish Ministerio de Educación y Ciencia (project TIN2006-15041-C04-01). Fernando Bobillo holds a FPU scholarship from the Spanish Ministerio de Educación y Ciencia. Juan Gómez-Romero holds a scholarship from Consejería de Innovación, Ciencia y Empresa (Junta de Andalucía).

References

Berners-Lee, T., Hendler, J., & Lassila, O. (2001). The semantic web. *Scientific American, 284*(5), 34–43.

Blanco, I., Martínez-Cruz, C., Marín, N., & Vila, M. A. (2005). About the use of ontologies for fuzzy knowledge representation. In *Proceedings of the joint 4th EUSFLAT & 11th LFA conference (EUSFLAT-LFA'2005)* (pp. 106–111).

Chou, T.-Y., & Liang, G.-S. (2001). Application of a fuzzy multi-criteria decision-making model for shipping company performance evaluation. *Maritime Policy & Management, 28*(4), 375–392.

Damasio, C., Pan, J., Stoilos, G., & Straccia, U. (2006). An approach to representing uncertainty rules in RuleML. In *Proceedings of the 2nd international conference on rules and rule markup languages for the semantic web (RuleML-06)* (pp. 97–106).

Dubois, D., Godo, L., Prade, H., & Esteva, F. (2005). An information-based discussion of vagueness. In H. Cohen, & C. Lefebvre, (Eds.), *Handbook of categorization in cognitive science* (pp. 892–913).

Eriksson, H. (2003). Using JessTab to integrate Protégé and Jess. *IEEE Intelligent Systems, 18*(2), 43–50.

Forgy, C. (1982). Rete: A fast algorithm for the many pattern/many object pattern match problem. *Artificial Intelligence, 19*, 17–37.

Friedman-Hill, E. (2003). *Jess in action: Java rule-based systems*. Manning Publications Co.

Gómez-Pérez, A. (1998). Knowledge sharing and reuse. In J. Liebowitz (Ed.), *The handbook of applied expert systems*. CRC Press.

Haase, V. H. (2000). Computer models for strategic business process optimisation. In *Proceedings of the 26th Euromicro conference* (Vol. 2, pp. 254–260).

Hájek, P. (1998). Ten claims about fuzzy logic. *Soft Computing, 2*(1), 14–15.

Horrocks, I., & Schneider, P. F. P. (2004). Reducing OWL entailment to description logic satisfiability. *Journal of Web Semantics, 1*(4), 345–357.

Kaplan, R. S., & Norton, D. P. (1992). The balanced scorecard: Measures that drive performance. *Harvard Business Review*.

Kaplan, R. S., & Norton, D. P. (1996). *The balanced scorecard: Translating strategy into action*. Harvard Business School Press.

Klir, G. J., & Yuan, B. (1995). *Fuzzy sets and fuzzy logic: Theory and applications*. Prentice-Hall.

Knublauch, H., Fergerson, R. W., Noy, N. F., & Musen, M. A. (2004). The Protégé OWL plugin: An open development environment for semantic web applications. In *Proceedings of the 3rd international semantic web conference (ISWC 2004). Lecture notes in computer science* (Vol. 3298, pp. 229–243). Springer.

Madrigal, F. J. (2005). Los conjuntos borrosos en el balanced scorecard [in Spanish]. In *Segundo Congreso Internacional y Sexto Nacional de Administración, Contaduría y Mercadotecnia*.

Mamdani, E. H., & Assilian, S. (1975). An experiment in linguistic synthesis with a fuzzy logic controller. *International Journal of Man–Machine Studies, 7*(1), 1–13.

McGuinness, D. L., & van Harmelen, F. (2004). *OWL web ontology language overview*. <http://www.w3.org/TR/owl-features>.

Nissen, V. (2005a). Die fuzzy balanced scorecard als instrument der unternehmensführung [in German]. In L. Mönch & J. Beyer (Eds.), *Aspekte der Wirtschaftsinformatik: Methoden, Werkzeuge und Anwendungen* (pp. 71–90). SCS Publishing House.

Nissen, V. (2005b). Die fuzzy balanced scorecard [in German]. Technical Report 2005-01, Technische Universität Ilmenau (Germany), Institut für Wirtschaftsinformatik.

Nissen, V. (2006). Modelling corporate strategy with the fuzzy balanced scorecard. In *Proceedings of the symposium on fuzzy systems in computer science (FSCS 2006)* (pp. 121–138).

Noy, N. (2005). Representing classes as property values on the semantic web. <http://www.w3.org/TR/swbp-classes-as-values/>.

Noy, N. F., Sintek, M., Decker, S., Crubezy, M., Fergerson, R. W., & Musen, M. A. (2001). Creating semantic web contents with Protégé-2000. *IEEE Intelligent Systems, 16*(2), 60–71.

Orchard, B. (2005). Discussion on Jess mailing list. <http://www.mail-archive.com/jess-users@mailgate2.sandia.gov/msg00360.html>.

Orchard, R. (2001). Fuzzy reasoning in Jess: The FuzzyJ toolkit and FuzzyJess. In *Proceedings of ICEIS 2001* (Vol. 1, pp. 533–542).

Pan, J. Z., Stoilos, G., Stamou, G., Tzouvaras, V., & Horrocks, I. (2006). f-SWRL: A fuzzy extension of SWRL. *Journal of Data Semantics, VI*, 28–46 [Special issue on Emergent semantics].

Pochert, B. (2005). *Konzeption einer unscharfen Balanced Scorecard: Möglichkeiten der Fuzzyfizierung einer Balanced Scorcard zur Unterstützung des Strategischen Managements* [in German]. Cuvillier.

Schockaert, S., Victor, P., Houben, G.-J., Cornelis, C., Cock, M. D., & Kerre, E. (2006). Reflections on modelling vagueness in description logics. In P. C. G. da Costa, K. B. Laskey, K.J. Laskey, F. Fung, & M. Pool (Eds.), *Proceedings of the 2nd ISWC workshop on uncertainty reasoning for the semantic web (URSW 2006). CEUR workshop proceedings* (Vol. 218).

Straccia, U. (2005). Description logics with fuzzy concrete domains. In *Proceedings of the 21st Conference on Uncertainty in Artificial Intelligence (UAI-05)*, (pp. 559–568).

Studer, R., Benjamins, V. R., & Fensel, D. (1998). Knowledge engineering: Principles and methods. *Data & Knowledge Engineering, 25*(1–2), 161–197.

Su, Y., Liang, G.-S., Liu, C.-F., & Chou, T.-Y. (2003). A study on integrated port performance comparison based on the concept of balanced scorecard. *Journal of the Eastern Asia Society for Transportation Studies, 5*, 609–624.

Zadeh, L. A. (1965). Fuzzy sets. *Information and Control, 8*, 338–353.

Zadeh, L. A. (1973). Outline of a new approach to the analysis of complex systems and decision processes. *IEEE Transactions on Systems, Man and Cybernetics, 3*(1), 28–44.