



HAL
open science

Fuzzy4U: A fuzzy logic system for user interfaces adaptation

Tanguy Giuffrida, Sophie Dupuy-Chessa, Jean-Philippe Poli, Éric Céret

► To cite this version:

Tanguy Giuffrida, Sophie Dupuy-Chessa, Jean-Philippe Poli, Éric Céret. Fuzzy4U: A fuzzy logic system for user interfaces adaptation. 13th International Conference on Research Challenges in Information Science, RCIS 2019, May 2019, Brussels, Belgium. hal-03283635

HAL Id: hal-03283635

<https://hal.science/hal-03283635>

Submitted on 12 Jul 2021

HAL is a multi-disciplinary open access archive for the deposit and dissemination of scientific research documents, whether they are published or not. The documents may come from teaching and research institutions in France or abroad, or from public or private research centers.

L'archive ouverte pluridisciplinaire **HAL**, est destinée au dépôt et à la diffusion de documents scientifiques de niveau recherche, publiés ou non, émanant des établissements d'enseignement et de recherche français ou étrangers, des laboratoires publics ou privés.

Fuzzy4U: A fuzzy logic system for user interfaces adaptation

Tanguy Giuffrida
Univ. Grenoble-Alpes,
CNRS, Grenoble INP, LIG
38000 Grenoble, France.
tanguy.giuffrida@univ-
grenoble-alpes.fr

Sophie Dupuy-Chessa
Univ. Grenoble-Alpes,
CNRS, Grenoble INP, LIG
38000 Grenoble, France.
sophie.dupuy@univ-
grenoble-alpes.fr

Jean-Philippe Poli
CEA LIST
91191 Gif-sur-Yvette,
France.
jean-philippe.poli@cea.fr

Éric Céret
Univ. Grenoble-Alpes,
CNRS, Grenoble INP, LIG
38000 Grenoble, France.
eric.ceret@univ-
grenoble-alpes.fr

Abstract— Adapting User Interfaces to various technological devices is most often than not a part of human-computer interaction requirements. Although many studies addressed this topic some challenges remain, such as context uncertainty and combination of adaptation rules. This article represents an attempt at tackling these challenges, using fuzzy logic to handle adaptation. It proposes an architecture where an adaptation engine is supported by both fuzzy logic and Boolean logic, and illustrated by a prototype. The relevance of such approach has been studied through a theoretical comparison and an experiment including eight experts.

Keywords—HCI adaptation, context of use, fuzzy logic, uncertainty, rules

I. INTRODUCTION

Adapting user interfaces (UI) based on their context of use, characterized by the triplet <user, platform and environment> [1], is a well-known requirement in Human Computer Interaction (HCI) design processes. It involves dynamically modifying an interface so that it responds more precisely and specifically to each and every user's needs and expectations. Nevertheless, this change ought to consider the constraints given by the device used for this interaction (i.e. the platform) and the environment in which this interaction ought to take place.

Nowadays, many studies on adaptive user interfaces can be found in the literature [2, 3, 4, 5, 6]. However, several scientific challenges remain, especially when the ability to *adapt* includes user considerations as well as the platform and environment [7, 8, 9]. This implies 1) the identification of the diverse users' characteristics and 2) the definition of how to integrate them in the calculation of the solution. Difficulties exist concerning both facets.

First of all, providing fine-grain adaptation requires to have a fine-grain vision of the context and therefore to consider a large number of characteristics. Focusing on the user, the cluster Virtual User Modeling and Simulation (VUMS) has identified more than 250 useful characteristics to model a human user [10]. However, this model does not cover all requirements, since it focuses mainly on physical abilities [1, 11].

A second challenge relies upon the uncertainty on characteristics values, in particular for those related to users. Every value, whether they are captured (e.g. due to a potential lack of accuracy of sensors or in the inference) or declared (e.g. because of different perceptions by various users, personal estimation of the values) may not be certain. This uncertainty is usually disregarded [1, 11, 12, 13, 14].

Once the context has been modeled, the adaptive system still has to calculate the necessary UI adaptations according to the current context. Classically, this calculation is performed by the way of *rules*. There are different approaches to express these adaptation rules, such as models

transformations in model-driven engineering [15], design patterns [11] or the development of precomputed solutions where one of these solutions is chosen according to logical rules [16].

The potentially huge amount of context characteristics may lead to the definition of complex and competitive rules that, in turn, can induce a combinatorial issue. Rules might indeed need to be combined. For instance, several of them can impact the font rendering (font type, spaces between characters and words according to dyslexia, font color according to color perception, and so on). They can also compete with each other (e.g. increasing the font size when the user has poor visual acuity while decreasing it when the display surface is small). Classic methods to choose a rule among a set of possible rules exist, such as weights [17].

A more interesting method would be to reach a compromise between all the rules. The large number of potential solutions for this approach opens a new issue: the combinatorial explosion [18], i.e. the difficulty to handle the multiplicity of possible compromises and make a choice among them. This combination may lead to performance issues [19].

This article proposes a solution based on fuzzy logic in the calculation of UI adaptations. It suggests expressing the UI adaptation rules in this formalism. It presents a comparative analysis of the results obtained from a fuzzy logic engine as well as from a more traditional approach in Boolean logic, accompanied by expert opinions. This article represents an extension of previous works [20] published in French., which focused more on a comparison between Boolean and fuzzy logics and gave less details on the experimental part of the work (prototype and experts' feedback).

This article is structured as follows. Section II presents the current state of the art in UI adaptation. Section III introduces the basic concepts of fuzzy logic. Section IV shows the architecture of the demonstrator and its extension to fuzzy logic with the expected benefits. Section V describes the fuzzy rules used, their comparison with the pre-existing Boolean rules and presents the opinion of experts on this subject. The article ends with a summary of the work and perspectives.

II. STATE OF THE ART

A. Related Work

The state-of-the-art is based on the study of two kinds of studies: articles dealing with UI adaptation and those using fuzzy logic. There are many studies dealing with adaptation to the context of use. Only the most representative will be discussed here.

Cameleon [1] is a reference in model-based approaches. It promotes the dynamic generation of UIs adapted from

models to their context of use. It is based on a process of reification within four levels of abstractions: tasks model, abstract UI, concrete UI and final UI. Theoretically, all contextual characteristics may be considered. However in reality, just a few are presented. The uncertainty of their values is not mentioned. The adaptation rules represent model transformations applied conditionally depending on the context of use. Rules should be mutually exclusive, which is particularly complex to achieve, for example when two characteristics should induce antagonistic adaptations.

"Adaptation Integration System" [7] is a first example based on Cameleon. It aims to integrate requirements of accessibility for people with disabilities and includes the adaptation rules in the development process. Nonetheless it uses few and predefined characteristics.

Maria [12] is a model-based user interface description language that allows for dynamic UI generation adapted to the different platforms used by each user during a migration process. The article focuses on the platform aspect of the context of use.

Cedar [13] provides an architecture that facilitates the development of adaptive UI within an enterprise resource planning (ERP), built on role-based simplifications. Cedar creates effective adaptations, but in a controlled environment.

[14] defines a 3-layer architecture for the conception of adaptive UI. Adaptation targets *smart environments* and is based on the context of use. The data used are directly read from sensors (e.g. localization systems, temperature sensors, e-health devices) without managing uncertainty. These sensors mainly focus on the environment and the adaptation process is limited to a certain scope defined at the design time.

MyUI [11] is a UI development infrastructure to improve accessibility through the adaptation of users' needs. The adaptation rules are managed thanks to design patterns and their combinations are handled manually during conception. MyUI lists a large number of user characteristics (e.g. contacts information, personality, emotional/physiological/mental states, nutrition, facial expression). However, these characteristics are mainly declared, and MyUI defines the different states of a characteristic through fixed thresholds. In terms of rules, MyUI focuses on improving accessibility for specific disabilities.

Supple [21] supports the automatic generation of UI tailored to the user's abilities, devices, tasks, and preferences. Visual and motor abilities represent the main user characteristics supported during the adaptation. Forty user interface factors (e.g. font size) are endorsed to optimize the layout of the UI elements. While slightly outside our research space, it has been noted in [11] that the performance of Supple proves problematic because it exceeds a reasonable calculation time; this would be even truer in the case of a complex combination of rules.

[19] examines the use of Machine Learning to adapt UI to the context of use, mainly in order to optimize user interactions and limit their errors. Machine learning makes it possible to manage complex and uncertain data, and to calculate complex inferences from them. However, such an algorithm also requires gathering a substantial knowledge dataset before it can offer adaptations that can improve the user experience. In addition, while the article presents potential applications, none were implemented.

A selection of studies relying on fuzzy logic to support adaptive systems can also be gathered from the literature:

FSAM [22] is a tool based on fuzzy logic allowing choosing the service that is most suited to a user according to one's context of use. This study is however limited to a choice between predefined services and does not address any HCI issues.

[23] proposes a fuzzy logic approach for the development of adaptive mobile applications, supporting the diversity of mobile platforms (hardware, OS, API, etc.) and the performance capabilities of each platform. The user is not considered, limiting the number of characteristics and their uncertainty factor.

[24] presents a tool making it possible for car drivers and passengers to develop applications thanks to a vocal interface. Fuzzy logic is used to calculate the level of expertise and concentration of the user: beginner, intermediate, or advanced. Each level leads to a higher or lower level of options and information provided. The system relies on 7 fuzzy rules that consider 10 characteristics (number of applications created, number of requests for help, number of oaths, ...) to calculate a single output: the level of expertise. This use of fuzzy logic does not correspond to our requirements in terms of adaptation rules because of the small number of characteristics used in input, their lack of uncertainty (e.g.: the amount of help requests is increased when the user asks for help), and the manual handling of the rules' combination.

[25] presents another approach in fuzzy logic that addresses webpage adaptation to improve user experience. This work is based on use cases, presenting seven of them (e.g. Place an order, Perform a search, ...). Data are inferred from the users' behaviors, thus limiting the size of the user model and the set of adaptation rules possible. It does not address the question of rules combination that seems central to UI adaptation.

[26] proposes a fuzzy logic-based HCI adaptation mechanism that considers three cognitive user factors defined by [27]: "Verbal / Imager" dimension, "Analyst / Wholist" dimension, and working memory capacity. Each of these aspects has specific implications in terms of design. As a result, this work only manages a small number of characteristics. The approach also includes manual management of the composition of rules, which limits the power of the system and its ability to handle unexpected situations.

Finally, [28] presents an approach for HMI customization using fuzzy logic based on semantic relationships between UI components and user preferences to select the most suitable components. The case study presented gives only one rule - having two inputs and one output - and the combination between the entries has been handled manually within the rule. This work does not consider aspects that seem important to us, such as the extent of the context of use and the combination of rules.

B. Analysis

Our analysis of existing studies was based on criteria relating to the difficulties identified to achieve an adaptation engine. Thus they concerned: 1) the extent of the context of use considered in terms of coverage of the three aspects (user, platform, environment), 2) the number of users' characteristics addressed to give clues about the management of a few or numerous uncertainty characteristics, 3) the management of uncertainty within the values of the characteristics of the context of use, 4) the complexity and 5) the combination of adaptation rules.

TABLE 1 presents the experimental aspects of the previously mentioned studies, with Harvey's balls.

Based on said criteria, the works going from Cameleon to Supple can be organized (cf. beginning of table 1). Indeed, all of them deal with dynamic adaptation according to a context of use of varying size (thus including a greater or lower number of characteristics). However, none of these studies address the rest of our criteria: uncertainty, rules complexity, or combination.

Papers [22, 23, 24, 25, 26, 28] use fuzzy logic and consider values uncertainty. In [22, 23], the composition of rules was managed by fuzzy logic, but the lack of an extensive context of use limited the contribution. None of these pieces of work clearly addressed the issue of rules combination.

Finally, the paper from Genaro Motti [19] had the best overall ratings, with 4 of our 5 criteria fulfilled. However, the use of Machine Learning required an important knowledge - difficult to acquire and maintain - before offering adaptations that effectively meet users' needs.

In order to overcome the limits of existing studies and to propose a solution that makes it possible to manage context uncertainty, many characteristics and adaptation rules at the same time, we investigated the possibilities offered by a UI adaptation system based on fuzzy logic. This solution goes beyond existing proposals in fuzzy logic by taking an interest in the complexity of rules and their combinations.

TABLE 1: COMPARISON OF STATE OF THE ART WORKS ACCORDING TO TARGETED CRITERIA

Approaches	Scope of the context of use				
	Quantity of Uncertainty	Combination of rules	Complexity		
user characteristics					
Cameleon [1]	●	●	○	○	●
Minon [7]	●	●	○	○	●
MyUI [11]	●	●	○	○	●
Maria [12]	●	○	○	○	○
Cedar [13]	●	●	○	○	○
3-layer architect. [14]	●	○	○	○	○
Supple [21]	●	●	○	○	○
FSAM [22]	●	●	●	○	●
Desruelle [23]	○	○	●	○	●
Cueva-Fernandez [24]	○	○	●	○	○
Nyongesa [25]	●	●	●	○	○
Papatheocharous [26]	○	●	●	○	○
Soui [28]	○	○	●	○	○

III. FUZZY LOGIC

In order to understand the use of fuzzy logic for UI adaptation, this section details its principles and inferring process.

A. Fuzzy Expert Systems

Fuzzy expert systems are expert systems based on fuzzy logic. Fuzzy logic can be viewed as an extension of classical logic, which allows to handle uncertainty: whereas classical logic deals only with two values (true and false), fuzzy logic considers an infinite number of real values between 0 and 1. This type of logic was introduced by Zadeh [29] in order to

get closer to the human reasoning and to avoid the threshold effects pertaining to classical logic approaches. The particularity of fuzzy expert systems is that rules can be written in a syntax close to natural language.

B. Concept of Fuzzy set and Linguistic Variable

Fuzzy logic is based on the concept of fuzzy sets. Let X be the universe of discourse, i.e. a collection of objects denoted x . A fuzzy set A of X is totally characterized by a function $\mu_A: X \rightarrow [0,1]$, called *membership function*. This function indicates how much an object x belongs to X . In crisp sets, an object does or does not belong to the set. Fuzzy sets are useful to describe imprecision or uncertainty. Fig. 1 represents a fuzzy set defines on $[0;10]$ with a triangular membership function and which can symbolize “approximately 5”. Membership functions can have different shapes (e.g. instance triangular, trapezoidal, Gaussian, or Bell functions).

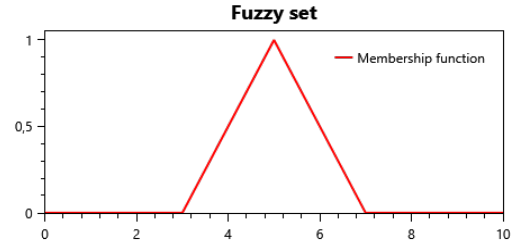


Fig. 1: Example of fuzzy set

The next concept in fuzzy logic, called linguistic variable [30], It corresponds to a triplet (V, X, T_V) where:

- V is the name of the linguistic variable,
- X is the domain on which it is defined (e.g. \mathbb{R} or $[0,10]$),
- $T_V = \{T_1, T_2, \dots\}$ is a finite collection of fuzzy sets called terms which qualify V (e.g.: “cold”, “hot”).

Each fuzzy set T_i of T_V is associated with a name and is called linguistic term. Fig. 2 shows a linguistic variable called “temperature” with three terms (“cold”, “medium”, “hot”) and their membership functions, defined on $[-20, 50]$.

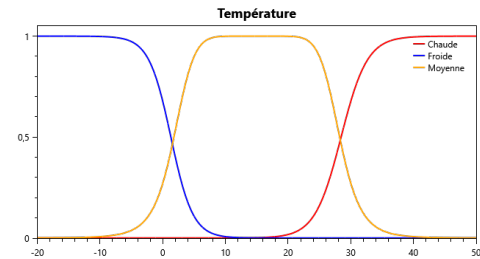


Fig. 2: Example of a linguistic variable “temperature”

C. Fuzzy Rules to Model Expert Knowledge

Previous definitions were used to formalize human reasoning through a set of rules. These rules depend on a combination of elementary fuzzy propositions. An elementary fuzzy proposition is the definition of “ V is A ” from a linguistic variable (V, X_V, T_V) where A is a term of T_V . For instance, if one used the linguistic variable “temperature” of Fig. 2, a fuzzy proposition could be “temperature is cold”. It is evaluated from its membership function μ_{cold} : for a particular temperature t , the truth-value of the proposition “temperature is cold” is given by $\mu_{cold}(t)$. The truth-value of a fuzzy proposition belongs to the interval

$[0, 1]$ in \mathbb{R} unlike classical logic where the truth-value of a proposition would be either 0 (false) or 1 (true).

A fuzzy expression stands for a composition of a set of elementary fuzzy propositions or other fuzzy expressions. The composition is created by selecting logical operators: negation (not), conjunction (and) and disjunction (or). The exclusive or (xor) is not usable.

Fuzzy propositions can be viewed as a special case of fuzzy expressions. For instance, let two fuzzy propositions be “V is A” using (V, X_x, T_x) and “W is B” using (W, X_w, T_w) . Thus, “V is A and W is B”, “V is A or not W is B” are fuzzy expressions. The truth-value of a fuzzy expression is given by the application of all the operator functions on the values of their operands.

There are different formulations of conjunction and disjunction operators, but they all have to fulfill the properties of respectively triangular norms and triangular co-norms. A triangular norm (t-norm) is a commutative, associative, monotonic function $T: [0; 1] \times [0; 1] \rightarrow [0; 1]$ which accepts 1 as neutral element. A triangular co-norm (t-co-norm) is a commutative, associative, monotonic function $\perp: [0; 1] \times [0; 1] \rightarrow [0; 1]$, which accepts 0 as neutral element. For instance, Zadeh's norm is the minimum function and Zadeh's co-norm is the maximum function.

The negation operator is a function $N: [0; 1] \rightarrow [0; 1]$, which generally associates to a fuzzy value n the value $1-n$.

The expressions are used to express the knowledge of an expert as fuzzy rules. A fuzzy rule is composed of a premise (also called antecedent) and a conclusion (also called consequent) in the form “IF premise THEN conclusion”.

A premise is a fuzzy expression and a conclusion is a fuzzy statement which value can be of different nature. There are two different types of rule conclusions and fuzzy rules: 1) A conclusion can be a fuzzy proposition. Then, the template of a fuzzy rule becomes “IF temperature is low THEN sweating is low”. 2) A conclusion can be a mathematical function of the inputs of the premise. These rules are called Takagi-Sugeno fuzzy rules and their template is “IF temperature is low THEN sweating= f (temperature).”

The rules are mapping the inputs to the outputs of the system regarding a fuzzy implication function.

The THEN part of fuzzy rule denotes a fuzzy implication function I such as $I: [0; 1] \rightarrow [0; 1]$ that performs on two fuzzy values. It verifies the properties of the implication operator in classical logic when the two operands equal 0 or 1.

The implication operator in classical logic states that either we do not have the antecedent or we have the consequent. There are different formulations of fuzzy implication function inspired from multi-valued logic. However, some conjunctive functions may be used instead of the implicative function. It states that inputs and outputs are possible together. That is the case for the well-established Mamdani fuzzy rules that make use of the minimum function.

D. Fuzzy Inference

Now that the basics of fuzzy logic have been introduced, we can focus on the process of inference in fuzzy expert systems. In the previous section, we introduced how rules work on their own. In this section, we present how rules work together to compute the outputs and thus process the

inference. In the remainder of this paper, we will use Mamdani inference process [31], summarized in Fig. 3.

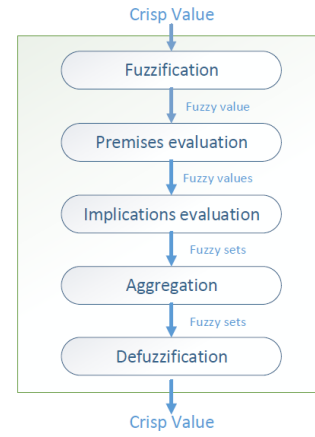


Fig. 3: Overview of Mamdani fuzzy inference.

The first step consists in the evaluation of each elementary fuzzy proposition in the premise of the rules: this step is called *Fuzzification*. It takes a crisp value (the value of the input in the real world) and associates a fuzzy value to it regarding the membership function of the term of the linguistic variable involved in the fuzzy proposition. For instance, with the linguistic variable “temperature” of Fig. 2 and its term “cold”, for a temperature of -10°C , “temperature is cold” is evaluated at 1 (i.e. -10°C is really a cold temperature). For a temperature of 1°C , both “temperature is cold” and “temperature is medium” equal 0.5 (approximately).

The premises are then computed with the application of the t-norms, t-co-norms, negation function and other operators according to the rules involved. The value of each premise is a fuzzy value also called *rule activation*. For instance, considering Zadeh's t-co-norm (i.e. the max function) and a temperature of 15°C , the value of the premise “temperature is cold or temperature is medium” equals $\max(1,0)=1$.

The implication function is then applied on the conclusion of each rule. The result is a fuzzy set describing the output. For instance, if the premise of the rule has been evaluated at 0.75, with the implication method being the minimum function, the output's fuzzy set is as shown in Fig. 4.

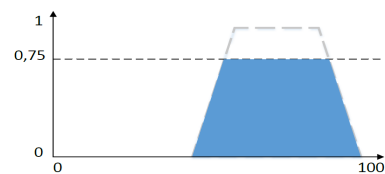


Fig. 4: Result of the implication

The diverse resulting fuzzy sets of a same output are aggregated by an aggregation function, often the max function for Mamdani systems, resulting in a fuzzy set for each output. For instance, Fig. 5 illustrates the process of aggregation of two fuzzy sets with the max aggregation function.

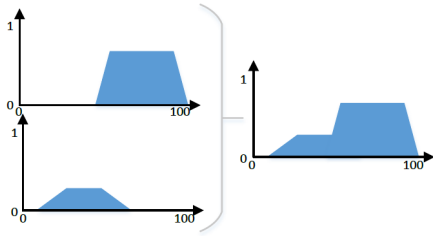


Fig. 5: Application of the max aggregation on two fuzzy sets

Each output's fuzzy set is changed into a crisp value: this process is called *Defuzzification*. There are several defuzzification methods; the best-known defuzzification method computes the centroid of the area under the membership function (the blue area of the aggregated fuzzy set in Fig. 5).

IV. THE FUZZY4U SYSTEM

This section describes a system named Fuzzy4U, which uses fuzzy logic for UI adaptation. The architecture of this system will first be explored, followed by discussions on the integration of fuzzy logic into its adaptation engine.

A. Architecture Presentation

The Fuzzy4U system is based on [32]'s architecture, proposed by our team for HCI adaptation proposed by our team for HCI adaptation (Fig. 6), illustrated by a prototype. It consists mainly of three components: an inference engine, an adaptation engine and the interactive system.

The inference engine is in charge of dynamically deducing the values of the context of use from captured data. Data can be provided by sensors (e.g. ambient light API in JavaScript) or ad hoc modules (e.g. the number of 'valid clicks' – clicks done on appropriate interactors – and 'invalid clicks' – clicks done on empty spaces – computed). Thanks to a meta-UI, the user can also state the values and even override the values deduced by the inference engine. The user can, for example, declare one's visual acuity or override the detected ambient light. From all those data, the inference engine infers a representation of the context of use. For instance, the user's hand accuracy is deduced from the numbers of "valid" and "invalid" clicks.

As we already mentioned, data can also be declared by the user through a meta-UI. Here, the user can select one of two states within the interface. First, the meta UI presents the characteristics of the context of use (user's dyslexia, color blindness, visual acuity, etc.) making it possible for the user to refine one's context model and thus indirectly control the adaptation. Secondly, the meta-UI exhibits the characteristics of the interfaces (font size, word spacing, etc.): when

modifying their values, the user can overpass the system's decisions.

Inferred and declared data are sent to the adaptation engine. When both inferred and declared data related to the same characteristic are available, only the value declared by the user is considered. The engine then uses these data and adaptation rules to select an appropriate version of the interface (named a variant) on which presentation parameters (colors, font,...) will be applied. For example, if the precision level of the user's hand is low, the adaptation engine can increase the space considered for the click on the interactors to improve the UI usability for this specific user.

The inference engine is distinct from the adaptation engine to ensure the separation of concerns: the first one is responsible for transforming raw values (e.g. 130 clicks on an interactor and 212 in an empty area) into values usable by the second one (e.g. "very low" hand accuracy).

In line with our research aims, the adaptation engine was initially based on rules in Boolean logic only. In this former version, characteristics had textual values (e.g. visual acuity could be 'normal', 'low', 'veryLow' or 'extraLow') and rules were coded thanks to a series of conditions (simple or combined), where the index of the characteristics value in its set of possible values was considered for expressing ranges of possibilities. For instance, a rule in Boolean logic calculating font size based on visual acuity was:

```

If      visualAcuity <= 1 then fontSize = 12 ;
elseif visualAcuity <= 2 then fontSize = 14 ;
elseif visualAcuity <= 3 then fontSize = 16 ;
else   fontSize = 18 ;

```

In this rule, 1, 2 and 3 were respectively the indices of 'normal', 'low' and 'veryLow' in the set ('normal', 'low', 'veryLow', 'extraLow'). In this approach, uncertainty was not considered, neither in the values of the characteristics nor in the adaptation rules. Moreover, the approach did not make it possible to deal with rules that would have required complex or subtle reasoning. In our example with four values for visual acuity, only four font sizes can be computed as a result, leading to poor adaptation granularity and threshold effects during adaptations. Finally, when results depend on several characteristics, the condition can only be written by embedding several *if... then... else* structures and thus leads to redundant sub-conditions. This makes for poor use and maintainability in complex cases. In other words, this approach supported no uncertainty, poor complexity and poor combination possibilities.

The adaptation engine aims at (a) choosing an adapted UI variant and (b) computing the adaptations required for

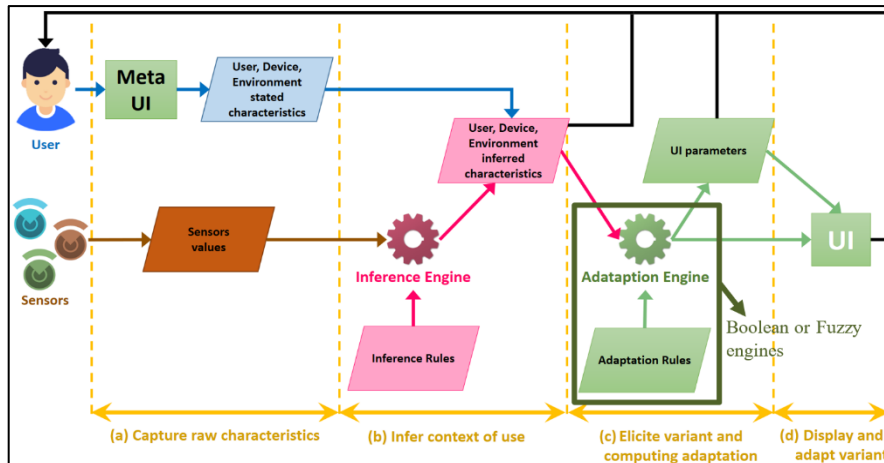


Fig. 6: Logical architecture used by the team for HCI adaptation

enhancing the rendering. A variant represents a version of one UI dedicated to some spatial organization. For instance, the homepage of a website could have two variants: one for a computer screen and the other for a Smartphone. Using different versions of a UI avoids the complex calculation of their specific spatial organization. This choice is done thanks to distance calculation: each variant is associated to an expected context of use (e.g. a display surface of 900x600 pixels and a visual acuity at least “normal”). A UI is thus associated to various variants (usually two or three), each of them in turn associated to an expected context of use. The current context of use is then compared to all the expected contexts of use of the UI variants. The variant whose characteristics are, in average, the closest to the current situation, is elicited.

The architecture is generic and has been implemented with Web technologies and is based on NodeJS, allowing the execution of JavaScript on both the server and client sides. The objects used respect similar structures on both sides and are stored in text files in JSON format on the server. Therefore, the common structure is kept with manual modifications. The context model and the user model were made according to these characteristics. The module in Boolean logic is executed internally in NodeJS, and the rules are stored in JSON. The full architecture is available as a patent in [32].

B. Integration of Fuzzy Logic

The originality of this paper lies in the choice of adaptation engine in fuzzy logic in addition to an already existing Boolean logic engine. Technically, the fuzzy logic engine uses the ExpressIF tool developed by CEA Tech, called through a Web Service via an HTTP request. To allow comparison and add some flexibility a parameter was added by developers for them to call either approach at the touch of a button (in Fig. 6, the “Adaptation engine” and “Adaptation rules” exist in both approaches).

In the following subsections, we will see how adaptation rules can be expressed in fuzzy logic. We will highlight how fuzzy logic has the potential to help breach the barriers of uncertainty of the context of use, the complexity of rules and their combination.

1) Uncertainty

All characteristics have a degree of uncertainty, whether they are inferred by the engine (e.g. accuracy of the calculation of the hand precision thanks to a ratio between valid clicks and invalid clicks) or declared by the user (e.g. two users with identical visual acuity evaluating it differently).

Uncertainty of the different characteristics’ values can be expressed thanks to membership functions of the variables in fuzzy logic. For example, Fig. 7 shows the definition of the visual acuity variable: the value is defined on the abscissa (here from 0 to 4), and the certainty of the four different states (‘normal’, ‘low’, ‘veryLow’ and ‘extraLow’) is specified on the ordinates. The ‘normal’ state is thus certain when the visual acuity varies from 0 to 0.5. From 0.5 and up to 1.5, the certainty of the ‘normal’ state decreases in favor of the ‘low’ state, which gains in certainty. At 1.5, the ‘low’ state is therefore at 100% certainty while the ‘normal’ state is at 0%. The crossing of two curves (at abscissas 1, 2 and 3) thus indicates a probability of 50% for each of the two states.

The introduction of uncertainty in variable definition makes it possible to limit the effects of thresholds (see rule in section IV.A) that induce a sudden change from one context value to another one. It is particularly useful in the management of rules for which several input variables affect the same output variable, as in section IV.B.3.

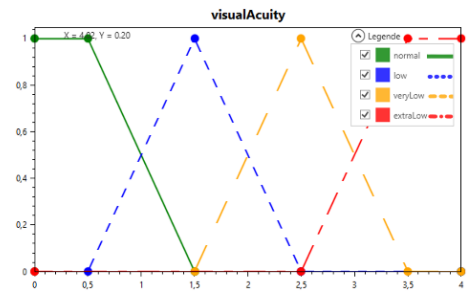


Fig. 7: Definition of the visualAcuity linguistic variable

2) Complexity

Next to uncertainty, the expression of adaptation rules represents another of our problem: rules with many thresholds can become complex quickly in Boolean logic. For example, let us consider the rule based on the user's visual acuity level to adapt the font size. As mentioned above, several thresholds are used (1, 2, 3 and 4) and the final granularity of the output variable is relatively small (4 possible values: 12, 14, 16, 18). While this granularity can be improved it remains at the expense of the number of thresholds. For such cases, the implementation of this rule in fuzzy logic would be to define the input and output linguistic variables with only two membership functions each (see Fig. 8 and details below) and to create the rules in fuzzy logic (Fig. 9).

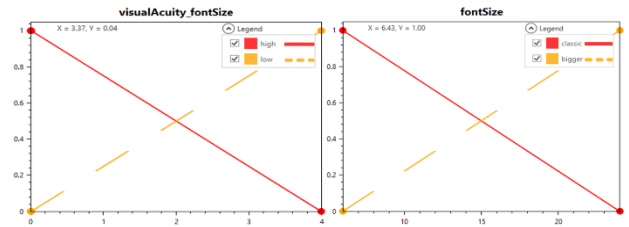


Fig. 8: Definition of the visualAcuity and fontSize linguistic variables

```

If visualAcuity_fontSize is high then fontSize is classic
If visualAcuity_fontSize is low then fontSize is bigger

```

Fig. 9: Fuzzy rules defined for adapting font size based on visual acuity

Given the definitions of the domain variable, the fuzzy logic set of rules becomes simpler, with only two cases.

Using the defuzzification method of weighted centers of gravity, the fontSize output value will be between 12 and 18 points. Its principle is to compute the centers of gravity of the output fuzzy subset for each of the rules, and to calculate a sum weighted by the area under the curve. Considering all rules with the same output, the defuzzified value can be written as:

$$d(output) = \sum_i C_i \times A_i / A_i$$

where C_i is the value of the center of gravity and A_i the area under the curve for the i^{th} rule concerning this output.

Defuzzification is usable on continuous linguistic variables. It was adapted here to discrete variables. As discrete domains are ordered, it gives more sense to the center of gravity. In this case, the area under the curve is replaced by the corresponding value in the histogram that serves as the membership function.

Two examples for this method are illustrated in figures 10 and 11. Fig. 10 shows that for a visual acuity of 0 in input, the term "high" is activated at 100% and the term "low" at 0%. According to the first rule, the activation of the "classic" output term is also done at 100%, represented by the red area. The center of gravity of this square triangle is located at

the first third of its base, which is a value of 12 for the `fontSize` output variable with a domain between 6 and 24.

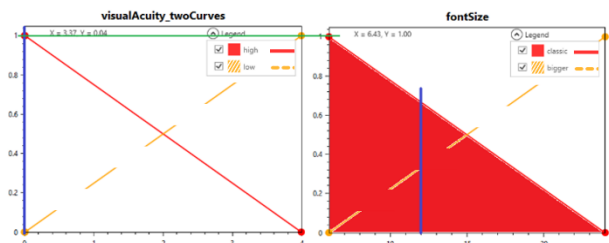


Fig. 10: Diagram of the weighted centers of gravity's defuzzification method for an input value of 0

For a visual acuity's input at 1 (Fig. 11), the term "high" is activated this time at 75% and the term "low" at 25%. This is represented by the two green lines and the activation of the "classic" output term at 75%, represented by the red area and the activation of the "bigger" output term at 25%, represented by the orange area. The centers of gravity of these two quadrilaterals are then calculated (12.3 for the barycenter bound to "classic" and 10.071 for the second one, related to "bigger"), then weighted according to their areas (respectively 84.375 and 39.375), in order to obtain the result of 13.5 points for the `fontSize` variable. We thus have the "bigger" state, taking slightly more weight when compared to the "classic" state, with a font size of 13.5 instead of the value 12 in the previous case. These results are therefore continuous and follow a function $\text{fontSize} = 12 + \text{visualAcuity} * 1.5$.

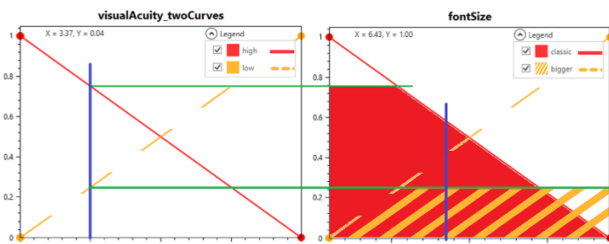


Fig. 11: Diagram of the weighted centers of gravity's defuzzification method for an input value of 1

This example shows how fuzzy logic avoids threshold effects completely. It enables the simplification of the expression of adaptation rules while obtaining more precise output values.

3) Combination of rules

The last challenge to address is the combination of rules that can occur when multiple input variables can affect the same output variable [18]. For example, it may be necessary to increase the clickable space around the interactors either when the visual acuity is weak, or when the accuracy of the clicks (and thus, of the hands) is weak. A problem arises when one input variable is high while the other is low.

Boolean logic does not provide easy answers to this problem. It is possible to choose one input value or the other, but, in which case, it is necessary to establish a system of weighting or prioritization of rules in order to give a result.

Fuzzy logic has the advantage of creating the combination of rules in an intrinsic way, especially through the management of uncertainty: without having a different weight, a greater certainty has more impact.

Consider the example of the size of the interactor click areas, named here `clickCorrection`. It is based on the user's visual acuity (Fig. 7) and hand precision (whose graph shape is identical to Fig. 7). To define a more or less important correction level of the size, the output language variable - `clickCorrection` - is composed of four discrete values

representing four degrees of correction ranging from 'noCorrection' to 'highCorrection'. These rules in fuzzy logic are listed in Fig. 12.

```

If handPrecision is normal then clickCorrection is noCorrection
If handPrecision is low then clickCorrection is lowCorrection
If handPrecision is veryLow then clickCorrection is mediumCorrection
If handPrecision is extraLow then clickCorrection is highCorrection
If visualAcuity is normal then clickCorrection is noCorrection
If visualAcuity is low then clickCorrection is lowCorrection
If visualAcuity is veryLow then clickCorrection is mediumCorrection
If visualAcuity is extraLow then clickCorrection is highCorrection

```

Fig. 12: Fuzzy rules defined for calculating `clickCorrection` based on hand precision and visual acuity

The first four rules are based on the state of the `handPrecision` variable - 'normal', 'low', 'veryLow', 'extraLow' - and provide an output value for `clickCorrection` - from 'noCorrection' to 'highCorrection', respectively. The last four rules are the same except for the input variable, which is `visualAcuity`. It may be noted that it is not necessary in the expression of these rules to worry about their possible contradiction or their combination; in fuzzy logic, the output value is determined during the defuzzification.

In the example regarding the correction of the interactor click area size, to obtain a smooth transition from one value to the next, we have opted for the weighted centers of gravity defuzzification method (see subsection IV.B.2). Thus, with a visual acuity of 0 and a hand precision of 4 in input, the result is defined as 'lowCorrection'. Similarly, a visual acuity of 1 and a hand precision of 4 in input shows a 'mediumCorrection' output.

Fuzzy logic therefore manages the combination of adaptation rules thanks to the defuzzification without setting up a specific prioritization algorithm.

V. COMPARATIVE ANALYSIS

Fuzzy logic adaptation rules seem to have advantages over a classical Boolean logic approach. In order to test this hypothesis, we conducted a comparative analysis between the two types of rules, first on a theoretical point of view then with HCI experts' opinions gathered during interviews.

A. Analysis method

The comparative analysis of the two types of rules is based on the expression and the analysis of the results of rules coming both from the literature [11, 33, 34, 35, 36, 37], and our own experience as HCI designers (e.g. `clickCorrection`).

The rules were implemented in the two adaptation engines (in Boolean and fuzzy logics), and both integrated in the architecture described in section IV.A. The context of use was manually manipulated as a dependant variable in input to test the differences in the rules produced by the two engines.

In input, we use 14 *user characteristics* and 4 *other characteristics* to describe the environment. The user characteristics described the user's age, hands precision, visual acuity, language comprehension, abstract signs understanding, attention level, tremor, the possibility of dyslexia or color blindness ('protanopia', 'deutanopia', 'tritanopia' or 'total color blindness'), IT experience, and website experience. The four environmental inputs expressed the room's brightness, as well as the hardware used in terms of orientation and display surface (window's height and width). These values were assumed to come either through sensors or by the user's input. These entries were then linked through adaptation rules to 17 output parameters acting on

the interface: spacing between letters, words or lines, font and font size, text alignment, text quantity, the presence or absence of animations or purely decorative elements, interactors click size area, contrast, colors changes, using hypertext links or replacing them with buttons, display modes of menu (icon/text), ordering bulleted lists, guidance and confirmation levels. There were 34 adaptation rules in Boolean logic, and 81 in fuzzy logic. This difference was explained by the choice to implement the rules in fuzzy logic. In Fig. 12, the 8 rules in fuzzy logic can be grouped into 2 rules in Boolean logic in the *if... then... else if...* structures.

B. Rules

To illustrate these rules, we will use the example of a hotel website, illustrated in Fig. 13 (default homepage interface, with no adaptation).

Five groups of rules - for which the results differed between Boolean and fuzzy logic - were selected for this experiment. The rules mainly depended on multiple inputs.

1) Display Mode

The first group of rules was responsible for calculating the "display mode" variable, which corresponds to the use of icons, in addition to the text, within the menus (Fig. 14 and Fig. 15). The calculation was based on three characteristics: 'website experience', 'abstract signs understanding' and 'language reception' which respectively represent the level of user's knowledge about the current website, one's ability to understand symbols/icons, and one's ability to understand written language.



Fig. 14: Menu with text and icons



Fig. 15: Menu with text only

2) Verbose

The second group of rules was used to calculate the 'verbose' level of the website (i.e. the amount of text shown to the user). By default, the text was in its complete/standard version (Fig. 16), and it could be more or less summarized depending on the user's characteristics (Fig. 17). These rules

depend on two characteristics: 'Language reception' (ability to understand the written language, see V.B.1) and the user's level of 'attention' (i.e. concentration level).

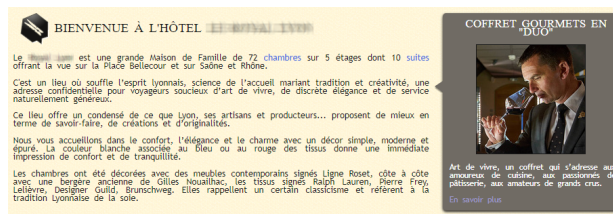


Fig. 16: Standard and complete presentation (default verbose)

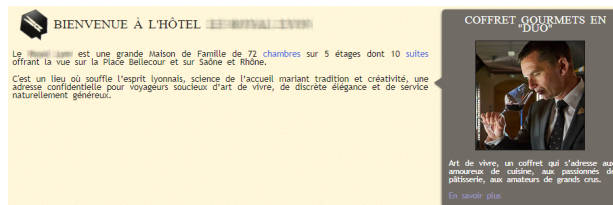


Fig. 17: Shortened presentation (concise verbose)

3) Guidance Level

The following group of rules calculated the 'Guidance level' to be implemented within the interface (standard or high) from two input characteristics: the 'IT experience' (with computers and web in general), and 'website experience' (with this particular website). Fig. 18 shows an example of a high guidance level in menus.

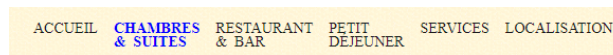


Fig. 18: Menu with high guidance level

4) Click correction

This fourth group was in charge of calculating a variable named 'Click correction', used to enlarge the clicks area of the interactors (Fig. 19; i.e. the size of the area around an interactor within which a click is associated to the interactor). This variable was calculated from the 'visual acuity' of the user, as well as from his 'hands precision' (movement of the mouse and clicks). This group of rules was used to avoid clicks made by mistake right next to an interactor.

TABLE 2: SYNTHESIS OF THE COMPARATIVE ANALYSIS RESULTS BETWEEN BOOLEAN AND FUZZY LOGICS

N°	Input situations	Results in boolean logic	Results in fuzzy logic
	Input variables	Values	
			1 websiteExperience
			abstractSignsUnderstanding
	languageReception 1/2/2	displayMode = textOnly	
	1/2/3	displayMode = textOnly	displayMode = textAndGraphics
	1/4/4	displayMode = textOnly	displayMode = textAndGraphics
	3/3/3	displayMode = textAndGraphics	
	2 languageReception		
	attention 1/1	verbose = thorough	
	1/2	verbose = thorough	verbose = summarized
	1/3	verbose = thorough	verbose = concise
	3/4	verbose = concise	verbose = laconic
	3 itExperience		
	websiteExperience 1/1	guidanceLevel = medium	
	1/3	guidanceLevel = medium	guidanceLevel = high
	3/3	guidanceLevel = high	
	4 visualAcuity		
	handPrecision 1/1	clickCorrection = noCorrection	



Fig. 13: Homepage of the hotel website

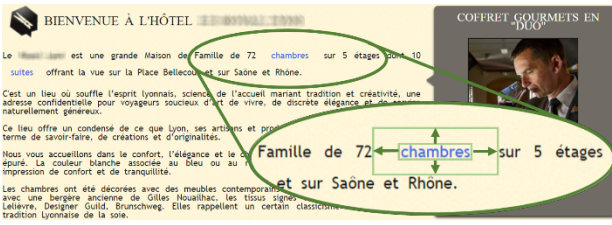


Fig. 19: Interactors click size area extended (High click correction)

5) Font size

The final rule regulated the increase of the font size, based on the visual acuity of the user.

C. Results

Table 2 summarizes the adaptation results of the rules presented in the section V.B. The table can be read as follows: the first column of the table (input situations) is subdivided into two sub-columns. The first sub-column details a set of several characteristics impacting the same output variable. The second sub-column defines examples of values corresponding respectively to each of the input variables. For example, the first input situation of the table is a user having a value of 1, 2 and 2 (1/2/2 in the table) for the listed input variables: websiteExperience (value of 1), abstractSignsUnderstanding (2), and languageReception (2). The next two columns specify the results in Boolean and fuzzy logics.

The differences of results in Boolean logic and in fuzzy logic, for the same input situation, make it possible to highlight two points:

a) For rules defined with a similar complexity between Boolean and fuzzy logic (i.e. rules written the same way in both logics), it seems to us that fuzzy logic makes it possible to obtain more relevant results with regard to the input situations, without having to define and use a system of weights between rules, as this is the case for Boolean logic. For example, the calculated value of the displayMode is identical in Boolean and fuzzy logic for the first and fourth situations, that is to say when the three input values are either all 'low' or all 'high'. However, when the input values provided are heterogeneous, the fuzzy logic returns a mean value, which is more precise. This is especially noticeable for the results of the second group of input situations (adaptation of verbose depending on languageReception and attention), where the verbose output variable can take four states ('thorough', 'summarized', 'concise' and 'laconic'). Indeed, when the value of one of the input variables varies from one extreme to the other, the results from fuzzy logic show that the final value is smoothed, passing through all the intermediate values, which is not necessarily the case in Boolean logic.

b) As this is the case with the font size adaptation according to visual acuity (last line of the table), the fuzzy logic makes it possible to easily obtain an extremely finer granularity than the Boolean logic. The latter could make it possible to achieve such granularity, but this would be to the detriment of the number of thresholds to be managed within the rule: our rules give only four possible font sizes (12, 14, 16 or 18) and we wanted more precise results we would need to pre-define all the values and include several conditions in the rules. Furthermore, it is important to note that in fuzzy logic, this granularity is not limited to the examples in the table: a visualAcuity value of 0.25 will have a fontSize output of 12.375, and so on.

This comparative analysis confirms that fuzzy logic makes it possible to obtain more consistent results with finer granularity without having to manage the combination of multiple rules.

D. Experts

1) Experimental setup

Our hypothesis is that fuzzy logic has the potential to create more relevant adaptations than Boolean logic for certain HCI adaptation rules. We tested this hypothesis by conducting interviews with HCI experts on this matter. These interviews also served as an idea generation workshop about rules improvement or fuzzy logic practical uses.

Our evaluation was conducted with eight experts, six men and two women, aged 37-57 years. They were all assistant professors or professors in HCI, working in France (Paris, Nice, Grenoble, Valenciennes) for more than ten years. The interviews lasted an average of 40 minutes.

The different input situations of the table were presented to the experts, as well as an example-based definition of the variables (to avoid misinterpretation). The example presented to the experts was the hotel website introduced in section V.B, along with screenshots of different adaptations pertaining to the rules. They were then asked to specify what they would recommend as an adaptation result for each input situation listed in Table 2, in the form of a list of choices corresponding to the possible values of the exit. Finally, they were asked to explain this choice during a free expression phase. Once the answers were given for the 5 proposed situations of Table 2, global feedback was gathered in the form of an open-discussion on the rules and logics tested.

The experts' opinions about the five groups of rules (see section V.B) are detailed below, with a summary in Table 3.

2) Group of rules n°1: DisplayMode

Six of the eight experts recommended the permanent use of icons in addition to the text in the menus (textAndGraphics), in contradiction to the recommendations made by MyUI [11, 35, 36], from which the rule was established. These recommendations propose rather to keep only the text when the user does not possess a major weakness for the various input variables (with an input value between 0 and 2) and to add the icons when he does (input values between 2 and 4). This result can hardly be interpreted from the point of view of the comparison between the two logics since it questions the rule itself. A seventh expert also disagreed with the rule at the level of the input variables, indicating that it would only take into account the "abstractSignsUnderstanding". The last expert agreed with the rule and his results join the fuzzy logic results.

3) Group of rules n°2: Verbose

Seven out of eight experts proposed results that

TABLE 3: SYNTHESIS OF THE EXPERTS AGREEMENTS FOR BOOLEAN AND FUZZY LOGICS

N°	Input variables	Output variable	Experts agreement with	boolean logic
1	websiteExperience abstractSignsUnderst.	displayMode	fuzzy logic	○
			boolean logic	○
2	languageReception attention	verbose	fuzzy logic	●
			boolean logic	○
3	websiteExperience	guidanceLevel	fuzzy logic	●
			boolean logic	●
4	handPrecision	clickCorrect.	fuzzy logic	●
			boolean logic	●

combined the two input variables to provide a more linear adaptation. The results here tend towards fuzzy logic, which allows more linearity and a simplified consideration of the

different input values. The last expert advised to rely only on the variable with the best value to perform the adaptation, in order to "rely on a strength".

4) *Group of rules n°3: GuidanceLevel*

Seven of the eight experts proposed a 'medium' guidance level for the first input situation (e.g. a user with both input characteristics at a relatively good level), and a 'high' guidance for the last one (e.g. a user with both input characteristics at a relatively low level). For the intermediate entry situation (e.g. a user with one good value and one bad for the two inputs characteristics), the results varied between 'medium' and 'high'. The eighth expert stated that "he would promote guidance in relation to the need" and therefore providing a high level of guidance in all situations. These results do not allow to put forward either the Boolean logic or the fuzzy logic, this group of rules having only two inputs and a two-state output, the differences remaining insignificant.

5) *Group of rules n°4: ClickCorrection*

Four of the experts stated that, in this case, they would use the maximum value between the two inputs to calculate the output, not an intermediate value calculated from the two inputs. In other words, this would result in a maximum weight for one of the inputs (the one with the best level), and zero weight for the second characteristic. This rather joins the operation of Boolean logic, although it remains possible to perform in fuzzy logic. The four remaining experts were inclined towards the use of a mean value between the two inputs, and therefore their results were more linear, closer to the fuzzy logic.

6) *Group of rules n°5: FontSize*

Six of the experts recommended "precise", "linear" font size calculations in order to avoid threshold effects, an inconvenient for users, which is consistent with fuzzy logic. The two remaining experts advocated the use of thresholds, as in classical Boolean logic, which also remains possible in fuzzy logic.

7) *Open discussion with the experts*

Two experts stated that the fuzzy logic way of handling uncertainty was particularly adapted for the values declared by users. Three experts also indicated that the rule combination management of the fuzzy logic was a good thing, although it may sometimes be necessary to maintain a weighting system for certain rules (e.g. a higher consideration of the websiteExperience than itExperience, when calculating the guidance level - see Group of rules n°3).

E. DISCUSSION

According to the results obtained during our evaluation, the fuzzy logic seems to answer the needs of a UI adaptation engine, while meeting our criteria: the uncertainty of values, the complexity of rules and their combination, without restricting the context of use. However, the experimental setup and the rules implementation are discussed further in this section.

1) *Threats to Validity*

The interviews were realized with a small number of experts. Nielsen & Mack [38] showed that eight general practitioner non-novices experts were enough to detect more than 90% of usability problems. But enlarging the panel to international experts could be interesting as it would provide a wider range in backgrounds. For instance, none of the experts tested was either an Ergonomist or an expert in fuzzy logic. It would be meaningful to interview experts specialized in Ergonomics or with a background in HCI and fuzzy logic to get more feedback during the final open

discussion. However, an expert evaluation cannot fully replace end-users' feedback. So conducting an end-user evaluation could complete our current work.

Another validity issue is related to the chosen example. As mentioned in section V.A, the input data used for the comparison analysis were manually selected by the authors to highlight the differences of the two logics, but the results were calculated solely by their respective engine. Other data could be considered in a similar experiment. Moreover, the rules were illustrated with the example of a hotel website. Other websites domain could be used to strengthen the results.

2) *Fuzzification, Defuzzification and Membership Function*

As presented in section III.D, fuzzy logic requires phases of fuzzification and defuzzification, as well as the definition of membership functions for each of the linguistic variable states. These phases are critical to obtain relevant results.

Fuzzification consists in associating a fuzzy value to an input value, according to the membership function of the linguistic variable term involved in the fuzzy proposition (e.g. the different membership functions for each term for visual acuity - Fig. 7 - or for the hand precision). The difficulty is that their definition remains subjective as the literature does not generally provides us with fixed values. However, it is possible to modify them after the fact.

In terms of defuzzification, several solutions are possible: first or last maximum, average maxima or a calculation of weighted center of gravity. The choice of the defuzzification method is defined for each output linguistic variable, and it will depend on both the type of the latter (continuous or discrete values) and the desired logic of the rule. In our work, the rules were all defined with a weighted centers of gravity, in order to calculate an average of the results for the rules that affect the same output variable instead of using only one of the results (which is the case with the maxima or minima defuzzification method). However, the calculation of the weighted centers of gravity is not perfect: for example, it was necessary to define the value domain of the font size (fontSize, Fig. 8) between 6 and 24 to obtain relevant results between 12 and 18 because of the way fuzzy logic and the membership functions work (see section IV.B.2). Nevertheless, it has always allowed us to obtain results that seem relevant to us.

Finally, the membership functions can be created in different manners: continuous piecewise functions, rectangular or triangular broken lines, Gaussian lines... Some types of curves reached beyond our research domain. For example, piecewise continuous functions would result in threshold effects at discontinuities and would not be the most suitable for representing users characteristics. Similarly, broken lines with rectangular shapes would mean that there is no management of the uncertainty and therefore fall back into an operation similar to the Boolean logic. In our work, we used triangular broken lines because, on one hand, they allow simpler calculations, and on the other hand, they are easier to be apprehended by experts. Gaussians can also be used, but are preferred when rules are learned automatically.

3) *Types of Variables Used*

In this work, we used quantitative variables (e.g. visualAcuity), ordinal variables (e.g. clickCorrection) and nominal variables (e.g. types of color blindness). Compared to Boolean logic, the contributions of fuzzy logic in this paper were about quantitative or ordinal variables. The nominal variables are used in the same way in the two logics, without contribution from fuzzy logic. There may be an

interest in using fuzzy logic for this last type of variable, but no example has been found yet.

4) Rules Relevance

According to our study, fuzzy logic seems adequate to support a UI adaptation engine. Nevertheless, there is always the question of the relevance and coverage of the rules we have expressed. These may be considered limited or arbitrarily chosen. To overcome these shortcomings, we have used rules mostly derived from the HCI adaptation literature [11, 33, 34, 35, 36, 37]. There were three rules in Boolean logic (six in fuzzy logic) relative to the devices (window's size and orientation), one rule (four in fuzzy logic) relative to the environment (brightness), and 30 rules (71 in logic fuzzy) related to the user. As far as the user is concerned, 23 of the 30 rules come from the literature [11, 33, 34, 35, 36, 37], including 11 rules taken from the European project MyUI [11, 39, 40].

We faced an issue regarding rule validation by the experts: Some did not agree with the rules themselves (most notably the group of rules n°1, for the displayMode output). So the problem was more related to a lack of consensus about rules in the HCI community than the rule itself.

VI. CONCLUSION

This article proposes a UI adaptation system based on a fuzzy logic adaptation engine. We showed the potential interest of using fuzzy logic in terms of uncertainty of the input values and combination of adaptation rules. The fuzzy logic and the Boolean logic solutions were compared first, theoretically, then through eight experts' interviews. This comparison has shown the advantages of fuzzy logic.

We can also note that the greater the number of characteristics and rules, the more useful fuzzy logic can be compared to Boolean logic. This was explained by fuzzy logic bringing advantages when an output variable depends on several inputs (see section IV.B.3). Should advances be made in the characterization of uncertainty, adaptation using fuzzy logic would benefit from them.

The question of the difficulty to handle the specific aspects of fuzzy logic remains, that is, fuzzification and defuzzification. Experiments with computer engineers could prove this point to be immaterial.

Finally, it is possible to consider an extension of the adaptation engine, which is not limited to logic but also proposes automatic learning techniques for the personal preferences of users such as those related to aesthetics.

REFERENCES

[1] D. Thevenin, J. Coutaz and G. Calvary, «A Reference Framework for the Development of Plastic User Interfaces.» in Multi-Device and Multi-Context User Interfaces: Engineering and Applications Frameworks, 2003.

[2] J.-S. Sottet, V. Ganneau, G. Calvary, J. Coutaz, A. Demeure, J.-M. Favre and R. Demumieux, «Model-Driven Adaptation for Plastic User Interfaces.» in Proc. INTERACT 2007, the eleventh IFIP TC13 International Conference on Human-Computer Interaction, 2007.

[3] M. Blumendorf, S. Feuerstack and S. Albayrak, «Multimodal User Interfaces for Smart Environments: The Multi-Access Service Platform.» in Proceedings of the working conference on Advanced visual interfaces, Napoli, 2008, pp. 478-479.

[4] C. Chesta, F. Paternò and C. Santoro, «Methods and Tools for Designing and Developing Usable Multi-Platform Interactive Applications.» PsychNology, vol. 2, n°1, pp. 123-139, 2004.

[5] M. Florins, Graceful Degradation: a Method for Designing Multiplatform Graphical User Interfaces, Louvain, 2006.

[6] T.-D. Nguyen, J. Vanderdonck and A. Seffah, «UIPLML: A Pattern Tool for Engineering Multi-Platforms Information Systems.» in Tenth International Conference on Research Challenges in Information Science (RCIS), 2016.

[7] R. Miñón, F. Paternò, M. Arrue and J. Abascal, «Integrating adaptation rules for people with special needs in model-based UI development process.» Universal Access in the Information Society, vol. 15, n°1, pp. 153-168, 2016.

[8] E. Yigitbas, S. Sauer and G. Engels, «Adapt-UI: an IDE supporting model-driven development of self-adaptive UIs.» in Proceedings of the ACM SIGCHI Symposium on Engineering Interactive Computing Systems, Lisbon, 2017.

[9] S. Abrahão, F. Bordeleau, B. Cheng, S. Kokaly, R. F. Paige, H. Störrle and J. Whittle, «User Experience for Model-Driven Engineering: Challenges and Future Directions.» in 2017 ACM/IEEE 20th International Conference on Model Driven Engineering Languages and Systems (MODELS), 2017.

[10] «Virtual User Modeling and Simulation Cluster.» 2013. [Online]. Available: http://vums.iti.gr/indexb60b.html?page_id=44.

[11] M. Peissner, D. Häbe, D. Janssen and T. Sellner, «MyUI: generating accessible user interfaces from multimodal design patterns.» in Proceedings of the 4th ACM SIGCHI symposium on Engineering interactive computing systems, New York, 2012.

[12] F. Paternò, C. Santoro and L. D. Spano, «MARIA: A universal, declarative, multiple abstraction-level language for service-oriented applications in ubiquitous environments.» ACM Transactions on Computer-Human Interaction, vol. 16, n°4, 2009.

[13] P. Akiki, «Engineering Adaptive Model-Driven User Interfaces for Enterprise Applications.» 2016.

[14] G. Lehmann, A. Rieger, M. Blumendorf and S. Albayrak, «A 3-layer architecture for smart environment models.» in Pervasive Computing and Communications Workshops, Mannheim, 2010.

[15] B. Collignon, J. Vanderdonck and G. Calvary, «Model-Driven Engineering of Multi-target Plastic User Interfaces.» in Proceedings of the Fourth International Conference on Autonomic and Autonomous Systems, Washington DC, 2008.

[16] D. Thevenin, Adaptation in Human Computer Interaction: the case of Plasticity, Grenoble, 2001.

[17] V. López-Jaquero, J. Vanderdonck and P. González, «Towards an Extended Model of User Interface Adaptation: The Isatine Framework.» in Engineering Interactive Systems, 2007, pp. pp.374-392.

[18] A. Blouin, B. Morin, O. Beaudoux, G. Nain, P. Albers and J.-M. Jézéquel, «Combining aspect-oriented modeling with property-based reasoning to improve user interface adaptation.» in Proceedings of the 3rd ACM SIGCHI symposium on Engineering interactive computing systems, New York, 2011.

[19] V. Genaro Motti, N. Mezhoudi and J. Vanderdonck, «Machine Learning in the Support of Context-Aware Adaptation.» in 10th International Conference on Mobile Ubiquitous Computing, Systems, Services and Technologies, 2012.

[20] T. Giuffrida, S. Dupuy-Chessa, J.-P. Poli and E. Cérêt, «Fuzzy4U : un système en logique floue pour l'adaptation des interfaces utilisateur.» in 30eme conférence francophone sur l'interaction homme-machine, Brest, 2018.

[21] K. Z. Gajos, D. S. Weld and J. O. Wobbrock, «Automatically generating personalized user interfaces with Supple.» Artificial Intelligence, vol. 174, n°12-13, pp. 910-950, 2010.

[22] J. Cao, N. Xing, A. Chan, Y. Feng and B. Jin, «Service adaptation using fuzzy theory in context-aware mobile computing middleware.» in Proceedings of the 11th IEEE International Conference on Embedded and Real-Time Computing Systems and Applications, 2005.

[23] H. Desruelle, D. Blomme and F. Gielen, «Adaptive Mobile Web Applications: A Quantitative Evaluation Approach.» in International Conference on Web Engineering, 2011.

[24] G. Cueva-Fernandez, J. P. Espada, V. García-Díaz, R. G. Crespo and N. Garcia-Fernandez, «Fuzzy System to Adapt Web Voice Interfaces Dynamically in a Vehicle Sensor Tracking Application Definition.» Soft Computing, vol. 20, n°8, pp. 3321-3334, 2016.

[25] H. O. Nyongesa, T. Shicheng, S. Maleki-Dizaji, S. T. Huang and J. Siddiqi, «Adaptive Web interface design using fuzzy logic.» Proceedings IEEE/WIC International Conference on Web Intelligence (WI 2003), pp. pp. 671-674, 2003.

[26] E. Papatheocharous, M. Belk, P. Germanakos and G. Samaras, «Proposing a Fuzzy Adaptation Mechanism Based on Cognitive Factors of Users for Web Personalization.» Artificial Intelligence Applications and Innovations, p. 135-144, 2012.

[27] R. Riding and I. Cheema, «Cognitive Styles — an overview and integration.» Educational Psychology, vol. 11, pp. 193-215, 1991.

[28] M. Soui, M. Abed and K. Ghedira, «Fuzzy Logic Approach for Adaptive Systems Design.» Human-Computer Interaction. Towards Intelligent and Implicit Interaction, pp. 141-150, 2013.

[29] L. Zadeh, «Fuzzy sets.» Information and Control, vol. 8, n°3, pp. 338-353, 1965.

[30] Zadeh, L., «The concept of a linguistic variable and its application to approximate.» Information Sciences, vol. 9, pp. 43-80, 1975.

- [31] B. Bouchon-Meunier, *La logique floue*, puf, 2007.
- [32] E. Céret, S. Dupuy-Chessa, G. Calvary and M. Bittar, «System and method for magnetic adaptation of a user interface». France INPI Brevet TPI2015053, 07 07 2015, second deposit 07 07 2016.
- [33] G. Moss, *Gender, Design and Marketing: How Gender Drives our Perception of Design and Marketing*, Gower, 2009.
- [34] L. Rello and R. Baeza-Yates, «Good fonts for dyslexia,» in *Proceedings of the 15th International ACM SIGACCESS Conference on Computers and Accessibility*, 2013.
- [35] C. Stephanidis, A. Paramythis, M. Sfyarakis, A. Stergiou, N. Maou, A. Leventis, G. Paparoulis and C. Karagiannidis, «Adaptable and adaptive user interfaces for disabled users in the AVANTI project,» in *Proceedings of the 5th International Conference on Intelligence and Services in Networks: Technology for Ubiquitous Telecom Services*, 2006.
- [36] J.-P. Leuteritz, H. Widroither, A. Mourouzis, M. Panou, M. Antona and A. Leonidis, «Development of Open Platform Based Adaptive HCI Concepts for Elderly Users,» in *Lecture Notes in Computer Science*, vol. 5615, 2009.
- [37] C. Denis and L. Karsenty, «Inter-Usability of Multi-Device Systems – A Conceptual Framework,» in *Multiple User Interfaces: Cross-Platform Applications and Context-Aware Interfaces*, 2005, pp. 373 - 385.
- [38] J. Nielsen and R. L. Mack, *Usability inspection methods*, New York: John Wiley & Sons, 1994.
- [39] «MyUI project Official Web Page,» [Online]. Available: <http://www.myui.eu/>.
- [40] «MyUI Pattern Browser,» 2013. [Online]. Available: http://myuipatterns.clevercherry.com/index.php?title=Main_Page.