# Enhancing Flexibility and Expressivity of Contextual Hierarchies

Yoann Pitarch
CS Department
Aalborg University, Denmark
Email: ypitarch@cs.aau.dk

Cécile Favre
ERIC
Lyon, France
Email: cecile.favre@univ-lyon2.fr

Anne Laurent and Pascal Poncelet
LIRMM
Montpellier, France
Email: {laurent,poncelet}@lirmm.fr

***Abstract*—Data warehouses are nowadays extensively used to perform analyses on huge volume of data. This success is partly due to the capacity of considering data at several granularity levels thanks to the use of hierarchies. However, in previous work, we showed that the experts' knowledge were not much considered in the generalization process. To overcome this drawback, we introduced a new category of hierarchies, namely the contextual hierarchies. Unfortunately, in contrast to the complexity of expert knowledge that should be considered, the knowledge definition process was too rigid. In this paper, we extend these hierarchies and their related techniques to drastically increase their flexibility and expressivity. To this purpose, we adopt a fuzzy-based methodology which allows to express expert knowledge in a very convenient way. Experiment results obtained on synthetic datasets show that the contextual generalization process is very fast and can thus be used in practice.**

## I. Introduction

Today the volume of generated data rapidly increases and their analysis represents a strategic challenge for decision makers. To this aim, data warehouses were designed to consolidate, store and organize data thanks to a multidimensional schema [1]. Often, attributes are equipped with hierarchies which allow to consider data at several levels of granularity [2]. On-Line Analytical Processing (OLAP) tools exploit these hierarchies by enabling the decision makers to aggregate and examine data at different combinations of levels [3]. Because of the effectiveness of these tools, data warehouses play a central role in various decisional information systems [4] such as sales analysis, equipment monitoring or medical data surveillance [5].

In this paper, we illustrate our purpose using this last domain since this work was partially supported by a french national project[1] that brings into play an Intensive Care Unit (ICU). Thus, let us assume that a medical data warehouse records vital signs, *e.g.*, blood pressure, of patients from an ICU. In order to achieve effective monitoring of patients, a doctor may want to know those who had a low blood pressure during the night. Formulating this type of query requires the existence of a hierarchy on blood pressure where the first level of aggregation is a categorization of blood pressure, *e.g.*, low, normal, high. However, this categorization is tricky because it strongly depends on both blood pressure measured and certain physiological characteristics of the considered patient, *e.g.*, the age, the average daily tobacco consumption. Therefore, the same pressure can be differently generalized according to the considered analysis context. For example, 13 is a high pressure for babies whereas it is a normal one for adults. To allow these generalizations, we proposed a new type of hierarchy, namely the *contextual hierarchies* [6].

The definition of these hierarchies makes possible to express and integrate some crisp and very precise expert knowledge into the navigation and analysis processes. However, it now appears that we need to go one step beyond to enhance expressivity and flexibility of the expert knowledge. Precisely, it may be very interesting to consider these two points. First, knowledge can be more or less precise. Second, the adequacy of a piece of knowledge to a patient is sorely crisp by nature.

In terms of expressiveness of knowledge, the fuzzy approach is really pertinent [7]. Various studies have also involved the integration of fuzzy logic in data warehouses to increase the expressivity of hierarchies [8], [9]. Thus, in this work, we propose to focus on the integration of fuzzy logic in the context of hierarchies, going a step further in the expressiveness hierarchies, ensuring analyses even more relevant.

The remainder of this paper is as follows. Section II presents a case study. Section III introduces some background to this work. In Section IV, we discuss the existing model and exhibit improvements to bring. In Section V, we propose our new extended model of contextual hierarchies. Implementation details are presented in Section VI. Experimental results are presented in Section VII. We briefly discuss the related work in Section VIII and Section IX concludes this paper.

## II. Case Study

Assume that we would like to determine the normality degree of the blood pressure (BP) measured on patient 1 (P1). Informations related to this patient are displayed in Table I. It can be noted that the attributes *Age* and *Tobacco Consumption* can each one be considered at two levels of granularity. Figure 1 displays the concerned samples of these two hierarchies. As stated in our previous work [6], BP generalization is impacted by some patient-related features. Table II displays some of the expert knowledge that could be useful to perform this generalization.

Based on the joint analysis of Tables I and II, the following remarks could be observed:

[1]Project ANR MIDAS (ANR-07-MDCO-008)

| IdP | Age | AgeSubCategory (ASC) | Tobacco Consumption (TC) | Tobacco Consumption Category (TCC) | Treatment | Blood Pressure (BP) | Blood Pressure Category (BPC) |
|---|---|---|---|---|---|---|---|
| P1 | 22 | Adult | 5 | Occasional | Hypotensive | 13 | ? |

TABLE I
TUPLE OF $T$ RELATED TO THE PATIENT 1

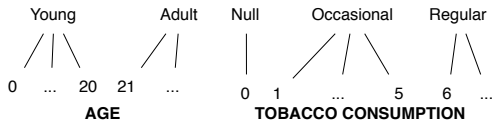



Fig. 1. Hierarchies over X and Y attributes

TABLE II
EXPERT KNOWLEDGE ASSOCIATED TO THE BP GENERALIZATION

| IdC | Connaissance |
|---|---|
| R1 | Young + Null or occasional + BP $\in [8;12] \rightarrow$ Normal |
| R2 | Young + Null or occasional + $BP > 12 \rightarrow$ High |
| R3 | Regular + $BP \in [9;14] \rightarrow$ Normal |
| R4 | Regular + $BP > 14 \rightarrow$ High |
| R5 | Hypotensive + $BP > 12 \rightarrow$ High |
| R6 | $BP \in [8;13] \rightarrow$ Normal |
| ... | ... |

1) Pieces of knowledge do not necessarily concern the same attributes, *e.g.*, $R4$ and $R5$.
2) Both $R5$ and $R6$ can be applied but their generalization differ.
3) $R5$ is more precise than $R6$ since more conditions have to be fulfilled.
4) $R2$ and $R3$ almost match with $P1$. Typically, one more smoked cigaret per day would have been enough to consider that $R3$ matches with $P1$.

In the two next sections, we recall the existing contextual hierarchy model and show that it does not fully integrate these observations. Thus, the model is extended and efficient solutions are proposed to handle these extended hierarchies.

## III. BACKGROUND

We first recall definitions related to *classical* contextual hierarchies and then described existing solution to handle and exploit them. Finally, we recall definitions related to fuzzy hierarchies.

### A. Classical Contextual Hierarchies

**Definitions.** Contextual hierarchies were first designed in the framework of data warehouses [10]. Consequently, the initial formalism redefines the standard concepts of *dimension*, *fact table*, *dimension attribute* and *measure*. In this paper, to lighten the definitions and without loss of generality, the formalism only assumes a functional dependency between an identifier attribute and a set of attributes.

More formally, let $Id = \{id_1, \ldots, id_n\}$ be the identifier attribute and $\mathcal{A} = \{A_1, \ldots, A_t\}$ be a set of $t$ attributes. For each attribute $A_i \in \mathcal{A}$, its domain of definition is denoted by $Dom(A_i)$. We assume that each attribute $A_i$ is equipped with a hierarchy and can thus be observed at several granularity levels $A_i = A_i^0, \ldots, A_i^{max_i}$. Such hierarchies are said to be *simple*, *i.e.*, the several levels of granularity form a chain. By convention, $A_i^0$ stands for the finest granularity level and $A_i^{max_i}$ stands for the coarsest granularity level. We denote $a_i^j \in Dom(A_i^j)$ if the attribute value $a_i^j$ belongs to the level $A_i^j$. The set of the granularity levels from all the attributes is denoted by $\mathcal{A}^* = \{A_1^0, A_1^1, \ldots, A_t^{M_t}\}$. We assume that data are stored in a single table $T$.

*Definition 1:* (Relational Table) Let $Id$ be an identifier attribute and $\mathcal{A}^*$ be the set of granularity levels of the descriptive attributes. Data are stored in a relational table $T = (Id, A_1^0, A_1^1, \ldots, A_t^{M_t})$ so that $Id \rightarrow A_1^0 \times A_1^1 \times \ldots \times, A_t^{M_t}$. Each tuple $t$ from $T$ is of the form $t = (id, a_1^0, a_1^1, \ldots, a_t^0, \ldots, a_t^{M_t})$ where (1) $a_t^0$ is the value to contextually generalize and (2) $a_t^1, \ldots, a_t^{M_t}$ are the contextualized generalizations.

*Example 1:* Considering our case study, Table I displays how data related to $P1$ are stored in $T$.

Given an attribute $A_i$, the generalization from a level to its direct upper level is guided by the so called *aggregation path*.

*Definition 2:* (Aggregation Path) Let $A_i^j$ and $A_i^{j+1}$, $j = 0, \ldots, M_i - 1$ be two granularity levels from the same attribute $A_i$. The aggregation path between $A_i^j$ and $A_i^{j+1}$, denoted $G = A_i^j \overset{\mathrm{C}}{\mapsto} A_i^{j+1}$ with $\mathrm{C} = \{A_i^j, \ldots\}$, is such that:

1) $A_i^j$ is called the *source attribute*,
2) $A_i^{j+1} \notin \mathrm{C}$ is the *result attribute*,
3) $\mathrm{C} \subseteq \mathcal{A}$ is the set of attributes that may affect the generalization,
4) $A_i^{j+1}$ functionally depends on C.

Traditionally, a generalization value only depends on its specialization. For instance, generalizing *Sydney* to the *Country* level does not necessitate external knowledge. Though, as motivated in Section II, some generalizations are not so trivial and require external knowledge to be correctly performed. In the following we make the distinction between these two cases.

*Definition 3:* (Classical and Contextual Aggregation Path) Let $G = A_i^j \overset{\mathrm{C}}{\mapsto} A_i^{j+1}$ be an aggregation path. $G$ is said to be *contextual* if $|\mathrm{C}| > 1$. Otherwise, $G$ is said to be a *classical aggregation path*.

*Definition 4:* (Contextualizing and Contextualized Attributes) Let $G = A_i^j \overset{\mathrm{C}}{\mapsto} A_i^{j+1}$ be a contextual aggregation path. The attribute is said to be *contextualized* by the attributes

of C that are said to be *contextualizing*.

*Definition 5:* (Context) Let $G = A_i^j \overset{\mathrm{C}}{\mapsto} A_i^{j+1}$ be a contextual aggregation path. The triplet $c = (A_i^j, \mathrm{C}, A_i^{j+1})$ is called the *context* of the attribute $A_i^j$.

*Definition 6:* (Instance of aggregation path) Let $G = A_i^j \overset{\mathrm{C}}{\mapsto} A_i^{j+1}$ be an aggregation path. An instance of $G$,denoted $g = a \overset{\mathrm{IC}}{\longmapsto} a'$, is such that:

1) $a \in Dom(A_i^j)$ and $a' \in Dom(A_i^{j+1})$,
2) $\mathrm{IC} = \{(A_l^m, \alpha) \ — \ A_l^m \in \mathrm{C}$ and $\alpha \subseteq Dom(A_l^m)\}$,
3) $\exists (A_l^m, \alpha)$ so that $A_i^j = A_l^m$ and $a \in \alpha$,
4) $\nexists A_l^m$ so that $A_l^m \in \mathrm{C}$ and $(A_l^m, \alpha) \notin \mathrm{IC}$.

*Definition 7:* (Instance of context) Let $g = a \overset{\mathrm{IC}}{\longmapsto} a'$ be an instance of the aggregation path $A_i^j \overset{\mathrm{C}}{\mapsto} A_i^{j+1}$. The pair $c^l = (\mathrm{IC}, a')$ is called an *instance of the context* $c$ of $A_i^{j+1}$.

*Example 2:* Assume that the generalization from the level *Blood Pressure* (*BP* for short) to the level *Blood Pressure Category* (*BPC* for short) is contextualized by attributes *Tobacco Consumption Category* (*TCC* for short) and *BP* only. The contextual aggregation path is thus denoted $G = BP \overset{\mathrm{C}}{\mapsto} BPC$ where $\mathrm{C} = \{TCC, BP\}$ and the associated context is denoted $c = (BP, \mathrm{C}, BPC)$. Here, *TCC* and *BP* are the contextualizing attributes and *BPC* is the contextualized attribute. The instance of context related to $R4$ is represented by $c^l = (\mathrm{IC}, \mathrm{High})$ where $\mathrm{IC} = \{(TCC, \{Regular\}), (BP, \{x > 14\})\}$.

**Knowledge Storage.** In previous work [6], we proposed to store the expert knowledge in an external relational database denoted by $\mathcal{DB}_{Know.}$. This solution allows to store different contexts in a similar way and thus guarantees the genericity of the knowledge storage within the data warehouse. This external database is composed of only two relations which are briefly described below.

The Knowledge Meta Table, denoted `KMT`, stores the structure of the different contexts of $T$. This table has the following structure:

- *IdCtxt* designates the context identifier. By convention, we consider that it is the contextualized attribute, *i.e.*, *BPC* in our case study.
- *Attribute* designates an attribute involved in *IdCtxt*.
- *Type* indicates if *Attribute* is contextualizing or contextualized. By convention, *Type* = 0 if *Attribute* is contextualizing and *Type* = 1 if *Attribute* is contextualized.

The Knowledge Table, denoted `KT`, stores the instances of the different contexts in the database. This table has the following structure:

- *IdCtxt* designates the related context.
- *IdInstance* is an integer identifying the instance of context.
- *Attribute* designates an attribute involved in the context.
- *Value* represents the set of values of *Attribute* that are concerned in this instance of context.

**Contextual Roll-Up.** Due to a lack of space, the functioning of `ROLL_UP_CTX`, the contextual roll-up operator, is shortly described below[2]. First, the concerned context has to be identified in `KMT` to exhibit the contextualizing attributes. `KT` is then queried to determine the unique instance that is concerned by this generalization. Once the instance of context is identified, the value of the contextualized attribute associated to this instance has just to be returned.

### B. Fuzzy Hierarchies

When combined to decision systems, fuzzy set theory [11] allows to improve performances and expressivity [9]. Particularly, fuzzy hierarchies described in [12] allow to model the fact that an element can belong to several generalizations with each distinct membership degree. In our context, using this kind of hierarchies can be extremely useful to model instances of context that almost match a given tuple.

More formally, given a set of references, denoted $Dom(A_i^j)$, where $j = 1, \ldots, M_i$, a fuzzy subset $A$ of $Dom(A_i^j)$ is defined by a membership function $f_A$. This function associates to each element $a$ in $Dom(A_i^0)$ a membership degree $f_A(a)$, $0 \leq f_A(a) \leq 1$, which indicates how much $a$ is in $A$. For instance, $f_{ASC=Adult}(22) = 0.7$ means that 22 belongs to the subset $\{Adult\}$ with a degree of 0.7.

## IV. Discussion and Objectives

Existing contextual hierarchies allow the integration of expert knowledge into the generalization process. Hence, they represent an original and very convenient solution to numerous real case scenarios, *e.g.*, medical data warehouse or alarm detection, where numerical values have to be aggregated into a semantically correct categorical value. Though, these hierarchies still suffer from some limitations. Mainly, it must exist one (referred as the *existence constraint*) and only one (referred as the *unicity constraint*) instance for a given contextual generalization.

The *existence constraint* implies that the set of expert knowledge is complete regarding the instances of $T$. Without this guarantee, a blood pressure value may not be generalized. This theoretical guarantee is of crucial importance but is difficult to check in practice. Indeed, tools for checking the completeness of a set of rules exist but fail when dealing with large number of attributes [13]. This point is a problem since we do not have any control on the number of contextualizing attributes in a given context.

The *unicity constraint* guarantees the consistency of the system but is extremely difficult to obtain in practice. Several arguments motivate this assumption. First, contextualizing attributes represent the factors that could impact on a generalization. Nevertheless, it is not straightforward that the combined impact of all the contextualizing attributes has been studied. Typically, considering our case study, there is no expert knowledge specifying values on the complete set of contextualizing attributes. Second, determining if a tuple

[2]Interested readers may refer to [6] for further details.

matches or not with an instance of context is sometimes tricky. Typically, in our case study, we have seen that $P1$ almost but not fully matches the instance $R3$. As we will see in the next section, the use of fuzzy hierarchies could overcome this inflexibility.

Relaxing both *existence* and *unicity* constraints is thus of high importance to enhance the expressivity of our model but it comes with new problematics mainly related to the knowledge storage and the generalization process.

## V. Extended Contextual Hierarchies

We now overcome the above described limitations by extending our model in two ways. First, *generalized instances of context* are introduced. These instances allow the description of pieces of knowledge which do not involve all the contextualizing attributes. Second, contextualizing attributes can be now equipped with *fuzzy hierarchies*.

### A. Generalized Instances of Context

As previously argued, guarantying the completeness in the sense of the tuples of $T$ can be tricky. Additionally, this completeness can be ensured at time $t$ but cannot be ensured at time $t+1$ if new tuples are inserted in the table. Thus, guarantying the universal completeness of the system, *i.e.*, any potential tuple of $T$ can match with an instance of context, would be preferable. For this, we authorize instances of context to be more or less precise. Typically, very general pieces of knowledge such as $R6$ could ensure that any blood pressure could be generalized. This knowledge could be completed by much more precise instances such as $R1$ and $R2$. Naturally, during the generalization process, the most precise instance would be chosen among the set of adequate instances. *Generalized instances of context* are now formally defined and we introduce a generality-based relation to partially order instances.

*Definition 8:* (Generalized instance of a contextual aggregation path) Let $G = A_i^j \overset{\mathrm{C}}{\mapsto} A_i^{j+1}$ be a contextual aggregation path. A generalized instance of $G$ , denoted $g = a \overset{\mathrm{IC}}{\longmapsto} a'$, is such that:

1) $a \in Dom(A_i^j)$ and $a' \in Dom(A_i^{j+1})$
2) $\mathrm{IC} = \{(A_l^m, \alpha) \mid A_l^m \in \mathrm{C} \wedge \alpha \subseteq Dom(A_l^m)\}$
3) $\exists (A_l^m, \alpha) | A_i^j = A_l^m \wedge a \in \alpha$

*Definition 9:* (Generalized instance of context) Let $g = a \overset{\mathrm{IC}}{\longmapsto} a'$ be a generalized instance of the aggregation path $A_i^j \overset{\mathrm{C}}{\mapsto} A_i^{j+1}$. The pair $c^l = (\mathrm{IC}, a')$ is named a *generalized instance* of the $c$ of $A_i^{j+1}$.

The contextualizing attributes of a given instance $g$ are denoted $Attrib(g)$. We now introduce the $Prec$ function to formally quantify the precision of a given generalized instance of context.

*Definition 10:* (Precision of a generalized instance of context) Let $c^l = (\mathrm{IC}, a)$ be a generalized instance of the context $c = (A_i^j, \mathrm{C}, A_i^{j+1})$. The precision of $c^l$, denoted $Prec(c^l)$, is defined as $Prec(c^l) = |\mathrm{IC}|$.

*Example 3:* $c^l = \big(\{(\mathrm{BP}, [8; 13])\}, \mathrm{Normal}\big)$ is the generalized instance of the context $c = (BP, \{BP, ASC, TCC, Treatment\}, BPC)$ which represents $R6$. Additionally, $Attrib(c^l) = \{BP\}$ and $Prec(c^l) = 1$.

In the initial model, the precision of any instance was the number of contextualizing attributes. From now on, instances could be *more general* than others.

### B. Taking Fuzzy Hierarchies into Account

The use of fuzzy hierarchies leads to redefine the table $T$ to store membership degrees. From now on, a tuple $t = (id, a_1^0, a_1^1, \ldots, a_t^0, \ldots, a_t^{M_t})$ from $T$ is stored on the form $t' = (id', e_1^0, e_1^1, \ldots, e_t^0, \ldots, e_t^{M_t})$ such that $id = id'$, $e_i^0 = a_i^0$ for $i = 1, \ldots, t$ and $e_i^j = \{(\alpha_i^j, f_{\alpha_i^j}(e_i^0)) \mid \alpha_i^j \in Dom(A_i^j)\ and\ f_{\alpha_i^j}(e_i^0) \neq 0\}$ for $i = 1, \ldots, t-1$ and $j = 1, \ldots, M_i$. The Table III illustrates how fuzzy hierarchies over the *Age* and *Tobacco Consumption* attributes are stored.

Taking fuzzy hierarchies into account automatically breaks the unicity constraint. Indeed, a fuzzy tuple from $T$ can match to several instances of context with distinct adequacy degrees. We now focus on how this adequacy degree can be computed. Intuitively, the process is very similar to the crisp case: the adequacy of a tuple to each condition has to be computed and these local adequacies have then to be combined.

We first describe how the local adequacy, *i.e.*, the adequacy of a tuple to a condition, is computed. For this, let us assume that we want to compute how much the patient $P1$ matches to the criterion $(ASC, \{Young, Adult\})$. Here, it comes to compute the membership degree of the element 22 to the fuzzy subset $\{Young, Adult\}$, *i.e.*, to the union of fuzzy subsets $\{Young\}$ and $\{Adult\}$. Since the t-conorm operator, denoted $\perp$, is the fuzzy extension of the union operator, it is used to define such local adequacy.

*Definition 11:* (Local adequacy) Let $cond = (A_i^j, \alpha)$ be a condition (with $j = 0, \ldots, M_i$ and $\alpha = \{\alpha_1, \ldots, \alpha_k\}$) and $t' = (id', e_1^0, e_1^1, \ldots, e_t^0, \ldots, e_t^{M_t})$ be a tuple from $T$. The adequacy of $t'$ to $cond$, denoted $\mu_{unit}(t', cond)$, is defined as:

$$\mu_{unit}(t', cond) = \perp\Big(f_{\alpha_1}(e_i^0), \ldots, f_{\alpha_k}(e_i^0)\Big)$$

Many t-conorm functions exist in the literature. A detailed study could be useful to determine which one is the most appropriate. Nevertheless, this study is out of the scope of this paper and we consider here the $max$ function as the t-conorm function as introduced by Zadeh.

*Example 4:* Let $t$ be the tuple corresponding to the patient $P1$ and $cond = (\mathrm{ASC}, \{\mathrm{Young, Adult}\})$ be a criterion. We have $\mu_{unit}(t, cond) = max(f_{ASC=Young}(22), f_{ASC=Adult}(22)) = 0.7$.

We now describe how the local adequacies can be combined to compute the global adequacy of a tuple to an instance of context. An instance can be typically seen as a conjunction of conditions. Thus, the global adequacy measures the adequacy to *all* the criteria involved in a given instance. In other words, it measures the membership degree of the intersection of fuzzy

| IdP | Age | ASC | TC | TCC | Treatment | BP | BPC |
|---|---|---|---|---|---|---|---|
| P1 | 22 | $f_{ASC=Young}(22) = 0.25$<br>$f_{ASC=Adult}(22) = 0.7$ | 5 | $f_{TCC=occ.}(5) = 0.3$<br>$f_{TCC=reg.}(5) = 0.6$ | Hypotensive | 13 | ? |

TABLE III
INTEGRATION OF FUZZY HIERARCHIES IN THE CASE STUDY

subsets that each represents a criterion involved in the instance. Since the t-norm operator, denoted $\top$, is the fuzzy extension of the intersection operator, it is used to define such global adequacy.

*Definition 12:* (Instance adequacy to a tuple) Let $c^l = (\text{IC}, a)$ be an instance of the context $c = (A_i^j, \text{C}, A_i^{j+1})$ and $t' = (Id, e_1^0, e_1^1, \ldots, e_t^0, \ldots, e_t^{M_t})$ be a tuple from $T$. The adequacy of $t'$ to $c^l$, denoted $\mu(t', c^l)$, is defined as:

$$\mu(t', c^l) = \top\Big(\mu_{unit}\big(t', (A_k^m, \alpha)\big) \text{ such that } A_k^m \in Attrib(c^l)\Big)$$

Many t-norm functions exist in the literature. Similarly to the t-conorm function, a detailed study could be useful to determine which one is the most appropriate. Nevertheless, this study is out of the scope of this paper and we consider here the $min$ function as the t-norm function as introduced by Zadeh.

*Example 5:* Let $t$ be the tuple corresponding to the patient $P1$ and $c^l = \big(\{(\text{BP}, [8; 13]), (\text{ASC}, \{Young, Adult\})\}, \text{Normal}\big)$ be an instance of the context *Blood Pressure*. We have:

$$\begin{aligned}\mu(t, c^l) &= min\Big(\mu_{unit}\big(t, (\text{BP}, [8; 13])\big), \\ &\quad \mu_{unit}\big(t, (\text{ASC}, \{Young, Adult\})\big)\Big) \\ &= min(0.7\ ,\ 1) \\ &= 0.7\end{aligned}$$

## VI. IMPLEMENTATION

Now we have extended the model, we discuss about its consequences over two crucial aspects: (1) the knowledge representation and storage and (2) the generalization process.

### A. Knowledge Representation and Storage

Interestingly, our new extended model does not impact on the knowledge storage described in Section III-A. This is due to three reasons. First, the definition of a *context* remains unchanged. Thus, the structure of the relation KMT has not to be modified. Second, our storage solution does not force to specify a valid subset of values for each contextualizing attribute of each instance. Therefore, the relation KT can indistinguishably store generalized instances of any precision without structural change. Finally, fuzzy hierarchies are defined over attributes of $T$ and thus do not particularly concern the expert knowledge itself[3].

[3]Interested readers on fuzzy hierarchy storage may refer to [8], [9] for further details.

*Example 6:* Table IV displays the tuples of KMT related to the *BPC* context. As we can see, contextualizing attributes, *i.e.*, where $Type = 0$ are *ASC*, *TCC*, *Treatment* and *BP* whereas the contextualized attribute is *BPC*. It should be noted that other contexts can be stored in the same way in this table.

TABLE IV
SAMPLE OF KMT RELATED TO THE BPC CONTEXT

| idCtx | Attribute | Type |
|---|---|---|
| BPC | ASC | 0 |
| BPC | TCC | 0 |
| BPC | Treatment | 0 |
| BPC | BP | 0 |
| BPC | BPC | 1 |
| ... | ... | ... |

Table V displays the tuples of KT related to R5 and R6. It should be noted that even if these two instances of context do not have the same precision, *i.e.*, $Prec(\text{R5}) = 2$ and $Prec(\text{R6}) = 1$, they are stored in the same way.

TABLE V
SAMPLE OF KT RELATED TO THE PIECES OF KNOWLEDGE R5 AND R6

| idCtx | IdInst | Attribute | Value |
|---|---|---|---|
| BPC | R5 | TCC | $\{Hypotensive\}$ |
| BPC | R5 | BP | $> 12$ |
| BPC | R5 | BPC | High |
| BPC | R6 | BP | $[8; 13]$ |
| BPC | R6 | BPC | Normal |
| ... | ... | ... | ... |

### B. Extension of the Contextual Generalization Algorithm

The contextual generalization algorithm we briefly described in Section III-A assumes the unicity constraint holds. This constraint is now relaxed, it is necessary to introduce techniques to elect the most appropriate instance to be used in a generalization. More precisely, two ways to relax this constraint have been proposed: (1) the possibility to incorporate generalized instances and (2) the possibility to use fuzzy hierarchies over contextualizing attributes. Consequently, candidate sequences to a given generalization can be more or less precise and more or less adequate. These two measures thus have to be aggregated to obtain a single score.

*Definition 13:* (Score) Let $t$ be a tuple from $T$ and $c^l$ be an instance of the context $c$. The score function which determines how much $c^l$ represents $t$ is defined as:

$$Score(c^l, t) = Aggr\big(Prec(c^l),\ \mu(t, c^l)\big)$$

It should be noted that this score function definition is voluntarily general. This function is of high importance since the selected contextual generalization is the one associated to

the instance maximizing this score function. Indeed, we are convinced that such function should be defined in collaboration with the users of the system depending on both the application domain and their motivation. Nevertheless, we now provide two possible score functions.

*Score 1:* The first score function we propose can be seen as a basic combination of precision and adequacy. Particularly, let $t$ be a tuple from $T$, $c^l$ be an instance of the context $c$ and $\alpha$ be a user-defined numerical value which ranges over $[0; 1]$. A possible score function, denoted $Score_1$, could be:

$$Score_1(c^l, t) = \alpha \times Prec(c^l) + (1 - \alpha) \times \mu(t, c^l)$$

Here, the parameter $\alpha$ indicates the importance that is given to each component.

*Score 2:* A realistic assumption is that some attributes are of higher importance in a given context. For example, in our case study, instances where *Treatment* values are specified may be considered of utmost importance. To materialize this idea, we assume the existence of a user-defined function, denoted $Priority$, which indicates how important is an attribute in a context. Typically, this function ranges over $[1; +\infty[$ and $Priority(A) > Priority(B)$ means that the attribute $A$ is more important than $B$ in the considered context. With these notations, a priority-based score function, denoted $Score_2$, could be:

$$Score_2(c^l, t) = \sum_{X \in Attrib(c^l)} \big(Priority(X)\big) \times Score_1(c^l, t)$$

Assuming defined this score function, we present ROLL_UP_CTX_GEN, the extended contextual generalization operator. Pseudo-code of the underlying algorithm is displayed in Figure 2. Here, we assume that the useful context is known in advance and omit this very cheap and trivial step in the described process. Schematically, the algorithm operates in two steps. First, given a tuple $t$, the set of candidate instances is searched (line 3). An instance is said to be a candidate to the generalization if its global adequacy is not null. Once extracted, each candidate is scanned to evaluate if its score is greater than the temporary score (line 6). If so, the instance is temporarily considered as the elected one (line 8). When all the instances in $\mathcal{I}_{Cand}$ have been scanned, the operator returns the contextual generalization associated to $Inst$, the last elected instance. Note that we do not consider the case where several instances can have identical highest score. In such a case, different strategies could be applied. First, if the elected instances share the same contextual generalization, we return to the above-described case. In conflicting situations, a policy should be implemented in collaboration with the experts to choose the generalization to return. Examples of policy could be: (1) returning the majority generalization or (2) display the diverse generalizations and their related instance to let the user deciding the most appropriate one.

Searching the candidate instances and electing the one with the highest score could be considered as a time-consuming task depending on the size of both the dataset and the

1: $Inst \leftarrow \emptyset$
2: $ScoreMax \leftarrow 0$, $ScoreTemp \leftarrow 0$
3: $\mathcal{I}_{Cand} \leftarrow selectMatchInst(Ctxt, t)$
4: **for all** $c^l \in \mathcal{I}_{Cand}$ **do**
5: $\quad ScoreTemp \leftarrow Score(c^l, t)$
6: $\quad$ **if** $(ScoreTemp > ScoreMax)$ **then**
7: $\quad\quad ScoreMax \leftarrow ScoreTemp$
8: $\quad\quad Inst \leftarrow c^l$
9: $\quad$ **end if**
10: **end for**
11: **return** the contextual generalization associated to $Inst$

Fig. 2. ROLL_UP_CTX_GEN($t$, $\mathcal{B}_{Know.}$) such that $t$ is a tuple from $T$ and $\mathcal{B}_{Know.}$ is the expert knowledge database

expert knowledge. Nevertheless, we experimentally show the feasibility of our approach in the next section.

## VII. Experimental Study

Using synthetic datasets, we now evaluate the efficiency and scalability of ROLL_UP_CTX_GEN.

### A. Datasets and Setups

**Synthetic Dataset Generation.** Synthetic datasets are generated using a multidimensional random data generator following a random uniform distribution. D10F3C1000CC10S1000 stands for 10 attributes, 3 of them are equipped with fuzzy hierarchies, *i.e.*, the others are equipped with simple crisp hierarchies, the cardinality of each lowest level attribute is 1000, the cardinality of the contextualized attribute is 10 and the size of the dataset is 1000.

**Crisp and Fuzzy Hierarchies Generation.** For sake of simplicity, the two following assumptions are made. First, except for the attribute to be contextually generalized, each attribute is equipped with a hierarchy. Second, this hierarchy contains only one level. Fuzzy hierarchies used in this experiment study implement trapezoidal membership functions. Nevertheless, our approach supports any type of membership function. The generation is driven by two parameters: *Interval* ($I$) and *Slope* ($Sl$). *Interval* specifies the length of the interval where the membership degree is maximum for each generalization. It should be noted that each value $v$ can be generalized into a value $v'$ such that $\mu_{v'}(v) = 1$. *Slope* indicates the slope of the trapeze. Figure 3 displays an example of fuzzy hierarchy generation with $I = 3$ and $Sl = 2$. Crisp hierarchies are similarly generated with $Sl = 0$. Note that the cardinality of upper levels is $\frac{C}{I}$ where $C$ is the cardinality of the lowest level.

**Knowledge Generation** The expert knowledge is randomly generated. The generation is controlled by two parameters: *SizeCtxt* and *NbInstances*. *SizeCtxt* attributes are randomly chosen among the set of $D$ available attributes. For each, the granularity level that is involved in the context is randomly determined. Then, *NbInstances* are randomly created. For each attribute of each instance, the set of values in the condition, *i.e.*, the set of attribute values that is inserted in the *Value*

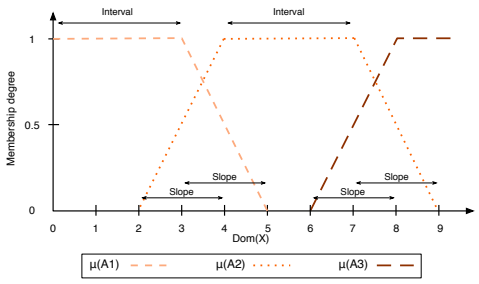



Fig. 3. Impact of the number of fuzzy hierarchies

attribute of KT, is randomly chosen. Cardinality of this set varies between 1 and the cardinality of the attribute. Finally, we check if each tuple of $T$ matches with at least one instance. If not, generation is run again until this condition is fulfilled.

**Implementation and Score Function.** Data are all stored as described in this paper in a Postgresql database. Additionally, the ROLL_UP_CTX_GEN operator has been implemented in the PL/Java language and can be used in SQL queries in WHERE clauses. We used $Score_1$ with $\alpha = 0.5$ as the score function.

### B. Metric and Evaluated Parameters

The ROLL_UP_CTX_GEN operator has been executed on each tuple of the dataset. In this experimental study, the average time needed to perform one contextual generalization is measured. Results are reported in milliseconds. To increase the comprehension of the operator behavior, this global time is decomposed in four distinct times that each represents a crucial step of the algorithm: (1) the selection of candidates instances, (2) the precision computation, (3) the adequacy computation and (4) the score computation.

The impact of the following four parameters has been studied: (1) the number of fuzzy hierarchies, (2) the number of contextualizing attributes, (3) the number of instances of context and (4) the cardinality of contextualizing attributes.

### C. Analysis of Results

**Impact of the Number of Fuzzy Hierarchies.** The proportion of fuzzy hierarchies in the dataset varies from 10% to 95%, *i.e.*, each attribute is equipped with a fuzzy hierarchy except the attribute to be contextually generalize. Some conclusions can be drawn from the results displayed in Figure 4(a). First, the score computation time is negligible. This is verified in each experiment and will not be mentioned anymore. Second, the more the number of fuzzy hierarchies, the slower the aggregation. This is obviously due to an increase of the time needed to compute adequacy. However, both candidate search and precision computation times remain stable. Finally, it can be noted that the global generalization time does not exceed 50 ms.

**Impact of the Number of Contextualizing Attributes.** The proportion of contextualized attributes in the dataset varies from 10% to 100%. Two distinct behaviors can be observed from the results displayed in Figure 4(b). First, from 10% to 30%, most of the running time is spent on the precision computation. Indeed, since the number of contextualizing attributes is low, researching candidates is made with very simple queries. Additionally, the probability to get attributes equipped with fuzzy hierarchies in the context is also low leading to low adequacy computation time. Moreover, since the number of instances does not change, few contextualizing attributes lead to many candidates. This explains why the time to compute the precision of all the candidates is proportionally high. Conversely, from 40% to 100%, most of the running time is spent on the candidate research phase. Indeed, the more attributes in the context, the more complex the queries to find candidates. Additionally, this leads to discover much less candidates. This explains why the time to compute both precision and adequacy is low. Finally, it can be noted that the global generalization time is almost lower than 100 ms.

**Impact of the Number of Instances of Context.** The number of instances of context varies from 10 to 10000. Some observations can be made from the results displayed in Figure 4(c). First, the number of instances obviously impacts on the time to find candidates. Second, the other component times remain stable except for very high number of instances. Indeed, the more instances the more candidates. Finally, it can be noted that the global generalization time is almost lower than 100 ms.

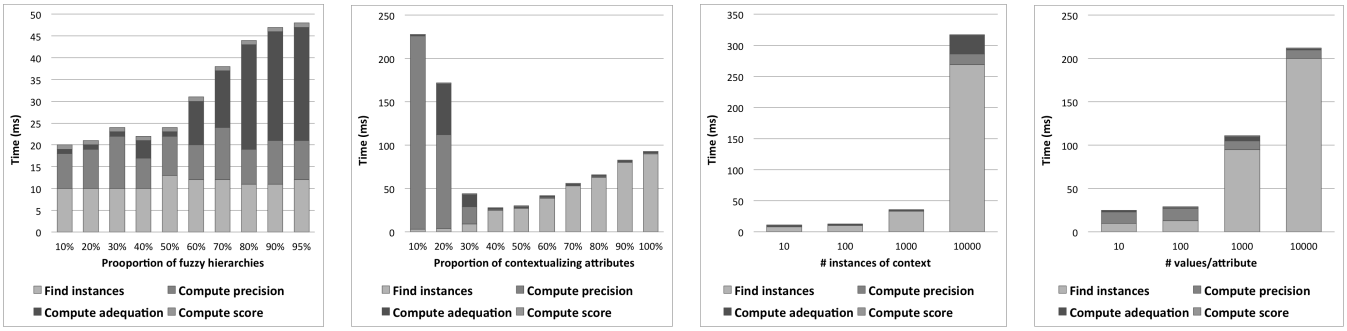
**Impact of the Cardinality of Contextualizing Attributes.** The cardinality of contextualizing attribute varies from 10 to 10000. Results are displayed in Figure 4(d). Clearly, the higher the cardinality, the higher the time to search candidates. Indeed, large domains of definition impact on the number of authorized values in conditions. This naturally leads to execute costly queries to determine if an attribute value appears in subsets of values. However, due to smaller sets of candidates, the time dedicated to other components tends to slightly decrease when cardinality increases. Finally, it can be noted that, with reasonable cardinalities, the global generalization time is low.

**Conclusion.** We highly benefit from the very efficient indexing strategies implemented in the Postgresql core engine to obtain very satisfactory results. These results over synthetic datasets representing extreme cases assess the scalability of the operator.

## VIII. Related Work

### A. Knowledge Integration

Various works have been proposed to integrate knowledge within data warehouses to extend their capabilities, particularly for the hierarchies that define navigation path within the data. Here is an extract of these works. In [14], definitions of rules enhance the flexibility of the navigation since they can express exception in the aggregation process. In [15], the user knowledge allows the creation of new levels in dimension hierarchies. However these hierarchies are still *classical*, *i.e.*, not contextualized. Some other authors proposed to enrich dimension hierarchies with new granularity levels by exploiting semantic relations provided by WordNet [16].




(a) Impact of the number of fuzzy hierarchies (*D20FxC100CC10S1000*, $I = 4$, $Sl = 2$, $SizeCtxt = 10$ and $NbInstances = 1000$)

(b) Impact of the number of contextualizing attributes (*D20F10C100CC10S1000*, $I = 4$, $Sl = 2$, $SizeCtxt = x$ and $NbInstances = 1000$)

(c) Impact of the number of instances of context (*D20F10C100CC10S1000*, $I = 4$, $Sl = 2$, $SizeCtxt = 10$ and $NbInstances = x$)

(d) Impact of the domain of definition size (*D20F10CxCC10S1000*, $I = 4$, $Sl = 2$, $SizeCtxt = 10$ and $NbInstances = 1000$)

Fig. 4. Average time-consumption of contextual generalization on different synthetic datasets

To conclude, enrichment of data warehouses with different types of knowledge is a very promising issue to enhance the analysis capacities. Among the different possible way to specify knowledge, fuzzy logic constitutes an interesting issue.

### B. Fuzzy Issues within the Decisional Process

In the field of data warehouses, many approaches have been proposed to integrate the fuzzy logic to increase the expressivity of hierarchies and/or facts [17], [8], [9], [18], [12].

All these approaches propose multidimensional models that are able to manage fuzzy logic to satisfy different needs, *e.g.*, reality of the data, reliability of sources. Particularly, they focus on the integration of the fuzzy logic during the conception level when designing the data warehouse. In this paper, our purpose is to study the consequences of using fuzzy hierarchies in our contextual hierarchy model and clearly not to propose a new model which integrates fuzziness at the data level. Thus, we are more focused on how the user can express his own knowledge in an expressive way with fuzzy logic to enrich the analysis possibilities. Consequently, cited approaches are complementary with our work.

## IX. Conclusion

In this paper, we have showed the limitations of the existing contextual hierarchy model and have motivated the need to extend it to enhance flexibility and expressivity. Extension of the model has been two-fold. First, *generalized instances of context* have been defined. Second, we have authorized contextualizing attributes to be equipped with *fuzzy hierarchies*. Finally, we have proposed and experimentally evaluated an efficient contextual generalization operator that fits this new model. Future work includes human validation, proposal of a non-relational structure, *e.g.*, a tree-based structure, to store and exploit expert knowledge and the introduction of personalized expert knowledge to differently aggregate data depending on the user.

## References


[1] W. H. Inmon, *Building the Data Warehouse, 2nd Edition*, Wiley, 2 edition, Mar. 1996.

[2] E. Malinowski and E. Zimányi, "OLAP Hierarchies: A Conceptual Perspective", in *CAiSE'04*. 2004, vol. 3084 of *LNCS*, pp. 477–491, Springer.

[3] R. Agrawal, A. Gupta, and S. Sarawagi, "Modeling multidimensional databases", in *ICDE'97*, 1997, pp. 232–243.

[4] E. G. Mallach, *Decision Support and Data Warehouse Systems*, McGraw-Hill Higher Education, 2000.

[5] J. S. Einbinder, K. W. Scully, R. D. Pates, J. R. Schubart, and R. E. Reynolds, "Case Study: a Data Warehouse for an Academic Medical Center", *Journal of Healthcare Information Management: JHIM*, vol. 15, no. 2, 2001.

[6] Y. Pitarch, C. Favre, A. Laurent, and P. Poncelet, "Context-aware generalization for cube measures", in *DOLAP'10*, 2010, pp. 99–104.

[7] D. Dubois, "The role of fuzzy sets in decision sciences: Old techniques and new directions", *Fuzzy Sets and Systems*, vol. 184, no. 1, pp. 3–28, 2011.

[8] D. Fasel and K. Shahzad, "A data warehouse model for integrating fuzzy concepts in meta table structures", in *ECBS'10*, 2010, pp. 100–109.

[9] D. Perez, M. J. Somodevilla, and I. H. Pineda, *Fuzzy Spatial Data Warehouse: A Multidimensional Model*, pp. 3–9, IEEE Computer Society, 2007.

[10] R. Kimball and M. Ross, "The data warehouse toolkit: the complete guide to dimensional modeling", *John Wiley & Sons Inc*, 2002.

[11] L. A. Zadeh, "Fuzzy sets*", *Information and control*, vol. 8, no. 3, pp. 338–353, 1965.

[12] A. Laurent, "Querying fuzzy multidimensional databases: Unary operators and their properties", *International Journal of Uncertainty, Fuzziness and Knowledge-Based Systems*, vol. 11, no. Supplement-1, pp. 31–46, 2003.

[13] A. Ligeza, *Logical foundations for rule-based systems*, vol. 11, Springer-Verlag New York Inc, 2006.

[14] M. M. Espil and A. A. Vaisman, "Efficient Intensional Redefinition of Aggregation Hierarchies in Multidimensional Databases", in *DOLAP'01*. 2001, pp. 1–8, ACM Press.

[15] C. Favre, F. Bentayeb, and O. Boussaid, "A knowledge-driven data warehouse model for analysis evolution", in *CE'06*, 2006, pp. 271–278.

[16] J. N. Mazón and J. Trujillo, "Enriching data warehouse dimension hierarchies by using semantic relations", in *BNCOD'06*, 2006, pp. 278–281.

[17] L. Rokach L. Sapir, A. Shmilovici, "A methodology for the design of a fuzzy data warehouse", *Information Systems Journal*, 2008.

[18] C. Molina, L. Rodríguez Ariza, D. Sánchez, and M. Amparo Vila Miranda, "A new fuzzy multidimensional model", *IEEE T. Fuzzy Systems*, vol. 14, no. 6, pp. 897–912, 2006.