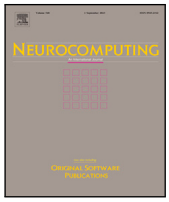
Neurocomputing 565 (2024) 126968

Contents lists available at ScienceDirect

Neurocomputing

journal homepage: www.elsevier.com/locate/neucom



A unified view of multi-grade fuzzy-set models in *J-CO-QL*$^+$

Paolo Fosci, Giuseppe Psaila *

University of Bergamo, Department of Management, Information and Production Engineering, Viale Marconi 5, Dalmine, 24044, (BG), Italy

ARTICLE INFO

Keywords:
Quick review of fuzzy-set models
Meta-model for multi-grade fuzzy sets
User-defined fuzzy-set models in *J-CO-QL*$^+$
Soft querying with multi-grade fuzzy-set models

ABSTRACT

The complexity of reality has driven the evolution of Fuzzy-Set Theory from the initial proposal made by Zadeh in 1965, towards more complex models. Moving from a quick survey of the evolution of Fuzzy-Set Theory, this paper highlights the aspects that are common to many Fuzzy-Set Models, in order to define a meta-model that is capable of providing a unified view to a wide variety of fuzzy-set models. In particular, this work focuses the attention on the family of "Multi-grade Fuzzy Sets", which are fuzzy sets characterized by more than one degree.

The lack of tools capable of querying the large amount of data that are nowadays available in NoSQL databases, has pushed us to devise the *J-CO* Framework: it is a platform-independent tool that is capable to manage, transform and query collections of *JSON* documents; the *J-CO* Framework relies on *J-CO-QL*$^+$, which is a high-level, general-purpose language with soft-querying capabilities. The latest advancements of *J-CO-QL*$^+$ allow for defining and exploiting user-defined Multi-grade Fuzzy-Set Models and Operators. In the paper, a case-study demonstrates the effectiveness of the *J-CO* Framework in performing a non-trivial soft query based on a Multi-grade Fuzzy-Set Model defined by the user.

1. Introduction

In the digital society, a plethora of information is continuously gathered, with the aim to exploit it to understand phenomena and to make decisions. The notion of Big Data is no longer a novelty: indeed, a multitude of data sets can be collected and provided to end-users through various channels; in particular, one of the most commonly used format to represent Big Data is *JSON* (acronym for JavaScript Object Notation, see [1]), which allows for representing data with complex structures in an easy-to-manage way. The advent of *JSON* has been so disruptive that a specific category of *NoSQL* databases has been developed to manage collections of *JSON* documents; a DBMS (DataBase Management System) belonging to this category is named "*JSON* Document Store" (the most well-known example is *MongoDB*, see [2]).

The variety of data sets that can be collected as *JSON* data sets requires "flexible tools" that are able to query and integrate *JSON* data sets, possibly coming from various sources and storage systems. The concept of "flexibility" could be interpreted in several ways: (i) ability to deal with heterogeneous structures; (ii) ability to manage registry data and geo-tagging; (iii) ability to express imprecise concepts and conditions. The latter ability is generally known as "soft querying": it is characterized by the fact that data items are ranked on the basis of their degree of satisfaction of "soft conditions". Usually, soft querying relies on "Fuzzy-Set Theory", introduced by Zadeh [3] to tackle typical issues of classical Boolean logic.

The *J-CO* Framework is a software tool, developed at University of Bergamo (Italy), that provides all the three above-mentioned capabilities: (i) the *J-CO* Framework is able to acquire *JSON* data sets from various sources and provides a query language, named *J-CO-QL*$^+$, that allows users to specify complex queries and transformations on possibly heterogeneous *JSON* data sets (see [4]); (ii) it natively deals with registry data and geo-tagging in *JSON* data sets (see [5]); (iii) *J-CO-QL*$^+$ provides sophisticated soft-querying capabilities, by supporting the evaluation of membership to multiple fuzzy sets of *JSON* documents. In particular, as far as this last point is concerned, in various works (see [6,7]), it has been shown that *J-CO-QL*$^+$ is an effective tool for performing complex soft queries on large *JSON* data sets, by exploiting classical fuzzy sets.

However, in the literature several extensions to classical fuzzy sets have been proposed; Section 2.1 presents a survey of these extensions, and it can be observed that all of them are characterized by the fact that they give an item in the universe U more than one degree (for example, "membership" degree and "non-membership" degree), while the classical model considers only the "membership degree". In this work, the models considered of interest are *Intuitionistic Fuzzy Sets*, *Neutrosophic Fuzzy Sets* and *Pythagorean Fuzzy Sets*; based on their

* Corresponding author.
E-mail address: giuseppe.psaila@unibg.it (G. Psaila).

https://doi.org/10.1016/j.neucom.2023.126968
Received 3 April 2023; Received in revised form 20 July 2023; Accepted 22 October 2023
Available online 29 October 2023

common characteristics, they can be considered as *Multi-grade Fuzzy Sets*, because items are given multiple yet crisp degrees. Differently, the family of extensions said *Type-2 Fuzzy Sets* introduces the idea of *fuzzy membership degree*.

In a previous work [8], it has been explored the capability of *J-CO-QL*$^+$ of dealing with Intuitionistic Fuzzy Sets without a specific support. That study provided the idea that it was possible to conceive an extension of the language to "manage multi-grade fuzzy sets", provided that a solid meta-model was defined.

The contribution of this paper is to present a unified meta-model for multi-grade fuzzy-set models; the goal of this meta-model is to provide a common formal framework that gives a unified view of existing (and future) models based on multiple crisp degrees. Based on the presented meta-model, novel constructs have been introduced in *J-CO-QL*$^+$ to support the definition of multi-grade models and operators to operate on them, in order to provide analysts with a real and effective tool to experiment with the exploitation of various fuzzy-set approaches to query *JSON* data sets. With the presented extensions, the *J-CO* Framework (and *J-CO-QL*$^+$) has actually provided a unique pool of features that it is not possible to find in any of the few tools that deal with fuzziness in *JSON* data sets.

The paper is organized as follows. Section 2 provides the background from which this work has originated; in particular, Section 2.1 provides the review about different models for fuzzy sets, while Section 2.2 briefly summarizes the literature about soft querying on *JSON* data sets. Section 3 presents the formal contribution of the paper: Section 3.1 formally defines the meta-model for multi-grade fuzzy-set models; Section 3.2 shows how to formalize popular multi-grade models. Section 4 briefly presents the *J-CO* Framework and its characteristics that are relevant for this paper, and Section 5 shows the novel constructs that have been introduced in *J-CO-QL*$^+$ to effectively define and deal with multi-grade fuzzy-set models to process *JSON* data sets. Finally, Section 6 draws the conclusions.

2. Background

This section presents the background of the work proposed in this paper. First of all, a review of models for fuzzy sets that inspired this work is reported in Section 2.1; then, previous research works on soft querying *JSON* data sets are briefly introduced in Section 2.2.

2.1. Review of models for fuzzy sets

The basic contribution of this section is to review the different models for fuzzy sets that have been proposed in the literature. Although this review is not exhaustive, it actually inspired the scientific contribution of this paper.

2.1.1. Classical fuzzy sets

Zadeh introduced the "Fuzzy-Set Theory" in 1965 (see [3,9]) to overcome the limitations of classical Boolean logic applied to natural-language predicates. If the set of *Historical Cars* is defined as the *set of cars whose age is over* 30 *years old*, should a 29 year-old car be considered as new? The basic idea of Zadeh is to associate a membership value (in the continuous range of $[0,1]$) to model the belonging degree of an entity to a specific set, hereinafter denoted as *fuzzy set*. So, while it can be possible to associate a membership value of 1.0 to a 31 year old car to indicate that it fully belongs to the fuzzy set of *Historical Cars*, and a membership value of 0.0 to a 5 year old car to indicate that it does not belong to the fuzzy set of *Historical Cars*, it is also possible to associate a membership value of 0.8 to a 28 year old car to indicate that it is quite old but not as much as to be considered fully historical.

A fuzzy set can be formally defined as follows.

Definition 1. Consider a universe U. The set

$$A = \{ \langle x,\ \mu_A(x) \rangle \mid x \in U \}$$

is said *Fuzzy Set*. The function $\mu_A(x) : U \rightarrow [0,1]$ is the "membership degree" of the item x to the fuzzy set A, in that it represents the degree (extent) with which the item x belongs to A.

The same entity can belong to different fuzzy sets. As an example, in addition to the fuzzy set of *Historical Cars*, it is also possible to define the fuzzy set of *Fast Cars*, which collects *cars that are particularly fast*.

Notice that "Historical Cars" and "Fast Cars" linguistically characterize an item in the two fuzzy sets. So, if someone is looking for "*cars that are historical* ***and*** *fast*", they can play the role of "linguistic predicates", by which it is possible to formulate a "linguistic condition", also said "soft condition". The soft condition that characterizes historical and fast cars can be expressed as:

"Historical and Fast Cars" = "Historical Cars" AND "Fast Cars"

by which it is possible to characterize the fuzzy set of *Historical and Fast Cars*.

However, since it is a derived fuzzy set, a "fuzzy-logic algebra" is necessary, so as to extend the common logical operators *NOT*, *AND* and *OR*. A widely accepted definition is the following.

Definition 2. Given the universe U, two fuzzy sets A and B and an item $x \in U$, the following relations hold:

- $C = \texttt{NOT}\ A \implies \mu_C(x) = 1 - \mu_A(x)$
- $C = A\ \texttt{AND}\ B \implies \mu_C(x) = min\ (\mu_A(x),\ \mu_B(x))$
- $C = A\ \texttt{OR}\ B \implies \mu_C(x) = max\ (\mu_A(x),\ \mu_B(x))$

where μ_A, μ_B and μ_C denote the membership degrees to the fuzzy sets A, B and C, respectively.

According to Definition 2, an interesting role is given to the operator NOT, which can be associated to the complementary concept of *non-membership degree* (denoted as ν): given a fuzzy set A, an item x and its membership degree $\mu_A(x)$ to A, $\nu_A(x) = 1 - \mu_A(x)$ represents the extent with which the item x does not belong to A; in the model of classical fuzzy sets, it is automatically derived from the membership degree, based on the principle of "excluded middle" (see [10]), i.e., the degree of "non membership" to A is necessarily $1 - \mu_A(x)$.

2.1.2. Intuitionistic Fuzzy Sets

The considerations made about the degree of non-membership led many researchers to conceive several extensions of the classical model in which the principle of excluded middle does not hold.

In 1983, Atanasov extended the model of classical fuzzy sets by introducing the concept of *Intuitionistic Fuzzy Set*, starting from the consideration that there are situations where uncertainty does not allow to automatically assign the complement of the membership degree to the non-membership degree (see [10–12]). Intuitionistic Fuzzy Sets are formally defined as follows.

Definition 3. Consider a universe U. The set

$$A = \{ \langle x,\ \mu_A(x),\ \nu_A(x) \rangle \mid x \in U\ \wedge\ \Theta_A(x) \}$$

is said *Intuitionistic Fuzzy Set*. The functions $\mu_A(x) : U \rightarrow [0,1]$ and $\nu_A(x) : U \rightarrow [0,1]$, represent, respectively, the "membership degree" and the "non-membership degree" of the item x to A. For them, the constraint $\Theta_A(x) \equiv (0 \leq \mu_A(x) + \nu_A(x) \leq 1)$ must hold.

The degree $\pi_A(x) = 1 - \mu_A(x) - \nu_A(x)$ is the *indeterminacy degree*: it denotes the degree that it is not possible to characterize neither as membership nor as non-membership; the degree $\pi_A(x)$ is also said "hesitation".

In the classical model, $\nu_A(x)$ was derived from $\mu_A(x)$; in the model of Intuitionistic fuzzy sets, $\nu_A(x)$ is not derived, while $\pi_A(x)$ is a derived degree.

The definition of the new model of Intuitionistic Fuzzy Sets requires a new definition of the operators *NOT*, *AND* and *OR*, in which the two different degrees $\mu(x)$ and $\nu(x)$ are explicitly considered.

Definition 4. Given the universe U, two Intuitionistic Fuzzy Sets A and B and an item $x \in U$, the following relations hold:

$$\bullet\ C = NOT\ A \implies \begin{cases} \mu_C(x) = \nu_A(x) \\ \nu_C(x) = \mu_A(x) \end{cases}$$

$$\bullet\ C = A\ AND\ B \implies \begin{cases} \mu_C(x) = min(\mu_A(x),\ \mu_B(x)) \\ \nu_C(x) = max(\nu_A(x),\ \nu_B(x)) \end{cases}$$

$$\bullet\ C = A\ OR\ B \implies \begin{cases} \mu_C(x) = max(\mu_A(x),\ \mu_B(x)) \\ \nu_C(x) = min(\nu_A(x),\ \nu_B(x)) \end{cases}$$

where μ_A, μ_B and μ_C denote the membership degrees to the fuzzy sets A, B and C, respectively; similarly, ν_A, ν_B and ν_C denote the non-membership degrees to the fuzzy sets A, B and C, respectively.

2.1.3. Neutrosophic Fuzzy Sets

An extension of Intuitionistic Fuzzy Sets was proposed in 1996 by Smarandache (see [13,14]). This is the model of *Neutrosophic Fuzzy Sets*. It is formally defined hereafter.

Definition 5. Consider a universe U. The set

$$A = \{\ \langle\ x,\ \mu_A(x),\ \nu_A(x),\ \pi_A(x)\ \rangle \mid x \in U\ \}$$

is named *Neutrosophic* Fuzzy Set. The functions $\mu_A(x) : U \to [0,1]$, $\nu_A(x) : U \to [0,1]$, and $\pi_A(x) : U \to [0,1]$ represent, respectively, the "membership degree", the "non-membership degree" and the "indeterminacy degree" of the item x to A.

The reader can notice that, now, $\pi_A(x)$ is no longer a derived degree. Furthermore, the constraint $\Theta(x) \equiv (0 \le \mu(x) + \nu(x) \le 1)$ has been removed; as a consequence, only the trivial constraint $0 \le \mu(x) + \nu(x) + \pi(x) \le 3$ holds (i.e., $\mu(x)$, $\nu(x)$ and $\pi(x)$ are completely independent from each other).

For the sake of brevity, the definitions of the operators *NOT*, *AND* and *OR* (as in Definitions 2 and 4) are not reported, but it should be clear that, for each new *multi-grade model of Fuzzy Sets*, a definition of the logical operators should be provided; the reader can refer to Section 3.2, where a precise formalization of the logical operators for Neutrosophic fuzzy sets is reported in Model 4. In this regard, it is interesting to notice that Smarandache [15] provides the definition of two different operators OR (i.e., *INCLUSIVE OR* and *EXCLUSIVE OR*), together with other logical connectors such as *IMPLICATION* and *EQUIVALENCE*.

2.1.4. Pythagorean Fuzzy Sets

A different extension of the model of Intuitionistic Fuzzy Set, named *Pythagorean Fuzzy Set*, was proposed by Yager in 2013 (see [16,17]) by replacing the constraint $0 \le \mu_A(x) + \nu_A(x) \le 1$ with a less restrictive constraint $0 \le \mu_A(x)^2 + \nu_A(x)^2 \le 1$, which relies on the Pythagorean distance.

Definition 6. Consider a universe U. The set

$$A = \{\ \langle\ x,\ \mu_A(x),\ \nu_A(x)\ \rangle \mid x \in U\ \wedge\ \Theta_A(x)\ \}$$

is named *Pythagorean Fuzzy Set*. The functions $\mu_A(x) : U \to [0,1]$ and $\nu_A(x) : U \to [0,1]$ represent, respectively, the "membership degree" and the "non-membership degree" of the item x to A. The constraint $\Theta_A(x) \equiv (0 \le \mu_A(x)^2 + \nu_A(x)^2 \le 1)$ must hold.

It is interesting to notice that in a Pythagorean Fuzzy Set A, it is possible that $\mu_A(x) + \nu_A(x) > 1$, even though the constraint $\mu_A(x)^2 + \nu_A(x)^2 \le 1$ is met.

In the literature, there is not a specific definition for the concept of hesitation (or indeterminacy).

The reader that is interested in definitions of logical operators, can look at Model 5 in Section 3.2.

2.1.5. Refined Fuzzy Sets

An interesting evolution of the models so far presented is the concept of *Refined Fuzzy Set*, which can be applied both to classical fuzzy sets and to any extension considered so far. The concept was introduced by Smarandache [18]. For the sake of clarity, only the simplest model of *Refined Fuzzy Sets* is presented hereafter.

Definition 7. Consider a universe U. The set

$$A = \{\ \langle\ x,\ \mu_1(x),\ \ldots,\ \mu_n(x)\ \rangle \mid x \in U\ \wedge\ n \ge 1\ \}$$

is said *Refined Fuzzy Set*. The function $\mu_i(x)$ (with $1 \le i \le n$) is named *sub-membership of type i*.

The phrase "sub-membership of type i" is taken from [18]: it should not be confused with the notion of "Type-2" fuzzy sets, i.e., fuzzy sets in which membership degrees are not crisp values (see Section 2.1.6). In this context, a "membership type" denotes a specific feature of an entity/item x that is evaluated in a fuzzy manner. The idea behind Refined Fuzzy Sets is to consider several features of entities at the same time by viewing them as a kind of "multi-dimensional fuzzy set", in place of using several distinct fuzzy sets, one for each single considered property.

The same idea is applied in [18] to Intuitionistic, Pythagorean and Neutrosophic fuzzy sets (by having multiple non-membership degrees and multiple indeterminacy degrees). Their usefulness is to model "multi-criteria decision-making". In particular, Rahman et al. [19] discusses *Refined Intuitionistic Fuzzy Sets*, while Saeed et al. [20] analyzes *Refined Pythagorean Fuzzy Sets*.

The conclusion is that the most adopted approach for extending classical fuzzy sets while keeping crisp values for degrees is to deal with many degrees at the same time. For this reason, all the extensions so far presented can be said *multi-grade fuzzy sets*.

2.1.6. Interval-Valued Fuzzy Sets

All the aforementioned models for fuzzy sets share one fundamental property, i.e., degrees are crisp values. However, it may be possible to envision that there might exist situations in which a crisp value for (membership) degrees is not realistic (thus, a further level of uncertainty is considered); such models of fuzzy sets are said *Type-2 fuzzy sets* (see [21,22]).

In particular, Turksen [23] proposed a model of Type-2 fuzzy sets in which the membership degree is assigned to an interval, whose lower and upper bounds are defined by two functions $\mu^l(x)$ and $\mu^u(x)$, which respectively denote the lower and the upper bound of the membership degree of an item $x \in U$ to a fuzzy set. Hereafter, the formal definition of this model is shown.

Definition 8. Consider a universe U. The set

$$A = \{\ \langle x,\ \mu^l_A(x),\ \mu^u_A(x)\ \rangle \mid x \in U\ \wedge\ \Theta_A(x)\ \}$$

is said *Interval-Valued Fuzzy Set*. The functions $\mu^l_A(x) : U \to [0,1]$ and $\mu^u_A(x) : U \to [0,1]$ represent, respectively, the "lower-bound membership degree" and the "upper-bound membership degree" of the item x to A. The constraint $\Theta_A(x) \equiv (0 \le \mu^l_A(x) \le \mu^u_A(x) \le 1)$ must hold.

As a consequence of Definition 8, also the non-membership degree to an Interval-Valued Fuzzy Set should be derived as an interval.

2.1.7. Hesitant Fuzzy Sets

A more complex model is the model of *Hesitant Fuzzy Sets*, proposed by Torra [24]. In this model, the uncertainty on the membership degree is given by the different interpretations of a natural-language predicate provided by several experts, who, consequently, might have different opinions/perceptions. In this case, the membership degree is associated to a set of values that can be different each others.

Definition 9. Consider a universe U. The set

$$A = \{ \langle x,\ h_A(x) \rangle \mid x \in U \}$$

is named *Hesitant fuzzy set*. The element $h_A(x)$ is a set $h_A(x) = \{h_{A_1}(x), h_{A_2}(x), \ldots\}$ where each function $h_{A_i}(x) : U \rightarrow [0,1]$ (with $0 \leq i \leq |h_A(x)|$) represents the opinion of an expert about the membership of the item x to A.

Notice that $h_A(x)$ is a set and not a vector. In other words, given two distinct Hesitant Fuzzy Sets A and B, it might be that $|h_A(x)| \neq |h_B(x)|$, depending on the number of involved observers. Indeed, they cannot be assimilated to Refined fuzzy sets: an Hesitant fuzzy set relies on multiple evaluations of the same feature made by many observers for the same entity; a Refined fuzzy set groups together the evaluations for many different features made by one single observer for a single entity.

Many other fuzzy-set models have been proposed since then. As cited in [25], some of them are *Spherical Fuzzy Sets*, *Picture Fuzzy Sets*, *Cubic Fuzzy Sets*. Describing all of them is out of the scope of this paper; the interested reader can refer to the specific literature.

2.2. Soft querying on JSON document stores

The idea behind "soft querying" is simple: being able to express vague and imprecise conditions to select data items from databases based on a "partial matching" approach: the satisfaction degree is expressed by the membership degree to a fuzzy set (as argued by Blair [26]).

However, when it was conceived, the only available data model for databases was the "relational model"; consequently, research efforts proposed extensions to SQL. As a very short summary, some proposals kept the underlying relational model untouched (i.e., crisp); consequently, soft concepts were introduced only at query level, by extending the statement SELECT; in this category, it is possible to mention *SQLf* (see [27–29]), *FQUERY for Access* (see [30,31]) and *Soft-SQL* (see [32–34]). Other proposals modified the data model towards a "fuzzy-relational model" (see [35,36]) for managing fuzzy values directly within the tables; *FSQL* (see [37,38]) is as a representative of this category. The interested reader can find them all in [39], in [40], in [41], and in [42].

Novel *JSON* document stores (NoSQL databases specifically designed to store *JSON* documents) could provide again fertile ground for research on soft querying databases. Currently, in the literature there are few works that propose fuzzy extensions to query languages for *JSON* document stores. Hereafter, there is a brief presentation of them.

Abir and Amel [43] proposed *fMQL*, an extension of *MQL* (the *MongoDB* query language). The extension is based on *JSON* documents previously tagged with "fuzzy labels": a fuzzy label is equivalent to a linguistic predicate, because it has an associated membership degree; such labels can be referred in selection conditions. Unfortunately, only one single fuzzy label can be associated to each single *JSON* document; furthermore, the paper does not address how to define them.

The work [44] proposes a technique to translate *fMQL* queries into *fXML*, a fuzzy language for querying XML documents. The positive aspect of this paper is that *fMQL* is presented in a more extensive way than in [43]; the negative aspect is that queries are not executed on *JSON* documents but on their XML representations; furthermore, the paper does not address how to deal with membership degrees.

Finally, Medina et al. [45] proposed an extension of the *MongoDB* data model to support fuzzy values at the level of a single document field. Definitely, they propose a fuzzy *JSON* document store, which they have been able to implement on top of *MongoDB*, by exploiting the internal support to execute JavaScript functions. To the best of the authors' knowledge, it can be regarded as the first attempt towards fuzzy *JSON* document stores.

In comparison with the above-mentioned proposals, the *J-CO* Framework and the *J-CO-QL*$^+$ query language adopt a different and unique approach. First of all, the *J-CO* Framework is able to acquire data from both *JSON* stores and web sources, and then it is able to store results into *JSON* stores. Second, it is independent of any specific *JSON* store (see [4]); thus, it relies on the pure *JSON* data model. Third of all, *J-CO-QL*$^+$ gives complete control about fuzzy constructs, as well as it is able to deal with multiple fuzzy sets at the same time for each single *JSON* document.

3. A unified meta-model for multi-grade fuzzy-sets

Based on the review of fuzzy-set models, it can be argued that devising a unified meta-model by which it could be possible to specify any kind of multi-grade fuzzy-set model (such as classical fuzzy sets, Intuitionistic Fuzzy Sets, Neutrosophic Fuzzy Sets and Pythagorean Fuzzy Sets) could be of interest from the scientific point of view.

In the remainder of this section, the definition of the meta-model is provided and this is the formal contribution of the paper.

3.1. The meta-model

In the following, the pool of definitions that constitute the proposed meta-model is presented.

Definition 10 (*Basic Degrees*)**.** The pool of *basic degrees* is a tuple

$$\Delta = \langle\ \delta_1,\ \ldots,\ \delta_n\ \rangle$$

with $n \geq 1$ (i.e., the pool of basic degrees is not empty).

The concept of *basic extent* derives from the concept of *basic degrees*, as defined hereafter.

Definition 11 (*Basic Extent*)**.** Consider an item $x \in U$ and a pool of basic degrees Δ. The *basic extent* of x with respect to Δ is a tuple

$$\xi^{\Delta}(x) = \langle\ d_1,\ \ldots,\ d_n\ \rangle$$

where $n = |\Delta|$ and $d_i \in [0,1]$, for each $1 \leq i \leq n$. $\xi^{\Delta}(x)$ can be written as ξ^{Δ}, for simplicity.

In other words, ξ^{Δ} is the tuple of values that instantiates the basic degrees in Δ.

Some fuzzy-set models encompass secondary or *derived* degrees, i.e., degrees that are computed on the basis of the values for basic degrees.

Definition 12 (*Derived Degrees*)**.** Given a pool Δ of basic degrees, the pool of *derived degrees* is a tuple

$$\Gamma = \langle\ \gamma_1,\ \ldots,\ \gamma_k\ \rangle$$

where $k \geq 0$, (i.e., the pool of derived degrees can be empty). Specifically, γ_i (with $1 \leq i \leq k$) is a pair

$$\gamma_i \equiv \eta_i : \phi_i$$

where η_i is the name of the derived degree (such that $\eta_i \notin \Delta$), while $\phi_i(\xi^{\Delta})$ is the function that derives its value (from a basic extent ξ^{Δ}).

Similarly to the notion of *basic extent*, it is possible to conceive the notion of *derived extent*.

Definition 13 (*Derived Extent*)**.** Consider an item $x \in U$, its basic extent ξ^{Δ} and a pool Γ of derived degrees. The *derived extent* of x with respect to Γ is a tuple

$$\xi^{\Gamma}(x) = \langle g_1, \ \ldots, \ g_k \rangle$$

where $k = |\Gamma|$ and $g_i \in [0, 1]$, for each $1 \leq i \leq k$ and $g_i = \phi_i(\xi^{\Delta})$. For simplicity, $\xi^{\Gamma}(x)$ can be written as ξ^{Γ}.

In other words, ξ^{Γ} is the tuple of values that instantiates the derived degrees in Γ.

Definition 14 (*Full Extent*)**.** Given an item $x \in U$, its basic extent $\xi^{\Delta}(x)$ and its derived extent $\xi^{\Gamma}(x)$, the *(full) extent* of x is $\xi(x) = \xi^{\Delta}(x) \bullet \xi^{\Gamma}(x)$, where $\bullet$ is the concatenation operator. For simplicity, $\xi(x)$ can be written as ξ.

Multi-grade fuzzy-set models often consider some kind of "validity constraint" on degrees.

Definition 15 (*Validity Constraint*)**.** Given a pool of basic degrees Δ and a pool of derived degrees Γ (derived from Δ), the *validity constraint* denoted as $\Theta : \Delta \bullet \Gamma \rightarrow \{true, false\}$ is a Boolean logical function on degrees either in Δ or in Γ.

Finally, a fuzzy-set model provides definitions for logical operators.

Definition 16. For a fuzzy-set model M and two fuzzy sets A and B defined on M, the pool of *logical operators* is a tuple

$$\Omega = \langle \neg, \wedge, \vee \rangle$$

where $\neg$ is a unary function $\phi(\xi_A) = \langle d_1 : \phi_1(\xi_A), \ldots, d_n : \phi_n(\xi_A) \rangle$, for each $d_i \in \Delta$; similarly, $\wedge$ and $\vee$ are defined by a binary function $\phi(\xi_A, \xi_B) = \langle d_1 : \phi_1(\xi_A, \xi_B), \ldots, d_n : \phi_n(\xi_A, \xi_B) \rangle$, for each $d_i \in \Delta$.

In other words, provided two full extents ξ_A and ξ_B of an item x to two fuzzy sets A and B defined on the same model, evaluating $A \wedge B$ (or $A \vee B$, respectively) on x gives rise to a new extent $\xi_{A \wedge B}(x)$ (or $\xi_{A \vee B}(x)$, respectively). Similarly for $\neg A$.

Consequently, a logical operator is defined as a "tuple of functions", one for each basic degree in Δ; derived degrees in Γ are automatically derived (by means of functions defined in Γ).

At this point, it is possible to define the meta-model for fuzzy-set models.

Definition 17 (*Fuzzy-Set Model*)**.** A Fuzzy-Set Model is a tuple

$$\langle Name, \ \Delta, \ \Gamma, \ \Theta, \ \Omega \rangle$$

where "$Name$" is the name of the fuzzy-set model.

The member "Δ" is the non-empty list (or tuple) of "basic degrees" (see Definition 10).

The member "Γ" is the (possibly empty) list (or tuple) of "derived degrees", i.e., degrees that are automatically derived from degrees in Δ (see Definition 12).

The member "Θ" is a *validity constraint*: it is a crisp Boolean condition on degrees to have a valid extent (see Definition 15).

The member "Ω" is a tuple $\langle \neg, \wedge, \vee \rangle$, whose members are functions that define the logical operators "$\neg$" (NOT), "$\wedge$" (AND), and "$\vee$" (OR) for the fuzzy-set model (see Definition 16).

In the following, given a Fuzzy-Set Model M, the dot-notation (e.g., $M.\Delta$) will be used to refer to a member of M (e.g., the member Δ of "basic degrees").

The reader can notice that, in Definitions 11, 13 and 14, the term *membership* is not mentioned. Indeed, from the review made in Section 2.1, it is clear that *membership* is just one degree; many models relies on the degree of *non-membership*, and so on. Thus, generalizing, given a multi-grade fuzzy-set model and a fuzzy set A defined on this model, the set ξ_A of degrees determines "the extent of the item x to the fuzzy set A" (denoted as $\xi_A(x)$).

Clearly, each single degree in ξ_A denotes a specific dimension that characterize the extent of x to A. Thus, the extent is a *multi-dimensional* concept, by which it is possible to consider several fuzzy features; consequently, it is possible to affirm that ξ_A is the *multi-dimension extent* of x to A (while the single membership degree in classical fuzzy sets is a sample of one-dimensional extent).

Definition 18 (*Valid Extent*)**.** Given an item x and its full extent $\xi_A(x)$ to a fuzzy set A, if $\Theta(\xi_A) = false$, this means that its extent to the fuzzy set A is not valid; as a result, an item x with an invalid extent is equivalent to an item x with an undefined (or unknown) extent to the fuzzy set A.

Finally, it is possible to conceive "operators" that evaluate the extent of an item x to a fuzzy-set defined on a model M, by possibly moving from properties of x and from its extents to other fuzzy sets (possibly of different type).

Definition 19 (*Multi-grade Operator*)**.** Given a fuzzy-set model M, a *multi-grade operator* evaluates the extent of an item x to a fuzzy set defined on M. It is defined as:

$$\begin{aligned} \Phi(p_1, \ldots, p_h) = \langle & d_1 : \phi_1(p_1, \ldots, p_h), \\ & \ldots, \\ & d_n : \phi_n(p_1, \ldots, p_h) \rangle, \end{aligned}$$

with $d_i \in M.\Delta$. The actual value v_j for a parameter p_j (with $1 \leq j \leq h$) could be any kind of value, including an extent to a fuzzy-set model M_j (possibly different from M).

The reader can notice that a multi-grade operator can be used to generate extents from properties of items only, as well as it can be used to convert extents of different fuzzy-set models into an extent to the target model M.

The remainder of this section shows how classical, Intuitionistic, Neutrosophic and Pythagorean fuzzy-set models can be defined through the meta-model introduced in Definition 17.

3.2. Formalizing fuzzy-set models

The meta-model introduced in Section 3.1 can be exploited to formally define, in a unified way, the classical fuzzy-set model and the multi-grade models for Intuitionistic, Neutrosophic and Pythagorean fuzzy sets.

3.2.1. Classical fuzzy sets

Based on Definition 17, the classical model for fuzzy sets can be defined as follows.

Model 1. *On the basis of Definitions 1 and 2, the classical model for fuzzy sets is defined by the following tuple.*

$$\begin{aligned} & \langle \ Name : \texttt{"classical"}, \\ & \quad \Delta : \langle \ \mu \ \rangle, \\ & \quad \Gamma : \langle \ \rangle, \\ & \quad \Theta : true, \\ & \quad \Omega : \langle \ \neg(\xi_A) \quad\quad : \langle \ \mu : 1 - \xi_A.\mu \ \rangle, \\ & \qquad\quad \wedge(\xi_A, \xi_B) : \langle \ \mu : min(\xi_A.\mu, \xi_B.\mu) \ \rangle, \\ & \qquad\quad \vee(\xi_A, \xi_B) : \langle \ \mu : max(\xi_A.\mu, \xi_B.\mu) \ \rangle \\ & \qquad \rangle \\ & \rangle \end{aligned}$$

In the classical model, only the concept of *membership* is defined; consequently, the member Δ contains only the field μ, while the member Γ is empty. Consequently, the extent ξ_A of an item x to a classical fuzzy set A is a tuple containing one single value; an example could be $\xi_{\texttt{HistoricalCars}} = \langle \mu : 0.9 \rangle$.

The member Θ is set to *true*, since there is no a specific constraint that the extent must satisfy.

Finally, the extended logical operators are defined: notice that they return a new extent (that, for classical fuzzy sets, contains only the field μ). Binary operators receive two extents (denoted as ξ_A and ξ_B), while the unary operator receives only one extent (denoted as ξ_A).

A little variation of this model originates from the consideration that it is possible to derive the non-membership degree as $\nu = 1 - \mu$.

Model 2. *On the basis of Definitions 1 and 2, the classical model for fuzzy sets, including the derived non-membership degree, is defined by the following tuple.*

$$
\begin{aligned}
&\langle\ Name : \texttt{"classicalEnriched"},\\
&\quad \Delta : \langle\ \mu\ \rangle,\\
&\quad \Gamma : \langle\ \nu : 1-\mu\ \rangle,\\
&\quad \Theta : true,\\
&\quad \Omega : \langle\ \neg(\xi_A) \quad : \langle\ \mu : 1-\xi_A.\mu\ \rangle,\\
&\qquad\quad \wedge(\xi_A,\xi_B) : \langle\ \mu : min(\xi_A.\mu, \xi_B.\mu)\ \rangle,\\
&\qquad\quad \vee(\xi_A,\xi_B) : \langle\ \mu : max(\xi_A.\mu, \xi_B.\mu)\ \rangle\\
&\qquad \rangle\\
&\rangle
\end{aligned}
$$

In this case, Model 2 is the same as Model 1, except that a new $Name$ is defined and the member Γ now contains the definition of the derived degree ν of non-membership and its defining function. Consequently, the extent ξ of an item x to a classical fuzzy set A is a tuple containing two values, such as $\xi_{\texttt{HistoricalCars}} = \langle \mu : 0.9, \nu : 0.1 \rangle$.

3.2.2. Intuitionistic Fuzzy Sets

According to Definition 17, the model for Intuitionistic fuzzy sets can be defined as follows.

Model 3. *On the basis of Definitions 3 and 4, the model for Intuitionistic fuzzy sets is defined by the following tuple.*

$$
\begin{aligned}
&\langle\ Name : \texttt{"intuitionistic"},\\
&\ \Delta : \langle\ \mu,\ \nu\ \rangle,\\
&\ \Gamma : \langle\ \pi : 1-\mu-\nu\ \rangle,\\
&\ \Theta : 0 \le \mu+\nu \le 1,\\
&\ \Omega : \langle\ \neg(\xi_A) \quad : \langle\ \mu : \xi_A.\nu,\\
&\qquad\qquad\qquad\quad \nu : \xi_A.\mu\ \rangle,\\
&\qquad \wedge(\xi_A,\xi_B) : \langle\ \mu : min(\xi_A.\mu, \xi_B.\mu),\\
&\qquad\qquad\qquad\quad \nu : max(\xi_A.\nu, \xi_B.\nu)\ \rangle,\\
&\qquad \vee(\xi_A,\xi_B) : \langle\ \mu : max(\xi_A.\mu, \xi_B.\mu),\\
&\qquad\qquad\qquad\quad \nu : min(\xi_A.\nu, \xi_B.\nu)\ \rangle\\
&\quad \rangle\\
&\rangle
\end{aligned}
$$

In the Intuitionistic Fuzzy-Set model, both the concepts of membership and non-membership are defined; thus, the member Δ contains both the degrees μ and ν, while the member Γ contains the degree π with its defining function. Consequently, the extent ξ of an item x to an Intuitionistic fuzzy set A is a tuple containing three values; an example could be $\xi_{\texttt{HistoricalCars}} = \langle \mu : 0.7, \nu : 0.2, \pi : 0.1 \rangle$.

The member Θ contains the constraint $0 \le \mu + \nu \le 1$, and, finally, the member Ω contains the definition of the logical operators.

3.2.3. Neutrosophic Fuzzy Sets

According to Definition 17, the model for Neutrosophic fuzzy sets can be defined as follows, considering that Smarandache [15] provided two different definitions for the logical operator *OR* (as reported in Section 2.1.3). In this case, it has been decided to apply both the definitions of the operator *INCLUSIVE OR* to the *OR* and to define a multi-grade operator, according to Definition 19, to perform the *EXCLUSIVE OR*.

Model 4. *On the basis of Definition 5, the model for Neutrosophic fuzzy sets is defined by the following tuple.*

$$
\begin{aligned}
&\langle\ Name : \texttt{"neutrosophic"},\\
&\quad \Delta : \langle\ \mu,\ \nu,\ \pi\ \rangle,\\
&\quad \Gamma : \langle\ \rangle,\\
&\quad \Theta : true,\\
&\quad \Omega : \langle\ \neg(\xi_A) \quad : \langle\ \mu : 1-\xi_A.\mu,\\
&\qquad\qquad\qquad\quad \nu : 1-\xi_A.\nu,\\
&\qquad\qquad\qquad\quad \pi : 1-\xi_A.\pi\ \rangle,\\
&\qquad\quad \wedge(\xi_A,\xi_B) : \langle\ \mu : \xi_A.\mu \times \xi_B.\mu,\\
&\qquad\qquad\qquad\quad \nu : \xi_A.\nu \times \xi_B.\nu,\\
&\qquad\qquad\qquad\quad \pi : \xi_A.\pi \times \xi_B.\pi\ \rangle,\\
&\qquad\quad \vee(\xi_A,\xi_B) : \langle\ \mu : \xi_A.\mu + \xi_B.\mu - \xi_A.\mu \times \xi_B.\mu,\\
&\qquad\qquad\qquad\quad \nu : \xi_A.\nu + \xi_B.\nu - \xi_A.\nu \times \xi_B.\nu,\\
&\qquad\qquad\qquad\quad \pi : \xi_A.\pi + \xi_B.\pi - \xi_A.\pi \times \xi_B.\pi\ \rangle\\
&\qquad \rangle\\
&\rangle
\end{aligned}
$$

The model is completed with the multi-grade operator Φ_{EX_OR} to describe the EXCLUSIVE OR.

$$
\begin{aligned}
&\Phi_{EX\_OR}(\ \xi_A,\ \xi_B\ ) = \langle\\
&\qquad \mu : \xi_A.\mu \times (1-\xi_B.\mu)\ +\ \xi_B.\mu \times (1-\xi_A.\mu)\\
&\qquad\quad -\ \xi_A.\mu \times \xi_B.\mu \times (1-\xi_A.\mu) \times (1-\xi_B.\mu),\\
&\qquad \nu : \xi_A.\nu \times (1-\xi_B.\nu)\ +\ \xi_B.\nu \times (1-\xi_A.\nu_A)\\
&\qquad\quad -\ \xi_A.\nu \times \xi_B.\nu \times (1-\xi_A.\nu) \times (1-\xi_B.\nu),\\
&\qquad \pi : \xi_A.\pi \times (1-\xi_B.\pi)\ +\ \xi_B.\pi \times (1-\xi_A.\pi)\\
&\qquad\quad -\ \xi_A.\pi \times \xi_B.\pi \times (1-\xi_A.\pi) \times (1-\xi_B.\pi)\\
&\rangle
\end{aligned}
$$

In the mode for Neutrosophic fuzzy sets, the concepts of membership, non-membership and indeterminacy are defined; thus, the member Δ contains the fields μ, ν and π, while the member Γ is empty. Consequently, the extent ξ of an item x to a Neutrosophic fuzzy set A is a tuple containing three values; an example could be $\xi_{\texttt{HistoricalCars}} = \langle \mu : 0.7, \nu : 0.8, \pi : 0.4 \rangle$.

The member Θ is set to true, since there is no specific constraint that the extent must satisfy. The member Ω contains the definitions of the basic logical operators, where the operator OR is defined as *INCLUSIVE OR*.

Finally, the model is completed with the multi-grade operator Φ_{EX_OR}, to which the ξ_A, ξ_B extents to two Neutrosophic fuzzy sets must be provided to calculate the *EXCLUSIVE OR*.

3.2.4. Pythagorean Fuzzy Sets

According to Definition 17, the model for Pythagorean Fuzzy Sets can be defined as follows.

Model 5. *On the basis of Definition 6, the model for Pythagorean fuzzy sets is defined by the following tuple.*

$$
\langle\ Name : \texttt{"pythagorean"},
$$

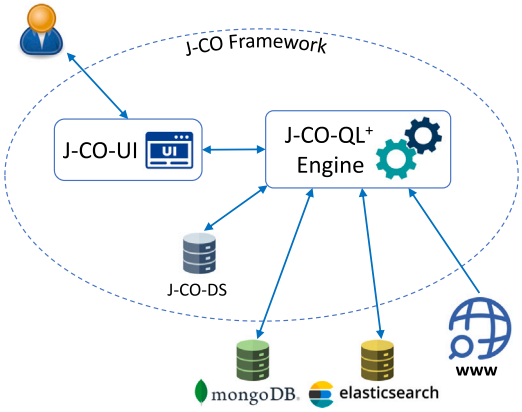


Fig. 1. Components of the *J-CO* Framework.

$$
\begin{aligned}
&\Delta : \langle\ \mu,\ \nu\ \rangle,\\
&\Gamma : \langle\ \rangle,\\
&\Theta : 0 \le \mu^2 + \nu^2 \le 1\\
&\Omega : \langle\ \neg(\xi_A) \quad : \langle\ \mu : \xi_A.\nu,\\
&\qquad\qquad\qquad\quad \nu : \xi_A.\mu\ \rangle,\\
&\qquad \wedge(\xi_A, \xi_B) : \langle\ \mu : min(\xi_A.\mu, \xi_B.\mu),\\
&\qquad\qquad\qquad\quad \nu : max(\xi_A.\nu, \xi_B.\nu)\ \rangle,\\
&\qquad \vee(\xi_A, \xi_B) : \langle\ \mu : max(\xi_A.\mu, \xi_B.\mu),\\
&\qquad\qquad\qquad\quad \nu : min(\xi_A.\nu, \xi_B.\nu)\ \rangle\\
&\quad\ \rangle\\
\rangle
\end{aligned}
$$

In the model for Pythagorean fuzzy sets, the concepts of membership and non-membership are defined; thus the member Δ contains both the degrees μ and ν, while the member Γ is empty (since there is no definition for indeterminacy). Consequently, the extent ξ of an item x to a Pythagorean fuzzy set A is a tuple containing two values; an example could be $\xi_{\texttt{HistoricalCars}} = \langle \mu : 0.7, \nu : 0.7 \rangle$.

The member Θ contains the constraint $0 \le \mu^2 + \nu^2 \le 1$, and finally the member Ω contains the definitions of the logical operators, according to the definitions provided by Yager [17].

4. The J-CO Framework

The *J-CO* Framework is a suite of software tools, developed at the University of Bergamo (Italy); together with its own query language named *J-CO-QL*$^+$, it is aimed at providing analysts with a powerful instrument to retrieve, manage, transform and integrate collections of *JSON* documents.

The project has already been introduced by other papers; consequently, the remainder of this section shows a quick overview of the main characteristics of the *J-CO* Framework to make clear the *J-CO-QL*$^+$ script exposed in Section 5.2 and to highlight the novelties introduced with this paper. First of all, Section 4.1 presents a description of the framework and its components. Then, Section 4.2 introduces the execution model. Finally, Section 4.3 explains how the *J-CO* Framework supports fuzzy sets and soft querying. Uninteresting details for this paper are omitted; the interested reader can refer to Bordogna et al. [5], Psaila and Fosci [4] and Fosci and Psaila [7,46].

4.1. Components of the J-CO Framework

Fig. 1 depicts the *J-CO* Framework in relation to the external services that can provide collections of *JSON* documents and storage capabilities. Hereafter, the framework is presented in details.

```
{
  "model"        : "Testarossa",
  "manufacturer" : "Ferrari",
  "year"         : 1995,
  "~fuzzysets "  : {
    "FastCars"          : 0.99
  }
}
```

Fig. 2. Example of *JSON* document with one membership degree.

- *J-CO-QL*$^+$ *Engine*. This component is the core of the framework: it actually executes scripts written in the *J-CO-QL*$^+$ language. The *J-CO-QL*$^+$ *Engine* is able to interact with several *NoSQL* databases, either internal, such as *J-CO-DS*, or external, such as *MongoDB* and *ElasticSearch*, to retrieve and save *JSON* collections. The engine is also able to retrieve data directly from the Internet.
- *J-CO-UI*. This is the interface that allows users to write *J-CO-QL*$^+$ scripts, execute them and inspect results.
- *J-CO-DS*. This component (see [47]) is a *NoSQL* repository able to store huge single documents. No query language is provided by *J-CO-DS*, because the computational capability is provided by the *J-CO-QL*$^+$ *Engine*.

The *J-CO-QL*$^+$ *Engine* has been designed to be independent of any *JSON* storage systems (see [4]): in fact, this way *J-CO-QL*$^+$ can be designed as a powerful query language, which can provide constructs whose execution is independent of the actual computational capabilities provided (if any) by the data sources. The *J-CO-QL*$^+$ *Engine* must interact with external systems only for getting data sets and possibly for storing new generated collections of *JSON* documents.

4.2. Execution model

The execution model of *J-CO-QL*$^+$ is based upon three concepts: *collections*, *process state* and *pipeline*.

A *collection* is a multi-set (i.e., a set with possibly multiple occurrences of the same item) of *JSON* documents.

The *process state* is a tuple $s = \langle tc, DBS, FO \rangle$, which describes the status of the *J-CO-QL*$^+$ *Engine* over time. In details:

- the member tc is the "temporary collection", which is the collection of *JSON* documents currently available to be processed by the next *J-CO-QL*$^+$ instruction;
- the member DBS is the set of database descriptors to which the *J-CO-QL*$^+$ *Engine* can connect to retrieve and save collections of *JSON* documents;
- the member FO is the set of "Fuzzy Operators". Defined through a specific *J-CO-QL*$^+$ statement, a Fuzzy Operator is able to evaluate the membership degree to a fuzzy set of a *JSON* document, upon a user-defined series of parameters, when used in a soft condition.

A *J-CO-QL*$^+$ query (or script) can be seen as a sequence $q = i_1 \bullet \ldots \bullet i_n$, where, as in a "pipeline", the execution of each instruction i_j modifies one of the members of the process state. The initial process state is $s_0 = \langle \emptyset, \emptyset, \emptyset \rangle$ (i.e., all members are empty); then, a generic process state s_j is given by the execution of the instruction i_j upon the process state s_{j-1}, that is $s_j = i_j(s_{j-1})$.

More details about *J-CO-QL*$^+$ statements will be introduced in Section 5.2.

4.3. Support to fuzzy sets and soft querying

This work describes the extensive improvements to the very basic capabilities for dealing with fuzzy sets already provided by *J-CO-QL*$^+$. In fact, a *JSON* document represents an entity (in a universe) that can be member of several fuzzy sets at the same time, with

```
{
  "model"        : "Testarossa",
  "manufacturer" : "Ferrari",
  "year"         : 1995,
  "~fuzzysets "  : {
    "FastCars"           : 0.99,
    "HistoricalCars"     : 0.80,
    "HistoricalFastCars" : 0.80
  }
}
```

Fig. 3. Example of *JSON* document with multiple membership degrees.

different membership degrees. Consequently, membership degrees to fuzzy sets should be incorporated within the same *JSON* document. To this end, the *J-CO-QL*$^+$ *Engine* exploits a special root-level field, named `~fuzzysets`. It is a flat nested document that behaves like a "key/value" map: the name of each inner field denotes a fuzzy-set name; its value, in the range $[0, 1]$, represents the membership degree of the document to that fuzzy set.

Fig. 2, depicts a *JSON* document whose membership degree to the fuzzy set `FastCars` has been already evaluated (the field `~fuzzysets` is highlighted by a blue box).

In *J-CO-QL*$^+$ the membership of a *JSON* document to a fuzzy set can be evaluated on the basis of the values of its properties (i.e., fields), by means of a user-defined "Fuzzy Operator" (details will be shown in Section 5.2).

Fuzzy Operators and fuzzy sets can be exploited in soft-conditions to evaluate the membership of the *JSON* document to fuzzy sets. To this end, several statements in *J-CO-QL*$^+$ are provided with an optional clause `CHECK FOR`, in which several branches `FUZZY SET` are allowed, each one to evaluate the membership to a fuzzy set by means of a specific soft condition. Hereafter, a simple example is shown:

```
CHECK FOR
  FUZZY SET HistoricalCars
    USING isHistoricalCar(.year)
  FUZZY SET HistoricalFastCars
    USING HistoricalCars AND FastCars
```

The first branch `FUZZY SET` evaluates the membership degree to the fuzzy set `HistoricalCars`, by means of a soft condition that is expressed after the keyword `USING`; the previously declared fuzzy operator `isHistoricalCar` (whose definition is not reported for the sake of brevity) computes the membership degree on the basis of the field `year` in the *JSON* document.

The membership degree to the fuzzy set `HistoricalFastCars` is evaluated in the second branch `FUZZY SET` by means of the soft condition "`HistoricalCars AND FastCars`".

Fig. 3 depicts the same *JSON* document shown in Fig. 2, after the evaluation of the soft conditions in the example. Notice that the field `~fuzzysets` now also contains the membership degrees to the fuzzy sets `HistoricalCars` and `HistoricalFastCars`.

5. Novel constructs for defining and processing multi-grade Fuzzy-set models in *J-CO-QL*$^+$

This section presents the final contribution of the paper, i.e., novel constructs for dealing with multi-grade fuzzy-set models in *J-CO-QL*$^+$. First of all, a plausible case study is described. Then, the novel constructs provided by the *J-CO-QL*$^+$ language are illustrated.

5.1. Case study

Suppose that a market-research company, named *ACME*, has to carry out a survey about the effectiveness of marketing campaigns on a series of products. For each product, the company prepares an internet questionnaire to ask people for their opinion about various topics. Each topic can be associated to a question for the participants. For instance, if the product is an item of clothing, *ACME* could ask: (i) *"Is the item suitable for doing sport?"* (ii) *"Is the item suitable for a fancy dinner?"* (iii) *"Did you like the commercial on TV?"*. For each topic, the possible answers are `YES`, `NO`, and `I DON'T KNOW` (*idk*). Under the hypothesis that people could be completely "non-rational", the participants to the survey are free to select, for each topic, one, or two, or even all the three possible answers. They are also free to choose not to answer to a topic.

At the end of the survey, *ACME* collects all the data and, for each topic, calculates the percentage of answers `YES`, of answers `NO`, and of answers `I DON'T KNOW`, on the basis of the number of participants. So, according to the hypothesis of non-rationality, for each topic there is a triplet of percentages, such as $(67.2\%, 45.6\%, 73.3\%)$ for, respectively, the answers `YES`, `NO`, and `I DON'T KNOW`. Notice that the three percentages are completely independent; thus, their sum can possibly be greater than 100%. This characteristic strongly reminds the Neutrosophic model: indeed, according to Definition 5, the answers to each topic can be easily mapped to the degrees of a Neutrosophic fuzzy set. In particular, the percentage of answers `YES` can be associated to the membership degree μ, the percentage of answers `NO` can be associated to the non-membership degree ν, and the percentage of answers `I DON'T KNOW` can be associated to the indeterminacy degree π.

As a final result of the survey, *ACME* obtains a collection of *JSON* documents, one for each product, reporting the data of the product and the percentages of the answers to each topic. Fig. 4 shows a sample document: the document reports the *identifier* of the product (the field `productId`), the *name* of the product (the field `name`) and other properties like `category`, `brand` and `price`. Moreover, the document also reports a series of structured fields, with a name that is based on the pattern "`eval`$\langle Topic\_Name \rangle$", one for each topic of the survey, that contain the percentage of answers `YES` (the sub-field `yes`), the percentage of answers `NO` (the sub-field `no`) and the percentage of answers `I DON'T KNOW` (the sub-field `idk`). Thus, the answers to the three example topics presented before about the suitability of a product for sport or for a fancy evening, and the appreciation of commercials on TV, can be contained in the three structured fields named, respectively, `evalSporty`, `evalFancy` and `evalCommercial`. In Fig. 4, the three fields are highlighted by colored boxes.

Suppose now that, at the end of the survey, *ACME* is committed by its clients to perform the following query:

Query 1. *Perform a query on the data set so as to discover "*those products that are suitable for doing sport *OR (*i.e., *EXCLUSIVE OR)* for a fancy evening, *AND* whose commercials were particularly appreciated*"*.

This is a "soft query" that is based on the Neutrosophic model.

5.2. *J-CO-QL*$^+$ *script*

This section shows how Query 1 can be executed by a *J-CO-QL*$^+$ script that exploits the novel constructs to deal with multi-grade fuzzy-set models. The script is divided in three parts:

- the first part, as reported in Listing 1 and discussed in Section 5.2.1, shows how to define a Fuzzy-Set Model;
- the second part, as reported in Listing 2 and discussed in Section 5.2.2, shows how to define novel Fuzzy Operators;
- the third part, as reported in Listing 3 and discussed in Section 5.2.3, shows how to apply Fuzzy-Set Models and Fuzzy Operators in order to perform a non-trivial soft query.

5.2.1. Defining a fuzzy-set model

As anticipated in Section 5.1, the Neutrosophic model presented in Definition 5 is the fuzzy-set model that better describes the answers to a topic in the survey about products, under the hypothesis of "non-rationality". In this case, Listing 1 provides a modified version of the Neutrosophic model, enriched with three derived degrees reporting,

```
{
    "productId"       : 10,
    "name"            : "Air Min XL Strip",
    "category"        : "running shoes",
    "brand"           : "xxx",
    "price"           : 144.99,
    …, other product properties, …,
    "evalFancy"       : {
        "yes"  :  9.10,
        "no"   : 85.30,
        "idk"  : 29.20
    },
    "evalSport"       : {
        "yes"  : 97.20,
        "no"   : 09.40,
        "idk"  : 21.10
    },
    "evalCommercial"  : {
        "yes"  : 95.70,
        "no"   : 11.40,
        "idk"  : 16.50
    },
    …, other product topics survey, …
}
```

Fig. 4. Example of starting document for the *J-CO-QL*$^+$ script.

respectively, the complement of the membership, non-membership and indeterminacy degrees of the standard Neutrosophic model. The model defines the logical operators `NOT`, `AND`, and `OR` (in its inclusive version) as defined in [15].

Listing 1 *J-CO-QL*$^+$ Script: Defining a Fuzzy-Set Model.

```
1. CREATE FUZZY SET MODEL neutrosophic
     DEGREES mu, nu, pi
     DERIVED DEGREES
       muc AS 1-mu,
       nuc AS 1-nu,
       pic AS 1-pi
     OPERATOR NOT
       EVALUATE mu AS x.muc
       EVALUATE nu AS x.nuc
       EVALUATE pi AS x.pic
     OPERATOR AND
       EVALUATE mu AS x.mu * y.mu
       EVALUATE nu AS x.nu * y.nu
       EVALUATE pi AS x.pi * y.pi
     OPERATOR OR
       EVALUATE mu AS x.mu + y.mu - x.mu * y.mu
       EVALUATE nu AS x.nu + y.nu - x.nu * y.nu
       EVALUATE pi AS x.pi + y.pi - x.pi * y.pi
  ;
```

Hereafter, Listing 1 is presented in details.

- The instruction `CREATE FUZZY SET MODEL` on Line 1 defines a novel Fuzzy-Set Model named `neutrosophic`. The clause `DEGREES` defines three basic degrees (see Definition 10) named `mu`, `nu`, and `pi` that denote, respectively, the membership degree μ, the non-membership degree ν, and the indeterminacy degree π of the formal Neutrosophic model defined in Definition 5 and formalized in Model 4.
  The subsequent clause `DERIVED DEGREES` defines three derived degrees: the degree `muc` (short name for "mu complement"), calculated as `1 - mu`; the degree `nuc` (short name for "nu complement"), calculated as `1 - nu`; the degree `pic` (short name for "pi complement"), calculated as `1 - pi`. They represent, respectively, the complements of the degrees μ, ν, and π of the formal Neutrosophic model. Remember that basic degrees give rise to the basic extent (Definition 11), from which the derived extent (Definition 13) is automatically derived; basic extent and derived extent constitute the full extent (see Definition 14).
  Then, the clause `OPERATOR NOT` defines the operator *NOT*: in detail, it specifies how to calculate the basic extent by applying the operator to the full extent of an item of a Neutrosophic fuzzy set. In this case, a full extent to a Neutrosophic fuzzy set is received as an implicit parameter, whose name is `x`; the following clauses `EVALUATE` report the expressions to calculate, respectively, the basic degrees `mu`, `nu`, and `pi` (derived degrees are automatically computed from basic degrees, as defined in the clause `DERIVED DEGREES`).
  The following clause `OPERATOR AND` defines the operator *AND* for an item in a couple of Neutrosophic fuzzy sets. In this case, two implicit parameters are received, named `x` and `y`, whose values are the full extents to two fuzzy sets of the model under definition. The clauses `EVALUATE` specifies how to calculate, respectively, the basic degrees `mu`, `nu`, and `pi`.
  Similarly, the clause `OPERATOR OR` defines the operator *(INCLUSIVE) OR* (see Model 4).

Notice the dot-notation to reference a degree (e.g., `x.mu`) in an extent and the exploitation of derived degrees to make expressions easier to write (e.g., `y.muc`).

The statement `CREATE FUZZY SET MODEL` is the first novel construct that has been added to *J-CO-QL*$^+$ to manage multi-grade fuzzy-set models.

5.2.2. Defining fuzzy operators

Listing 2 *J-CO-QL*$^+$ Script: Defining Fuzzy Operators.

```
2. CREATE neutrosophic FUZZY OPERATOR Neutrosophify
     PARAMETERS
       m TYPE FLOAT,
       n TYPE FLOAT,
       p TYPE FLOAT
     PRECONDITION
       m >= 0 AND m<=100 AND
       n >= 0 AND n<=100 AND
       p >= 0 AND p<=100
     EVALUATE mu AS m/100
     EVALUATE nu AS n/100
     EVALUATE pi AS p/100
  ;

3. CREATE neutrosophic FUZZY OPERATOR ExclusiveOR
     PARAMETERS
       x TYPE neutrosophic,
       y TYPE neutrosophic
     EVALUATE mu AS x.mu*y.muc + y.mu*x.muc
                  - x.mu*y.mu*x.muc*y.muc
     EVALUATE nu AS x.nu*y.nuc + y.nu*x.nuc
                  - x.nu*y.nu*x.nuc*y.nuc
     EVALUATE pi AS x.pi*y.pic + y.pi*x.pic
                  - x.pi*y.pi*x.pic*y.pic
  ;

4. CREATE FUZZY OPERATOR FuzzifyNeutrosophic
     PARAMETERS
       x TYPE neutrosophic
     EVALUATE x.mu * x.mu * x.nuc * y.pic
       POLYLINE
         [ (0.0, 0.0), (0.3, 0.1),
           (0.7, 0.9), (1.0, 1.0) ]
  ;
```

In *J-CO-QL*$^+$, a fuzzy operator is a key tool to write soft conditions, because it allows for evaluating extents (memberships, for the classical model) to fuzzy sets, based on entity properties (i.e., document fields) and to work on extents to possibly different fuzzy-set models.

Listing 2 defines three fuzzy operators: the first one builds the extent to a Neutrosophic fuzzy set from three numerical parameters; the second fuzzy operator evaluates the extent for the operator *EXCLUSIVE OR* between the extents to two Neutrosophic fuzzy sets (according to the definition provided by Smarandache [15]); finally, the fuzzy operator on Line 4 provides a method to summarize/collapse the extent to a Neutrosophic fuzzy set into the membership to a classical fuzzy set.

Hereafter, the fuzzy operators are presented in details.

- The instruction CREATE neutrosophic FUZZY OPERATOR on Line 2 defines a fuzzy operator named Neutrosophify, which returns the extent to a Neutrosophic fuzzy set according to the model neutrosophic that was defined on Line 1 (Listing 1). The goal of this operator is to generate the extent to a Neutrosophic fuzzy set from three real numbers, named m, n and p that are received as parameters, as specified in the clause PARAMETERS.
  The clause PRECONDITION verifies that every parameter is in the range $[0, 100]$; if not satisfied, the fuzzy operator cannot be applied.
  The clauses EVALUATE compute the basic degrees mu, nu, and pi of the fuzzy-set model neutrosophic. In particular, the degree mu (nu, and pi, respectively) is obtained by dividing the parameter m (n and p, respectively) by 100.
- The instruction CREATE neutrosophic FUZZY OPERATOR on Line 3 computes the operator *EXCLUSIVE OR* for the Neutrosophic model. The operator is named ExclusiveOR and returns the extent to a fuzzy set according to the fuzzy-set model neutrosophic defined on Line 1 (Listing 1).
  The clause PARAMETERS specifies that the operator must receive the extents to two neutrosophic fuzzy sets, as parameters named x and y.
  Since there is no PRECONDITION to verify, the clauses EVALUATE immediately evaluate, respectively, the basic degrees mu, nu, and pi of the resulting extent to a neutrosophic fuzzy set.
- The instruction CREATE FUZZY OPERATOR on Line 4 evaluates the membership degree to a classical fuzzy set (notice the absence of the model name between the keywords "CREATE" and "FUZZY").
  The operator is named FuzzifyNeutrosophic and its goal is to summarize the extent to a Neutrosophic fuzzy set in a single membership degree to a classical fuzzy set.
  The clause PARAMETERS defines the x parameter, which has to receive the extent to a Neutrosophic fuzzy set.
  The clause EVALUATE specifies a mathematical expression to summarize/collapse degrees in x into one single degree; notice that it enhances the role of the degree mu when associated to low values of the degrees nu and pi.
  The result of the expression in the clause EVALUATE is used as x-axis value against the polyline function defined by the clause POLYLINE. The polyline function is depicted in Fig. 5: notice that it boosts values in the range $[0.5, 1]$, while it penalizes values lower than 0.5. The corresponding y-axis value is the final membership degree provided by the operator.
  Notice that this is the form of the statement CREATE FUZZY OPERATOR that was available before extending the language with multi-grade models. Indeed, the classical fuzzy-set model is still natively defined in *J-CO-QL*$^+$, thus it does not need be declared in advance to be used.

To conclude this sub-section, a quick note about the clauses POLYLINE: each clause EVALUATE, in an instruction CREATE FUZZY OPERATOR, either classical or multi-grade, can be followed by an optional clause POLYLINE to determine the final degree. If no clause POLYLINE is present, a default $[(0, 0), (1, 1)]$ polyline is considered.

5.2.3. Performing a soft query in J-CO-QL$^+$ with multi-grade fuzzy sets

Listing 3 shows how to exploit the fuzzy-set model and the fuzzy operators declared in Listings 1 and 2 to perform a non-trivial soft query.

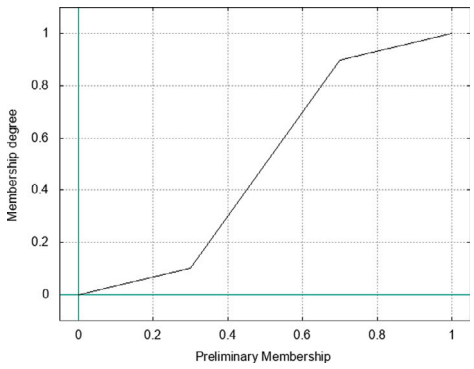


Fig. 5. Polyline function defined in the fuzzy operator FuzzifyNeutrosophic.

Listing 3 *J-CO-QL*$^+$ Script: Soft querying.

```
5. USE DB MarketSurvey
    ON SERVER jcods 'http://127.0.0.1:17017';

6. GET COLLECTION ProductReviews@MarketSurvey;

7. FILTER
    CASE WHERE WITH   .evalFancy, .evalSporty,
                      .evalCommercial
    GENERATE
      CHECK FOR
        neutrosophic FUZZY SET fancy
          USING Neutrosophify ( .evalFancy.yes,
                                .evalFancy.no,
                                .evalFancy.idk ),
        neutrosophic FUZZY SET sporty
          USING Neutrosophify ( .evalSporty.yes,
                                .evalSporty.no,
                                .evalSporty.idk ),
        neutrosophic FUZZY SET appreciation
          USING Neutrosophify ( .evalCommercial.yes,
                                .evalCommercial.no,
                                .evalCommercial.idk ),
        neutrosophic FUZZY SET efficacy
          USING appreciation AND
            ExclusiveOR (EXTENT(fancy), EXTENT(sporty))
        FUZZY SET wanted
          USING FuzzifyNeutrosophic (EXTENT(efficacy))
      ALPHACUT
        0.70 ON wanted
  ;

8. SAVE AS ProductEvaluations@MarketSurvey;
```

Hereafter, each instruction is presented in details.

- The instruction USE DB on Line 5 connects to a database named MarketSurvey managed by a *J-CO-DS* server (see Section 4.1) running on the local machine.
- The instruction GET COLLECTION on Line 6 retrieves a collection of *JSON* documents, named ProductReviews, containing the data collected by *ACME* at the end of the survey. Fig. 4 shows an example of document in the collection. The acquired collection becomes the temporary collection of the process (see Section 4.2).
- The role of the instruction FILTER on Line 7 is to select and transform the documents in the current temporary collection in order to perform the soft query committed to *ACME*.
  Fig. 6 shows how the execution of the instruction FILTER modifies the document in Fig. 4.
  - The clause CASE WHERE selects those documents, in the input temporary collection, holding the root-level structured fields evalFancy, evalSporty and evalCommercial, needed to perform the soft query. Indeed, collections of *JSON* documents can be heterogeneous; thus, it is necessary to focus on documents having the fields to process.

– In the following section GENERATE, the clause CHECK FOR adds the root-level field ~fuzzysets with the extents to the fuzzy sets declared in the following branches FUZZY SET. In Fig. 6, the field ~fuzzysets is highlighted by a black-dashed box.
– The first branch neutrosophic FUZZY SET adds the extent to a neutrosophic fuzzy set named fancy. The extent is generated by the soft condition specified by the clause USING, by exploiting the operator Neutrosophify defined on Line 2 (Listing 2) to which the values of the sub-fields evalFancy.yes, evalFancy.no, and evalFancy.idk are passed as actual parameters. In Fig. 6, the extent to the Neutrosophic fuzzy set fancy is highlighted by a red box: notice the sub-field type that specifies the fuzzy-set model "neutrosophic"; also notice that every other sub-field is related to a basic or derived degree of the fuzzy-set model neutrosophic.
– The second branch neutrosophic FUZZY SET evaluates the extent to a second neutrosophic fuzzy set named sporty in the same way, by passing the values of the sub-fields evalSporty.yes, evalSporty.no and evalSporty.idk to the operator Neutrosophify. In Fig. 6, the extent to the sporty Neutrosophic fuzzy set is highlighted by a blue box.
– The extent to a third neutrosophic fuzzy set, named appreciation, is evaluated by the third branch neutrosophic FUZZY SET; the soft condition in the clause USING passes the values of the sub-fields yes, no, and idk in the field evalCommercial to the operator Neutrosophify as actual parameters. In Fig. 6, the extent to the neutrosophic fuzzy set appreciation is highlighted by a green box.
– The fourth branch FUZZY SET evaluates the extent to a neutrosophic fuzzy set named efficacy, by exploiting the operator AND of the fuzzy-set model neutrosophic, defined on Line 1 (Listing 1), as well as the fuzzy operator ExclusiveOR defined on Line 3 (Listing 2), to which the extents to the Neutrosophic fuzzy sets sporty and fancy are provided by the built-in function EXTENT.
In Fig. 6, the extent to the Neutrosophic fuzzy set efficacy is highlighted by a yellow box.
Notice that the Neutrosophic fuzzy set efficacy actually corresponds to Query 1.
– The last branch FUZZY SET evaluates the membership degree to a classical fuzzy set named wanted; the soft condition in the clause USING exploits the operator FuzzifyNeutrosophic defined in Line 6 (Listing 2) to summarize the extent to the Neutrosophic fuzzy set efficacy into a classical fuzzy set. In Fig. 6, the fuzzy set wanted is shown at the end of the field ~fuzzysets. Notice that, in the case of the classical fuzzy-set model, the corresponding field is not structured but its value represents directly the membership degree. This solution was the approach that was adopted before introducing multi-grade models in *J-CO-QL*$^+$.
– The clause ALPHACUT selects those documents whose membership degree to the fuzzy set wanted is no less than 0.7.

• Finally, the instruction SAVE AS on Line 8 saves the resulting collection into a new collection named ProductEvaluations in the MarketSurvey database.

Summarizing, the novelties introduced in the *J-CO-QL*$^+$ language and in the data model are listed hereafter.

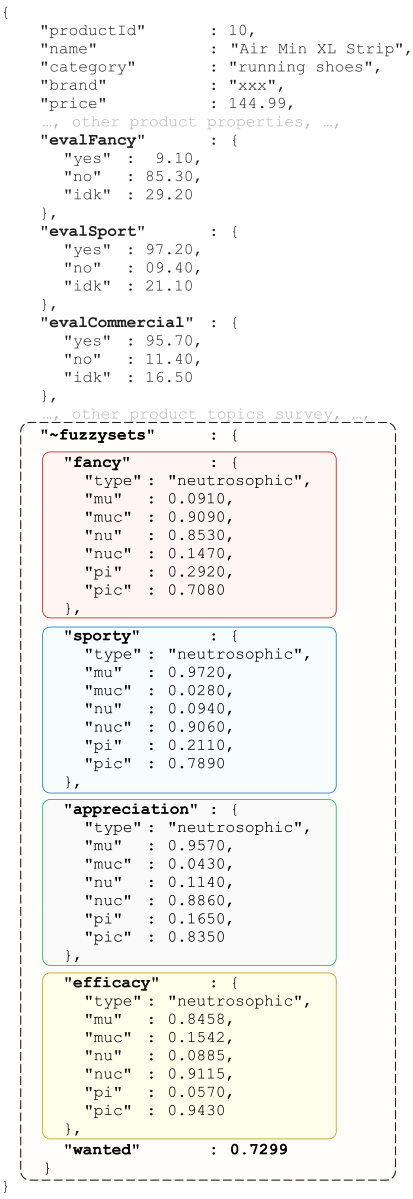


Fig. 6. Example of output document for the *J-CO-QL*$^+$ script.

• The novel statement CREATE FUZZY SET MODEL allows for defining a multi-grade fuzzy-set model.
• The statement CREATE FUZZY OPERATOR has been extended and now is CREATE *[model]* FUZZY OPERATOR; this allows for separately specifying each single basic degree. If *model* is missing, it downgrades to the original version for classical fuzzy sets.
• Within the clause CHECK FOR, the former FUZZY SET *name* has evolved as *[model]* FUZZY SET *name*, so as it is able to specify a soft condition on the specified *model* and generate the corresponding extent; if *model* is missing, the target model is the classical fuzzy-set model, as before the introduction of multi-grade fuzzy-set models.
• Finally, in the root-level field ~fuzzysets in *JSON* documents, the internal sub-fields are able to represent extents of multi-grade fuzzy sets, as nested sub-documents.

6. Conclusions and future work

The paper presented a quick survey about the evolution of Fuzzy-Set Theory from the first classical model provided by Zadeh in 1965, towards more complex models. The attention is mostly focused on multi-grade fuzzy sets and, in particular, the Intuitionistic, Neutrosophic and Pythagorean fuzzy-set models are presented. Interval-Valued fuzzy sets (a particular case of Type-2 Fuzzy Sets) and Hesitant fuzzy-sets are also discussed, but they are not considered in the remainder of the paper.

Each model was formally presented, in order to let common aspects emerge. As a result, many models can be classified as "Multi-grade Fuzzy-Set Models", in that they are based on multiple degrees, such as membership and non-membership.

Subsequently, a meta-model for describing multi-grade fuzzy-set models and their operators in a unified way has been proposed. To show the goodness of the proposal, the meta-model has been used to provide a description of classical, Intuitionistic, Neutrosophic and Pythagorean fuzzy-set models and some possible variations; clearly, the meta-model can be applied to any user-defined multi-grade fuzzy-set model.

Finally, the latest novelties introduced in the *J-CO* Framework and in the *J-CO-QL*$^+$ query language have been presented. These novelties allow users to define novel multi-grade fuzzy-set models and operators to perform advanced soft querying through *J-CO-QL*$^+$ scripts.

A plausible case study was presented, to show how to perform a non-trivial soft query through a user-defined fuzzy-set model (namely, an extension of the Neutrosophic model). Indeed, a few *J-CO-QL*$^+$ instructions allowed for defining the novel fuzzy-set model, three fuzzy operators and actually performing the soft query on a collection of *JSON* documents previously stored within a *JSON* repository.

To the best of the authors' knowledge, the proposal introduced by this paper is the first attempt to define a generic meta-model, together with its implementation within a real software tool for defining fuzzy-set models and exploit them in practical activities.

The *J-CO* Framework is under a continuous evolution, through which novel features are continuously added. Thus, it is possible to foresee some future developments. The most obvious one is to extend the proposed meta-model, so as to capture a wider range of fuzzy-set models, including Hesitant and Type-2 Fuzzy sets; additionally, the extensions to the meta-model will give rise to further extensions of *J-CO-QL*$^+$.

However, for some multi-grade models (such as Intuitionistic Fuzzy Sets) complex transformations based on aggregations have been proposed in the literature, which imply the aggregation of membership and non-membership degrees of multiple items (documents); to achieve this goal, a specific statement for defining fuzzy aggregators, which constitutes a missing feature in the current version of *J-CO-QL*$^+$, is about to be introduced.

Finally, a library of scripts with the definitions for the most common multi-grade models will be published, so as to allow users to incorporate and use them within their soft queries.

The *J-CO* Framework is available on a public GitHub repository.[1]

Declaration of competing interest

The authors declare that they have no known competing financial interests or personal relationships that could have appeared to influence the work reported in this paper.

Data availability

No data was used for the research described in the article.

[1] Github repository of the *J-CO* Framework: https://github.com/JcoProjectTeam/JcoProjectPage.

References

[1] T. Bray, The javascript object notation (JSON) data interchange format, 2014, URL https://www.rfc-editor.org/rfc/rfc7159.txt.
[2] K. Chodorow, MongoDB: The Definitive Guide: Powerful and Scalable Data Storage, O'Reilly Media, Inc, 2013.
[3] L.A. Zadeh, Fuzzy sets, Inform. Control 8 (1965) 338–353.
[4] G. Psaila, P. Fosci, J-CO: A platform-independent framework for managing geo-referenced JSON data sets, Electronics 10 (5) (2021) 621.
[5] G. Bordogna, S. Capelli, D.E. Ciriello, G. Psaila, A cross-analysis framework for multi-source volunteered, crowdsourced, and authoritative geographic information: The case study of volunteered personal traces analysis against transport network data, Geo-Spat. Inf. Sci. 21 (3) (2018) 257–271.
[6] P. Fosci, G. Psaila, Towards flexible retrieval, integration and analysis of json data sets through fuzzy sets: a case study, Information 12 (7) (2021) 258.
[7] P. Fosci, G. Psaila, Soft integration of geo-tagged data sets in J-CO-QL+, ISPRS Int. J. Geo-Inf. 11 (9) (2022) 484.
[8] P. Fosci, G. Psaila, Intuitionistic fuzzy sets in J-CO-QL+? in: International Conference on Soft Computing Models in Industrial and Environmental Applications, Springer, Cham, 2022, pp. 134–145.
[9] L.A. Zadeh, The concept of a linguistic variable and its application to approximate reasoning—I, Inf. Sci. 8 (3) (1975) 199–249.
[10] K. Atanassov, Intuitionistic fuzzy sets, Int. J. Bioautom. 20 (2016) 1.
[11] K.T. Atanassov, S. Stoeva, Intuitionistic fuzzy sets, Fuzzy Sets Syst. 20 (1) (1986) 87–96.
[12] K.T. Atanassov, K.T. Atanassov, Intuitionistic Fuzzy Sets, Springer, 1999.
[13] F. Smarandache, A unifying field in logics: Neutrosophic logic, in: Philosophy, American Research Press, 1999, pp. 1–141.
[14] H. Wang, F. Smarandache, Y. Zhang, R. Sunderraman, Single valued neutrosophic sets, Infinite Study 12 (2010).
[15] F. Smarandache, Neutrosophic logic-a generalization of the intuitionistic fuzzy logic, Multispace Multistructure 4 (2010) 396, Neutrosophic transdisciplinarity (100 collected papers of science).
[16] R.R. Yager, Pythagorean fuzzy subsets, in: 2013 Joint IFSA World Congress and NAFIPS Annual Meeting (IFSA/NAFIPS), IEEE, 2013, pp. 57–61.
[17] R.R. Yager, Properties and applications of pythagorean fuzzy sets, Imprecision and Uncertainty in Information Representation and Processing: New Tools Based on Intuitionistic Fuzzy Sets and Generalized Nets (2016) 119–136.
[18] F. Smarandache, Neutrosophic set is a generalization of intuitionistic fuzzy set, inconsistent intuitionistic fuzzy set (picture fuzzy set, ternary fuzzy set), pythagorean fuzzy set, spherical fuzzy set, and q-rung orthopair fuzzy set, while neutrosophication is a generalization of regret theory, grey system theory, and three-ways decision (revisited), J. New Theory (29) (2019) 1–31.
[19] A.U. Rahman, M.R. Ahmad, M. Saeed, M. Ahsan, M. Arshad, M. Ihsan, A study on fundamentals of refined intuitionistic fuzzy set with some properties, J. Fuzzy Extens. Appl. 1 (4) (2020) 279–292.
[20] M. Saeed, M.R. Ahmad, A.U. Rahman, Refined pythagorean fuzzy sets: properties, set-theoretic operations and axiomatic results, J. Comput. Cogn. Eng. 2 (1) (2023) 10–16.
[21] J.M. Mendel, R.B. John, Type-2 fuzzy sets made simple, IEEE Trans. Fuzzy Syst. 10 (2) (2002) 117–127.
[22] J.M. Mendel, Uncertain rule-based fuzzy systems, Introd. New Dir. 684 (2017).
[23] I.B. Turksen, Interval valued fuzzy sets based on normal forms, Fuzzy Sets and Systems 20 (2) (1986) 191–210.
[24] V. Torra, Hesitant fuzzy sets, Int. J. Intell. Syst. 25 (6) (2010) 529–539.
[25] P. Sevastjanov, L. Dymova, K. Kaczmarek, On the neutrosophic, pythagorean and some other novel fuzzy sets theories used in decision making: invitation to discuss, Entropy 23 (11) (2021) 1485.
[26] D.C. Blair, Information retrieval, 2nd ed. c.j. Van rijsbergen. London: Butterworths; 1979: 208 pp. Price: $32.50, J. Am. Soc. Inf. Sci. 30 (6) (1979) 374–375.
[27] P. Bosc, O. Pivert, SQLf: a relational database language for fuzzy querying, IEEE Trans. Fuzzy Syst. 3 (1) (1995) 1–17.
[28] P. Bosc, O. Pivert, SQLf query functionality on top of a regular relational database management system, in: Knowledge Management in Fuzzy Databases, Springer, 2000, pp. 171–190.
[29] J. Galindo, J.M. Medina, O. Pons, J.C. Cubero, A server for fuzzy SQL queries, in: International Conference on Flexible Query Answering Systems, Springer, 1998, pp. 164–174.
[30] J. Kacprzyk, S. Zadrozny, Fquery for access: Fuzzy querying for a windows-based DBMS, in: P. Bosc, J. Kacprzyk (Eds.), Fuzziness in Database Management Systems, in: Studies in Fuzziness, vol. 5, Physica, Heidelberg, 1995.
[31] S. Zadrozny, J. Kacprzyk, Fquery for access: towards human consistent querying user interface, in: Proceedings of the 1996 ACM Symposium on Applied Computing, 1996, pp. 532–536.
[32] G. Bordogna, G. Psaila, Customizable flexible querying in classical relational databases, in: Handbook of Research on Fuzzy Information Processing in Databases, IGI Global, 2008, pp. 191–217.
[33] G. Bordogna, G. Psaila, Modeling soft conditions with unequal importance in fuzzy databases based on the vector p-norm, in: Proceedings of the IPMU, Malaga, 2008.

[34] G. Bordogna, G. Psaila, Soft aggregation in flexible databases querying based on the vector p-norm, Int. J. Uncertain. Fuzziness Knowl.-Based Syst. 17 (supp01) (2009) 25–40.
[35] P. Bosc, H. Prade, An introduction to the fuzzy set and possibility theory-based treatment of flexible queries and uncertain or imprecise databases, in: Uncertainty Management in Information Systems, Springer, 1997, pp. 285–324.
[36] J.M. Medina, O. Pons, M.A. Vila, Gefred: A generalized model of fuzzy relational databases, Inform. Sci. 76 (1) (1994) 87–109.
[37] J. Galindo, A. Urrutia, M. Piattini, Fuzzy Databases: Modeling, Design, and Implementation, IGI Global, 2006.
[38] J. Galindo, New characteristics in FSQL, a fuzzy SQL for fuzzy databases, WSEAS Trans. Inf. Sci. Appl. 2 (2) (2005) 161–169.
[39] J. Kacprzyk, S. Zadrozny, SQLf and FQUERY for access, in: Proceedings Joint 9th IFSA World Congress and 20th NAFIPS International Conference (Cat. No. 01TH8569), Vol. 4, IEEE, 2001, pp. 2464–2469.
[40] Z.M. Ma, L. Yan, Generalization of strategies for fuzzy query translation in classical relational databases, Inf. Softw. Technol. 49 (2) (2007) 172–180.
[41] A. Urrutia, L. Tineo, C. Gonzalez, FSQL and SQLf: Towards a standard in fuzzy databases, in: Handbook of Research on Fuzzy Information Processing in Databases, IGI Global, 2008, pp. 270–298.
[42] J. Galindo, Handbook of Research on Fuzzy Information Processing in Databases, IGI Global, 2008.
[43] B.K. Abir, G.T. Amel, Towards fuzzy querying of NoSQL document-oriented databases, in: DBKDA 2015, 2015, p. 163.
[44] F. Mehrab, A. Harounabadi, Apply uncertainty in document-oriented database (MongoDB) using F-XML, J. Adv. Comput. Res. 9 (3) (2018) 87–101.
[45] J.M. Medina, I.J. Blanco, O. Pons, A fuzzy database engine for mongoDB, Int. J. Intell. Syst. Online library (2022).
[46] P. Fosci, G. Psaila, Powering soft querying in J-CO-QL with JavaScript functions, in: International Workshop on Soft Computing Models in Industrial and Environmental Applications, Springer, 2021, pp. 207–221.
[47] G. Psaila, P. Fosci, Toward an anayist-oriented polystore framework for processing JSON geo-data, in: International Conferences on WWW/Internet, ICWI 2018 and Applied Computing 2018, Budapest; Hungary, 21-23 October 2018, IADIS (International Association for Development of the Information Society), 2018, pp. 213–222.