On the use of Hesitant Fuzzy Linguistic Term Set in FLINTSTONES

Francisco J. Estrella, Rosa M. Rodríguez, Macarena Espinilla and Luis Martínez

Abstract—The use of linguistic information to model and manage uncertainty in Decision Making (DM) has been a key subject of many proposals in the literature. The 2-tuple linguistic model and its extensions in linguistic DM has been very successful and extensive due to their flexibility and accuracy. Flintstones is a novel fuzzy linguistic decision tool enhancement suite that implements tools to facilitate the solving of linguistic DM problems that model the linguistic information with such a model and its extensions. However, both the 2-tuple linguistic model and Flintstones can not deal with uncertain situations modelled linguistically in which experts hesitate among several linguistic terms. For these cases, recently, it has been proposed the use of Hesitant Fuzzy Linguistic Term Sets (HFLTS) that have attracted a lot of research interest, mainly regarding its application in DM. Hence in this contribution it is proposed an extended version of Flintstones that includes the ability and functionality of dealing with HFLTS in linguistic decision problems and enables the integration, validity and performance of hesitant linguistic decision models and operators.

I. INTRODUCTION

A Decision Making (DM) process is used to select the best alternative from a set of alternatives. Generally, DM problems are defined under uncertainty, which may have non-probabilistic nature, in such cases the use of linguistic information closer to human beings cognitive model has provided reliable and flexible results [5], [14], [20], [25].

Fuzzy logic and fuzzy linguistic approach [29] provide tools to model and manage such an uncertainty by means of linguistic variables, improving the flexibility and offering reliability of the decision models in different fields [17] such as risk assessment [26], sensory evaluation [6] or marketing strategy selection [7]. The use of linguistic information involves processes of Computing with Words (CWW) in which the objects of computation are words or sentences from a natural language and the results are also expressed in a linguistic expression domain [18], [19].

The 2-tuple linguistic model has been compared with different linguistic representations and computing models for CWW in DM and the 2-tuple linguistic model has been showed as the most appropriate model in linguistic DM, considering the CWW paradigm [12], [21]. The main advantage of the 2-tuple linguistic model is its computational model that offers linguistic results in the original linguistic domain in a precise way. Furthermore, this model has been extended [4], [8], [9], [10], [11], [17] to solve different problems in complex decision frameworks with linguistic information.

This contribution has been supported by the research project TIN2012-31263 and AGR-6487

- Heterogeneous frameworks in which the assessments are expressed in different domains such as numerical, interval or linguistic [11], [15].
- Multi-granular linguistic frameworks in which the assessments are expressed in multiple linguistic scales [4], [8], [10], [13].
- Unbalanced linguistic frameworks in which the assessments are expressed in an unbalanced linguistic scale [2], [9].

Due to the big amount of proposals related to the 2-tuple linguistic model and its extensions, a software tool suite called Flintstones 1 (Fuzzy LINguisTic deciSion TOols eN-hancEment Suite), has been recently developed for solving linguistic DM problems defined in the previous frameworks modelled by the 2-tuple linguistic model and its extensions, offering linguistic results that facilitate their understandability.

Flintstones has been designed as a component-based application with special attention to its design aspects such as reusability or the inclusion of new functionalities. Therefore, the deployment of new features, new solving processes and new aggregation operators in order to extend the functionality of Flintstones is relatively simple.

Even though, the previous linguistic models and the software Flintstones can be applied in many decision problems under uncertainty modelled linguistically. It should be pointed out the importance of modeling uncertainty related to hesitancy in DM. Recently Torra [24] introduced the concept of Hesitant Fuzzy Sets (HFS) to deal with hesitancy in quantitative settings and Rodríguez et al. proposed in [22] the concept of Hesitant Fuzzy Linguistic Term Sets (HFLTS) that keeps the basis on the fuzzy linguistic approach [29] and extends the idea of HFS to linguistic contexts. So, the joint use of HFLTS and a context-free grammar allows managing decision situations where experts might hesitate among different linguistic terms, using linguistic expressions close to the natural language [22], [23]. Furthermore, the interest and utility of the concept of HFLTS in DM has grown very quickly in the literature [1], [16], [22], [23], [27], [30]. Therefore, it seems quite useful and promising to provide a software that allows solving, testing and validating new proposals related to HFLTS and DM.

Hence and due to the fact that Flintstones was conceived as a software suite for linguistic DM easy to extend with new components and functionalities, this contribution develops a set of features in Flintstones related to HFLTS to solve DM

1http://sinbad2.ujaen.es/flintstones
problems with HFLTS. To do so, we present the architecture and resolution scheme of the suite with HFLTS as well as the functionality with HFLTS. This updated version allows analyzing and testing the results of DM problems with HFLTS, using different aggregation operators.

The remaining of this contribution is structured as follows: Section II reviews the theoretical fundamentals of HFLTS. Section III presents Flintstones and its functionality to support DM processes using HFLTS. Section IV shows the use of HFLTS in Flintstones. Finally, in Section V, conclusions are drawn.

II. BACKGROUND

This section revises some basic concepts of HFLTS and several aggregation operators defined to aggregate this type of information that will be implemented in our proposal.

A. Hesitant Fuzzy Linguistic Term Sets

The concept of HFLTS was introduced to improve the flexibility of the elicitation of linguistic information when experts hesitate among different linguistic terms to express their assessments or preferences.

Definition 1: [22] Let $S$ be a linguistic term set, $S = \{s_0, \ldots, s_g\}$, a HFLTS, $H_S$, is an ordered finite subset of consecutive linguistic terms of $S$,

$$H_S = \{s_i, s_{i+1}, \ldots, s_j\}, \text{ such that } s_k \in S, k \in \{i, \ldots, j\}$$ (1)

To facilitate the computations with HFLTS, it was proposed the concept of envelope of a HFLTS defined as follows.

Definition 2: [22] The envelope of a HFLTS $\text{env}(H_S)$, is a linguistic interval whose limits are obtained by means of its upper bound and lower bound:

$$\text{env}(H_S) = [H_{S-}, H_{S+}], \quad H_{S-} \leq H_{S+}$$ (2)

where the upper bound is defined as $H_{S+} = \max\{s_k\}$ and the lower bound $H_{S-} = \min\{s_k\}, \forall s_k \in H_S, k \in \{i, \ldots, j\}$.

More operations and properties for HFLTS were defined in [22].

Although the concept of HFLTS can be used directly by experts to provide multiple linguistic terms, such elements are not similar to the expressions used by human beings’ in real world problems. Therefore, Rodríguez et al. proposed the use of context-free grammars to generate linguistic expressions similar to human beings’ expressions which are easily represented by HFLTS. A context-free grammar $G_H$ which generates comparative linguistic expressions close to the expressions used by experts in DM problems was introduced in [22] and extended in [23].

Definition 3: [23] Let $G_H$ be a context-free grammar and $S = \{s_0, \ldots, s_g\}$ be a linguistic term set. The elements of $G_H = (V_N, V_T, I, P)$ are defined as follows:

$V_N = \{\text{primary term}, \text{composite term}\}$, $\text{primary term}$, $\text{composite term}$, $\langle\text{unary relation}\rangle$, $\langle\text{binary relation}\rangle$, $\langle\text{primary term}\rangle$, $\langle\text{composite term}\rangle$, $\langle\text{conjunction}\rangle$

$V_T = \{\text{at most}, \text{at least}, \text{between, and, so, \ldots, s_g}\}$

$I \in V_N$

$P = \{I ::= \langle\text{primary term}\rangle | \langle\text{composite term}\rangle$
2) Max lower operator: This symbolic operator obtains the best of the minimum linguistic terms.

**Definition 6:** [22] Let \( X = \{x_1, \ldots, x_n\} \) be a set of alternatives, \( C = \{c_1, \ldots, c_m\} \) a set of criteria, \( S = \{s_0, \ldots, s_g\} \) a linguistic term set and \( \{H^i_S(x_i)/i \in \{1, \ldots, n\}, j \in \{1, \ldots, m\}\} \) a set of HFLTS, the max lower operator consists of two steps:

- Apply the lower bound for each HFLTS associated with each \( x_i \):
  \[
  H_{S-}^i(x_i) = \{H^1_S(x_i), \ldots, H^m_S(x_i)\}, \ i \in \{1, \ldots, n\} \quad (6)
  \]

- Obtain the maximum linguistic term for each \( x_i \):
  \[
  H_{S-}^i(x_i) = \max\{H_{S-}^i(x_i) / j \in \{1, \ldots, m\}\}, \ i \in \{1, \ldots, n\} \quad (7)
  \]

3) Hesitant fuzzy linguistic weighted averaging operator: This operator generalizes the Linguistic Weighted Averaging operator to aggregate HFLTS by using the convex combination. Being the convex combination for HFLTS as follows:

**Definition 7:** [27] Let \( S = \{s_0, \ldots, s_g\} \) be a linguistic term set, \( H^1_S \) and \( H^2_S \) be two HFLTS, the convex combination of \( H^1_S \) and \( H^2_S \) is

\[
C^1(w_1, H^1_S, w_2, H^2_S) = w_1 \odot H^1_S \oplus w_2 \odot H^2_S =
\]

\[
= (C^2(w_1, s^1, w_2, s^2)|s^1 \in H^1_S, s^2 \in H^2_S) \quad (8)
\]

where \( w_k \geq 0 (k = 1, 2) \) and \( w_1 + w_2 = 1 \).

**Definition 8:** [27] Let \( X = \{x_1, \ldots, x_n\} \) be a set of alternatives, \( C = \{c_1, \ldots, c_m\} \) a set of criteria, \( S = \{s_0, \ldots, s_g\} \) a linguistic term set, \( \{H^i_S(x_i)/i \in \{1, \ldots, n\}, j \in \{1, \ldots, m\}\} \) a set of HFLTS, and \( w = (w_1, \ldots, w_m)^T \) a weighting vector of \( H^i_S(x_i) \) with \( w_j \geq 0 (j = 1, \ldots, m) \) and \( \sum_{j=1}^{m} w_j = 1 \), the HLWA operator is defined as follows.

\[
HLWA(H^1_S(x_i), \ldots, H^m_S(x_i)) =
\]

\[
= C^k\{w_j, H^j_S(x_i), j = 1, \ldots, m\} =
\]

\[
= w_1 \odot H^1_S(x_i) \oplus (1-w_1) \odot C^{k-1}\{w_h/\sum^{m}_{j=2} w_j, H^h_S(x_i)\} \quad (9)
\]

with \( h = \{2, \ldots, m\} \) and where \( i \in \{1, \ldots, n\} \) is the number of alternatives.

4) Hesitant fuzzy linguistic ordered weighted averaging operator: This operator extends the Linguistic Order Weighting Averaging operator to HFLTS by using also the convex combination.

**Definition 9:** [27] Let \( X = \{x_1, \ldots, x_n\} \) be a set of alternatives, \( C = \{c_1, \ldots, c_m\} \) a set of criteria, \( S = \{s_0, \ldots, s_g\} \) a linguistic term set, \( \{H^i_S(x_i)/i \in \{1, \ldots, n\}, j \in \{1, \ldots, m\}\} \) a set of HFLTS, and \( w = (w_1, \ldots, w_m)^T \) a weighting vector of \( H^i_S(x_i) \) with \( w_j \geq 0 (j = 1, \ldots, m) \) and \( \sum_{j=1}^{m} w_j = 1 \), the HLOWA operator is defined as follows.

\[
HLOWA(H^1_S(x_i), \ldots, H^m_S(x_i)) =
\]

\[
C^k\{w_j, H^j_S(x_i), j = 1, \ldots, m\} =
\]

\[
= w_1 \odot H^1_S(x_i) \oplus (1-w_1) \odot C^{k-1}\{w_h/\sum^{m}_{j=2} w_j, H^h_S(x_i)\} \quad (10)
\]

Many approaches have been proposed to compute the associated weighting vector \( w = (w_1, \ldots, w_m)^T \) of the ordered weighted averaging operator (see [28]).

III. IMPLEMENTING HFLTS IN FLINTSTONES

This section shows how it has been carried out the integration of HFLTS in Flintstones. To do so, we first present the architecture and resolution scheme of the suite. Then, we expose its functionality to support DM processes using HFLTS.

A. Flintstones Architecture

Flintstones has been developed as an Eclipse Rich Client Platform (Eclipse RCP)\(^2\) application that is a platform to build and deploy rich client applications developed by IBM and maintained by Eclipse Community. The key value of Eclipse RCP is that it allows quickly developing professional applications with native look-and-feel on multiple platforms which can be easily extended, modified and reused. Eclipse RCP is based on a component-based architecture, which tries to solve some common problems in software development such as reusing, maintaining, extending and modifying.

An Eclipse RCP application consists of several Eclipse components, also called plug-ins, bundles or OSGi components. Flintstones includes more than 15 components, which can be grouped into four basic types: i) libraries, ii) Graphical User Interface (GUI), iii) methods and iv) operators.

Figure 1 illustrates the architectural diagram of Flintstones which shows only some of the components implemented in it.

B. Flintstones Resolution Scheme

To solve DM problems, Flintstones adapts the common decision resolution scheme proposed in [3] (see Figure 2) whose main steps are:

1) Defining framework. In this first step, the set of alternatives and the set of criteria which characterize the alternatives are established as well as the group of experts that will evaluate the alternatives are fixed. Furthermore, the expression domains used to assess the alternatives are also defined. Flintstones allows creating different expression domains using wizards that guide the user through the process. This step is performed in the Framework perspective of Flintstones.

\(^2\)http://www.eclipse.org/home/categories/rcp.php
2) Gathering information. In this step, experts provide their assessments for each criterion of each alternative in the expression domains defined in the framework. This step is performed in the Gathering perspective of Flintstones.

3) Rating alternatives. In this step, it is computed the global assessment for each alternative using the selected solving process. Each solving process establishes a set of steps to solve the DM problem. This step is performed in the Rating perspective of Flintstones.

C. Inclusion of HFLTS in Flintstones

The component-based architecture of Flintstones allows including new functionalities without changing its components. However, in some cases, in order to maintain components cohesion and not increase the coupling among them, it is more convenient the modification of existing components to include new functionalities.

The integration of HFLTS in Flintstones involves to include multiple new features, such as support for hesitant assessments, graph visualization and storage of these assessments, the addition of Hesitant solving processes and implementation of aggregation operators that can operate computationally with HFLTS.

Below, we present briefly the way in which the above features have been included in Flintstones.

- **Support for Hesitant Assessments.** As shown in Figure 1, Flintstones provides support to the assessment (valuations) at its core. To maintain this structure, the functionalities implemented to support hesitant assessments were added to this core. Furthermore, the integration into Flintstones core of the basic support for hesitant assessments, allows us developing in future works new solving processes based on HFLTS and new aggregation operators that can operate with this type of information. The main changes that have been made are:
  - Adding hesitant valuation to Flintstones flintstones.valuation component so that it is possible to use valuation of this nature in the suite.
  - Adding support for saving hesitant valuation to Flintstones flintstones.io component so that it is possible save and load these kind of valuations.
  - Adding hesitant assessment panel to Flintstones flintstones.rcp component so that it is possible gathering assessments of this nature in a simple way (see Figure 3).
  - Adding graph visualization of hesitant assessments to Flintstones flintstones.rcp component so that it is possible visualize clearly these assessments (see Figure 3).

As previously mentioned, the inclusion of these functionalities in Flintstones has been carried out modifying the existing components in order to maintain the cohesion of its components and not increase the coupling among them. However, these functionalities could have been also included in Flintstones with new components, using for the inclusion of them mechanisms as extension points, Eclipse RCP functionality designed to extend a specific functionality, and Fragments, a special type of Eclipse component designed to extend a specific component.

- **Hesitant Solving Process.** The solving process with HFLTS is carried out in Flintstones using a special type of components called methods. Flintstones uses
a common scheme to define methods that allows any user to develop a new method reusing the functionality available in the suite, such as the aggregation of the assessments or the unifying of them (see Figure 4). Furthermore, *Flintstones* uses its own component manager which allows using external components. So, anyone interested in developing a new solving process can do it without having the source code of the tool suite. In general terms, the creation of a solving process is carried out by implementing an OSGi component which satisfies the following conditions:

- The component must indicate in its manifest file the general solving process data such as its name or its description.
- The component must indicate in its manifest file that it is a *Flintstones* solving process.
- The component must include an activator class that allows *Flintstones* activating the solving process.
- The component must include a solving process class that allows *Flintstones*:
  - Knowing the solving process conditions such as the allowed expression domains or the nature of the assessments that it employs.
  - Knowing the set of steps carried out by the solving process.
  - Knowing the data needed for each step and the results obtained after its running.

**Fig. 4. Method scheme**

The hesitant solving process has been developed following this scheme, having been created the *flintstones.method.hesitant* component that fulfills the conditions imposed by *Flintstones*. In the case of this solving process, it specifies:

- A single step to solving a problem of DM, the aggregation of the set of assessments, which takes as input the assessments provided by experts for each criterion of each alternative as well as the used aggregation operators to compute the collective valuation for each alternative.
  - The condition that the solving process only can be carried out if the DM problem is a problem with a single expert in which all assessments are expressed in the same linguistic expression domain.

- **Aggregation Operators.** To implement a new aggregation operator in *Flintstones* is similar to develop a solving process. The suite employs a defined scheme that allows any user developing and distributing their own aggregation operators (see Figure 5). Creation of an aggregation operator is carried out by implementing an OSGi component which satisfies the following conditions:
  - The component must indicate in its manifest file the aggregation operator description, i.e., its name and type of supported valuations.
  - The component must indicate in its manifest file that it is a *Flintstones* aggregation operator.
  - The component must include an activator class that allows *Flintstones* activate the aggregation operator.
  - The component must include a aggregation operator class that allows *Flintstones* to obtain the collective assessment of a set of assessments.

**Fig. 5. Aggregation operator scheme**

See *Flintstones* website for further detailed information. All aggregation operators implemented in *Flintstones* for HFLTS have been developed following this scheme, having been created the following components that fulfill the conditions imposed by *Flintstones*: *Max_lower* [22], *Min_upper* [22], *HLWA* [27] and fi-
nally, HLOWA [27].

IV. ILLUSTRATIVE EXAMPLE

To show the usefulness and effectiveness of Flintstones with HFLTS, in this section a multicriteria DM problem is solved by using the software tool suite. The case study of this illustrative example can be found in Flintstones website\(^3\) together with a repository of case studies and datasets for different DM problems with linguistic and complex frameworks.

Let us suppose the manager of garment company which plans to develop a new series of sports jackets. After preliminary screening, there are three possible textile fabrics \(X = \{x_1, x_2, x_3\}\) to be assessed, according to three benefit criteria \(C = \{c_1, c_2, c_3\}\), which are respectively: quality, reliability and business reputation.

Due to the lack of information and knowledge about the DM problem, it is difficult for the manager of the company to provide all assessments by means of single linguistic terms. Instead, the manager can provide his hesitant assessments with several linguistic terms as well as comparative linguistic expressions that are close to the natural language.

In this decision problem is employed the context-free grammar \(G_H\), introduced in Def. 3, that uses the linguistic term set shown in Figure 6, to generate the comparative linguistic expressions used to assess the set of alternatives.

To solve the DM problem with Flintstones, each main step of the common decision resolution scheme, which was presented in Figure 2, is described in detail.

A. Framework

The elements involved in the DM problem are included in Flintstones, i.e, the manager of garment company \(e_1\), the set of three alternatives \(X = \{x_1, x_2, x_3\}\) and the set of three criteria \(C = \{c_1, c_2, c_3\}\).

Furthermore, in order to assess the set of alternatives, a linguistic term set with 7 labels is established. In Figure 7 is illustrated the framework of this illustrative example.

\(^3\)http://sinbad2.ujaen.es/flintstones/?q=node/10

B. Gathering Information

Once the framework has been defined to assess the different alternatives, the knowledge must be obtained from the manager of the garment company.

So, the manager of garment company can provide assessments in \(S\) with single linguistic terms, several linguistic terms as well as comparative linguistic expressions.

The assessments given by the manager of garment company to the set of alternatives are shown in Figure 8. For example, the assessment provided by the manager \(e_1\) for the alternative \(x_2\) about the criterion \(c_3\) is expressed by means of the expression: At least High.

C. Selecting a solving process to rate the alternatives

Due to the fact that our illustrative example is defined with hesitant assessments, in this step, a solving process for a DM problem with hesitant linguistic information must be carried out (see Figure 9) to conduct the rating process.

The solving process for hesitant linguistic information includes an aggregation phase to obtain a result for each alternative which is used to rank the set of alternatives. Transparently to the user, in this phase, the suite converts the comparative linguistic expressions into HFLTS by means of the transformation function \(E_{G_H}\).

Currently, Flintstones has implemented the following aggregation operators for HFLTS: min_upper, max_lower, HLWA and HLOWA.
From Figure 10 to 13 are illustrated the obtained results with each implemented aggregation operator, respectively. Note that the HLWA and HLOWA aggregation operators have been used with the following weight vector $w = (0.5, 0.25, 0.25)^T$.

Finally, in view of the results, a ranking among the alternatives is established with the purpose of identifying the best one. Although the obtained results are different in some cases, depending on the used aggregation operator, we can see that the set of alternatives keep the same ranking regardless of the aggregation operator utilized $x_2 > x_3 > x_1$. Therefore, the best alternative is the second textile fabric $x_2$.

With this illustrative example, it has been shown as Flintstones achieves very easily the results of a DM problem, using different aggregation operators. Therefore, Flintstones is an excellent option to make comparisons among aggregation operators in a DM problem in an easy way.

V. CONCLUSIONS

Flintstones is a software tool suite to solve linguistic decision making problems under uncertainty, this suite is a component-based application and its design is focused on reusability and the inclusion of new features. In this contribution has been implemented a set of functionalities built into Flintstones to solve decision making problems using Hesitant Fuzzy Linguistic Term Sets (HFLTS). Also it has been described the architecture and the resolution scheme developed in the suite. The set of functionalities implemented to support HFLTS into Flintstones facilitate the development and integration of future aggregation operators and decision models that can operate with linguistic hesitant valuations in order to analyze, test and validate their results.

REFERENCES


