A Methodology for the Design of a Fuzzy Data Warehouse

Lior Sapir, Armin Shmilovici, *Member IEEE*, Lior Rokach

***Abstract*— A data warehouse is a special database used for storing business oriented information for future analysis and decision-making. In business scenarios, where some of the data or the business attributes are fuzzy, it may be useful to construct a warehouse that can support the analysis of fuzzy data. Here, we outline how Kimball's methodology for the design of a data warehouse can be extended to the construction of a fuzzy data warehouse. A case study demonstrates the viability of the methodology.**

***Index Terms*— Data Warehouse, Fuzzy Data Warehouse, Fuzzy Systems**

I. Introduction

A data warehouse (DW) is a special database used for storing business-oriented information for future analysis and decision-making.

The most commonly used methodology today is Kimball's [11], [12]. It describes the process of translating business data and business processes into a *dimensional model*. The dimensional model presents a *fact table*(s) that is indexed by several *dimensions tables* (see Figure 5 for example).

The fact table presents some numeric measures and their descriptive text, such as values that represent interesting measures of the business processes (e.g., costs, profits, etc.). The descriptive values - annotated as dimensions - represent business entities (e.g., customers, time, products, etc.). The facts are presented and manipulated according to the entities combination (e.g., facts regarding the "clothing" product category for the year 2002). In some cases dimensions are also divided into levels of hierarchies (e.g., products, their categories and subcategories).

After designing the basic schema of the DW, *Data staging* and *ETL* processes (Extract, Translate, Load) are used to extract the business oriented data from the organization's transaction-oriented operational databases and load it into the DW. Knowledge workers query the DW for business questions of interest (such as which product is the most profitable). There are already well established methodologies for the construction of a *crisp* data warehouse [10], [11], [12].

The theory of fuzzy sets facilitates the coding of human knowledge in the form of linguistic concepts [20], [13]. For example, the concept product *promotion impact* can be scored as *low*, and *high*. Each promotion can be assigned degree values (usually between 0 and 1) for each label (e.g., 0.3/low, 0.9/high). Afterwards, these values can be incorporated into a computational framework that will support a decision process (e.g., approval of a promotion) [4].

When important business data or business measures or entities are fuzzy, it may be useful to construct a fuzzy data warehouse that can directly support the analysis of fuzzy data. To the best of our knowledge, no one has ever investigated the fuzzy data warehouse.

Fuzzy relational databases extend the relations of the classic relational database and allow the storing and mapping of fuzzy data. An extensive literature exists on fuzzy relational databases [2], [5], [7], [14], [15]. Galindo *et al.* [9] review most of the previous work. The fuzzy EER model [9] supports the modeling of fuzzy attributes, fuzzy aggregations, fuzzy constraints, and more.

The processing of fuzzy data in the OLAP model has been studied by [3]. The aggregation of uncertain and ambiguous data has been studied in [18] and [8]. Delgado *et al.* [6],[7], present the most relevant work in this field - a fuzzy cube model, which includes structural and operational definitions for fuzzy dimensions, fuzzy facts, fuzzy cube tuple, and fuzzy hierarchies. Furthermore, they present operational definitions for aggregations, manipulation by dimensions (roll-up, drill-down, slice, dice), and fact tables.

While the work of [7] presents the mathematical foundations for the fuzzy OLAP *implementation*, it does not discuss the *design* phase. The novelty and contribution in this paper is the extension of Kimball's methodology [12] for the construction of a *fuzzy DW*. Moreover, in this paper we present techniques and guidelines for implementing a fuzzy DW.

The rest of this paper is as follows: Section II presents the design methodology for a fuzzy DW, with an example; section III presents some fuzzy operators and operations supported in our fuzzy DW; and section IV concludes with a discussion.

II. The Multi-Dimensional Modeling Methodology

The purpose of this section is to introduce an extension for Kimball's methodology for designing a DW. The extended methodology enables the incorporation of fuzzy elements into the classical dimensional model. We outline the new capabilities and the greater flexibility of this methodology with a case study.

The case study introduced in this section is based on the

L. Sapir, A. Shmilovici, and L. Rokach are with the Department of Information Systems Engineering, Ben-Gurion University, P.O.Box 653, Beer-Sheva, 84105, Israel.
e-mail: {sapirlio, armin, liorrk}@bgumail.bgu.ac.il
A. Shmilovici is the corresponding author.

"AdventureWorks Cycles" DB[1]. In the case study we will focus on the *tables dealing with sales over the internet and marketing*. The products sold are bicycles, their accessories and components, and related clothing. The customers can be individual customers or commercial stores. Some of the most significant management decisions have to do with pricing and product promotion. Both store management and marketing headquarters spend a great deal of time tinkering with pricing and promotion. Some business data are crisp (e.g., the price of a certain product at a specific time), while other data are fuzzy from a business point of view. For example, stability is a desirable attribute for the sales of a product in general or in a given region. Another example is the case of a product promotion, and its impact on the product's sales. A fuzzy DW can help the chain managers analyze their business and make better business decisions.

We now present the extension of Kimball's four steps methodology [12].

A. The First Step: The Selection of the Business Process

A business process is a natural business activity performed in the organization that is supported by a data-collection system. The understanding of the business requirements should be combined with the understanding of the available data, and translated into business processes. The most efficient means for selecting the business process is listening to the users. The process with the most impact on the business is modeled first. Examples of business processes include raw materials purchasing, ordering, shipping, invoicing, inventory, and sales. This step is similar for the design of either a fuzzy or a crisp data warehouse. The difference is that business processes should be also analyzed for possible (or natural) fuzzy extension in the representation of the chosen process.

*First step example***:** The process with the most impact on the business is modeled first. In the AdventureWorks example, the management wants to understand better customer purchases as captured by the POS (point of sale). They would like to analyze which products are sold at each location and under what promotions.

B. The Second Step: Identifying what is the Grain of the Crisp and Fuzzy Data

The grain conveys the level of detail associated with the fact table measurements. It provides the answer to the question, *"How do you describe a single row in the fact table?"*, or *"What level of detail should be made available in the dimensional model?"* The recommendation is to include the most atomic information captured by a business process [12]. For identifying the grain of the fuzzy data, we need to answer the question *"what is the most atomic level that will give valuable and reliable data?"* Fuzzy data, by their nature, are not precise, and yet they have to be reasonably correct. For facilitating the query processes, we generally prefer that the granularity of the fuzzy data will be equivalent to the granularity of the crisp data. When this is not possible, the designer can incorporate the fuzzy elements in the DW using a summarized or lightly summarized cube. Please remember that an unreliable granularity will decrease the trust of the users in the fuzzy data. It is possible that the designer will discover in steps 3 or 4 that the grain statement is wrong. In that a case, he must return to step 2, re-declare the grain correctly, and revisit steps 3 and 4 again.

Second step example: The atomic information captured by a business process is *a single transaction* done on the internet. For the crisp data warehouse, we keep the highest granularity of the data: the single internet transaction. For the fuzzy data warehouse we will take the same granularity.

C. The third step: selecting the dimensions

The question to ask here is *which data can contribute to the understanding of the business questions?* The *dimensions* represent the data that result from the business process. The dimensions contain descriptive information regarding the facts. The designer should associate each fact with all possible descriptions understandable to the business users. Examples of common dimensions include date, product, customer, Sales Territory, etc.

Once the grain of the fact table has been chosen, most of the dimensions are easily identified. Dimensions which are harder to identify are derived from the business questions, and, therefore, the designer must understand the business questions. The dimensions in a fuzzy data warehouse might contain crisp as well as fuzzy labels. We will first discuss the fuzzy dimension definition (C.1) and then discuss the fuzzy hierarchy definition (C.2).

C.1 Fuzzy Dimensions

In order to identify fuzzy elements in the dimensions, the designer should investigate the source data-collection system for unknown or missing data. Then, he needs to consider if the concepts of the business process may benefit from fuzzy labeling. For example, the *stability* of the sales of a single store is a fuzzy concept *{stable sales, unstable sales}* which is interesting from a business point of view.

A fuzzy dimension is actually an extension of a crisp dimension. For each crisp instance (row) we associate the fuzzy variable values. This is done by defining a fuzzy dimension table with a column for each label of the fuzzy variable (e.g., the labels/columns high, low for the promotion impact variable). The fuzzy dimension might not conform to the crisp one. In some data marts, the users might not need the fuzzy dimensions. In addition the fuzzy dimension might contain many labels, burdening the crisp dimension in terms of disk space. Therefore, this extension can be done by associating a new fuzzy dimension table to the table in a 1:1 relationship, the two being united into a single table on demand. Fig. 1 outlines this relationship. Both tables share the crisp dimension key attribute, which identifies the given attribute instance.

We now describe how to define a fuzzy variable. The basic information needed is the source table and column names, the crisp dimension to associate, the fuzzy dimension name, the attribute type, its properties, and how it is calculated. These details are defined by the DW designer after consulting the "experts" — usually the

[1] The Adventure Works Cycles is a fictitious multinational manufacturing company that manufactures and sells metal and composite bicycles to North American, European and Asian commercial markets. The AdventureWorks DB includes data on sales and marketing, products produced, purchasing needs, vendor relationships and more. Further details can be found in http://msdn2.microsoft.com/en-us/library/ms124825.aspx.

business managers and the fuzzy logics expert if needed. The designer can also use classification and other algorithms which output fuzzy values. The fuzzy attribute values are based on a source crisp value (e.g., age). Each label of the variable (old, young, etc.) has a function which transforms the crisp value into a membership degree value (e.g., trapeze, triangular, etc.). We call these details the "Fuzzy attribute definition report", which is assigned to every fuzzy attribute defined in the model. More details can be found in the following step example. This report is used at the physical design and ETL stages. The ETL programmer populates the DW according to this report. Notice that the designer has to decide when to populate the fuzzy attribute, in the extraction phase or at the data staging phase after the needed crisp values are already populated. The decision should be based on whether additional staging zone calculations are needed or not. For example if the fuzzy attribute is dependent on the age of the customer, then this information already exists at the operational DB.

In cases when the operational systems have missing or vague data, the designer can define fuzzy concepts that will describe it. These concepts can be used as a "certainty" degree regarding the data correctness.

In the following example we represent an example of the "Fuzzy customer age dimension report" (Fig. 2).

Third step example part C.1: Once the grain of the fact table has been chosen, dimensions such as *date*, *product*, and *sales territory* are obvious. Other dimensions, such as the *promotion dimension*, are derived from the business questions. In order to simplify the case study, we assume that there are no missing or incomplete data since the transactions are done on the internet; therefore, the dimensions result from fuzzy business concepts.

The first fuzzy dimension we will discuss and fully demonstrate is the customer *age dimension*. This dimension is probably the easiest to understand and the age attribute is a very common fuzzy attribute. The idea is to define the customer age in groups *{Young, Mid-Age, Old}*. Computing of the customer's membership in each group is based on the customer's age.

In Fig. 1 we present an instance of the "Customer Age" dimension. The figure demonstrates how any given customer age can have several values (*Young, Mid-Age, Old*). A membership degree is given for each value. The "Customer Age" dimension is actually an extension of the crisp "customers" dimension. It is associated to the table in a 1:1 relationship. Both tables share the customer key attribute, which identifies the given customer. In the "Customer Age" dimension we define a column for each label of the fuzzy variable "Age".

TABLE 1: AN EXAMPLE OF THE FUZZY CUSTOMER DIMENSION.

KEY	NAME	AGE	YOUNG	MID AGE	*OLD*
1	C1	40	0.4-0.5	0.8-0.9	*0*
2	C2	55	0	0.6-0.7	*0.3-0.4*
3	*C3*	*35*	*0.7-0.8*	*0.6-0.7*	*0*

In order to populate the fuzzy age dimension the designer has to specify the "Fuzzy customer age dimension report". First we define the source column in the operational DB on which the fuzzy attribute will be based. In our case it is the age field in the customers table. Then, we need to specify the definitions for the fuzzy attribute "Age" and the fuzzy dimension's name. These definitions can be done by specifying a fuzzy classification algorithm to use. In our example we will represent these fuzzy variable labels with trapezoidal membership functions. Trapezoidal functions are defined by four points: alpha, beta, gamma, delta, which graphically form the trapeze (see table and graph, Fig. 2). The points' values are defined by the "experts," us in this example. These functions transform the age into membership degrees. For example, if the age is 40 it is transformed to mid-age/1 and young/0.3. The age variable can already be populated in the extraction phase, since it depends on the customer age that exists in the operational DB. The final report is presented in Fig. 2.

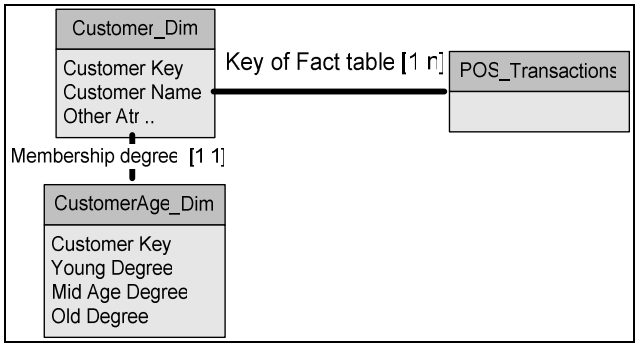

Fig. 1. An example of the fuzzy dimension Customer Age

SOURCE: TABLE- CUSTOMERS, FIELD- AGE
DESTINATION: FUZZY DIMENSION- DIMCUSTOMERAGE
CRISP DIMENSION - DIMCUSTOMERS
FUZZY AGE LABELS:

LABEL NAME	ALPHA	BETA	GAMMA	DELTA
YOUNG	15	20	33	45
MID AGE	30	39	45	60
OLD	50	60	INFINITY	NULL

INTERVALS: YES {0, 0.1, 0.2, 0.3, 0.4, 0.5, 0.6, 0.7, 0.8, 0.9, 1}

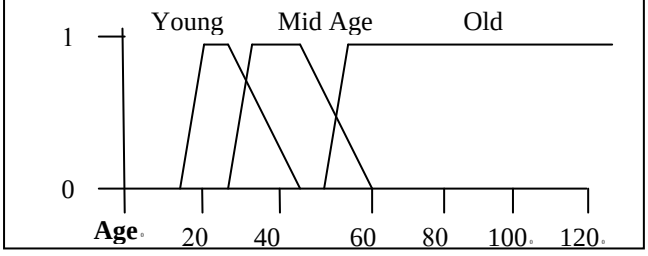

Fig. 2. The "Fuzzy customer age dimension report"

Note that in the last section of the report we specified the attribute's intervals. We have chosen to divide the values into ten equal intervals, but the designer can choose any other separation or use the actual values without intervals. The designer can make use of the intervals when representing the fuzzy attribute as a fuzzy dimension attribute. The intervals are translated into bands (e.g., 0 until 0.1, 0.1-0.2 etc.) The same technique is used when describing numeric crisp fields such as the number of customer's children or yearly income.

After the report is ready, the ETL programmer can use it without having to ask the designer about its definitions. The only additional action is to implement a customized fuzzy trapezoidal membership function. It will use the crisp age column and assign the membership degree of each label to

the customer's age value. The calculation is according to the labels trapezoidal function. In Table 1 we present an example of the union between the customer and fuzzy age dimension.

There are many more optional fuzzy dimensions. For example, consider that the *stability* of the sales in a specific store (possibly also for a specific time and product) could have fuzzy values, (e.g., the sales could be *unstable*, *stable*). Sales stability is an interesting business concept for the management. This fuzzy attribute may indicate if a certain sales territory is always successful or only at specific times (e.g., seasons). The *sales territory stability* may be estimated from the variance in the hourly average sales (in dollar amounts), as recorded by the POS (point of sale).

In the *products dimension*, the management is interested in what product brands items are sold more (or, respectively, less) frequently than others. Therefore, a fuzzy label representing the product brand popularity (e.g., *unPopular*, *Popular*, *VeryPopular*) is useful. This attribute can be estimated by comparing the sum of the product sales to other products' selling rate.

Fuzzy dimensions and attributes can be based on more sophisticated calculations than we have shown, or on results from querying the stores personnel, and yet, in the end will be represented by simple labels that are intuitive and promote the understanding of the data warehouse users (e.g., young, old, stable, etc.). Our methodology supports any fuzzy attribute (some good definitions are presented in Galindo *et al.* [9], chapters 4, 5).

C.2 Fuzzy Hierarchies

In some cases the designer might find that a dimension hierarchy should be represented as a fuzzy hierarchy. For example, consider the product dimension. A typical product dimension may have the product level, the category level, and the subcategory level. In many cases, a product subcategory can belong to several product categories. For example, Bicycle Helmets can be considered either as a clothing element, a bicycle component, or a bicycle accessory. Therefore, the designer can associate the "Helmet" product subcategory to several product categories and assign different membership degrees to each product category (e.g., 0.3/Component, 0.9/accessories, and 0.5/clothing). The technical meaning behind the fuzzy hierarchy is that the child and parent levels will have a many-to-many relationship. This kind of relationship is not common in the classical DW but Kimball did mention the feasibility of such a relationship in a DW. This kind of relationship may occur, for example, when several customers share the same bank account and at the same time have other private bank accounts. The solution for this problem is adding a "bridge" table between the parent and child levels. The bridge table holds combinations of the parent key and the child key and, in addition, the membership degree is added between the two level items. The designer will specify the membership degree between each category and subcategory. The specification of a fuzzy hierarchy is done using a fuzzy variable type that contains a similarity matrix. The similarity matrix defines the membership degree in the bridge table, which connects the parent level and the child level. Galindo *et al. [9]* defined this attribute as a type 3 or 4 attribute in their book. In Table 2, we present the format of such a variable.

The designer can define several membership degree attributes, each one having a different meaning. In such a case the bridge table will be extended with more columns for each similarity degree between the levels.

TABLE 2: THE FORMAT OF A FUZZY VARIABLE WITH SIMILARITY MATRIX.

PARENT / CHILD	P1	…….	PN
C1	0.1	……	0.7
…..	…..	……	…..
CM	0.3	……	0.9

Third step example part C.2: In Fig. 3 we present an example of the fuzzy hierarchy between product category and subcategory. We have defined two similarity degrees the "*fuzzy Categorical Similarity*" and "*crisp Categorical Similarity*". The first describes the fuzzy similarities between the levels and the second one represents the original crisp relationship. The crisp relationship is defined as one when there was a relationship and zero when none.

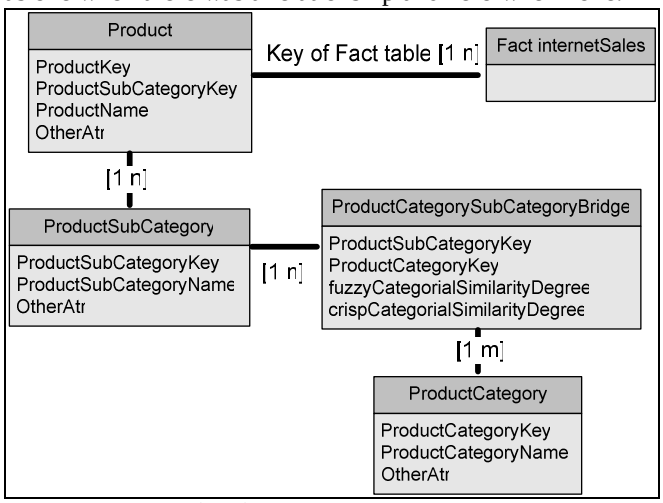

Fig.3. An example of the fuzzy product hierarchy

Using a fuzzy hierarchy has advantages as well as disadvantages. If we use a fuzzy hierarchy, then we can analyze child elements that can have several different parents. On the other hand, the fuzzy hierarchy might make the DW very complex and confusing. The similarity degrees stored at the bridge table can be used as a measure in the fact table. They can also be used as a weight that affects the other measures in the fact table. We will further discuss this property in the next step where we will deal with defining the facts and their measures.

D. The fourth step: identifying the facts

In this step the designer determines which fuzzy and crisp facts will be included in the fact table. When determining the facts, the designer must understand what the business users want to measure. Considering the potential facts, he may discover that some adjustments are required, either to the grain assumptions (step 2), or to the choice of dimensions (steps 3). If a fuzzy fact belongs to a different grain, it will be placed in a different, more or less summarized, fact table. In the case of a crisp fact, the designer can decide to transform the crisp fact to a fuzzy one for adapting to the chosen granularity or to place it in a different table. Examples of crisp facts are the *sales quantity*, *the sales dollar amount*, etc. For efficiency purposes, some crisp or fuzzy measures should be pre-computed (e.g., *profit = sales price – costs price*) and

stored in the fact table.

Fuzzy measures are defined on the base of crisp data/measures when the designer wants to represent measures that cannot be translated to a crisp number. For example, if we consider the *transaction costs* and *price* we can define a fuzzy measure called "*sale profitability*" (*lowProfits, normalProfits, highProfits*). This measure can be computed from the ratio between the price and costs of the transaction. If the ratio is *high* then it will have a higher membership degree to the *highProfits* label. Another option is to use the fuzzy dimensions attributes as measures. For example, we can add the fuzzy measure *product popularity*. Each fact in the fact table has a product key associated to it. Therefore, we can add the product popularity values to the fact table line.

Our fuzzy DW also offers a hybrid measure. The hybrid measure is a combination of fuzzy and crisp measures. This kind of measure can be used when we want to use the fuzzy degree as a weight which affects the crisp measure. For example, we can use the fuzzy product category hierarchy introduced previously as a weight. The fuzzy degrees stored in the bridge table will be added to the fact table and will be used as weights.

The fact table is populated with the fuzzy measures by adding a column to represent each fuzzy attribute label. Again, the designer has to define everything in the fuzzy variable report. The report format is almost identical to the dimension fuzzy attribute report format. The main difference is that we do not specify intervals for the measure. The measure is numeric and we will store for each fact the exact similarity degree to the attribute label. An attribute should not be defined with many labels, because, since for each label we create a separate measure column, the user will be confused if he is presented with more than 3-4 different labels. Moreover, the typical fact table will include more measures (crisp and fuzzy).

Fourth step example: The crisp facts are collected from purchasing transactions over the internet. They include the *sales quantity*, the *sales dollar amount*, *sales costs amount*, and the *net profit*, which is computed at the staging area as sales profits minus costs. The fuzzy measures of interest are the *product popularity*, *sales profitability and promotion impact on sales*. Computation of the first two measures has already been explained in the previous paragraphs.

The *promotions impact* is possibly the fuzziest, since it is difficult to capture the contribution of a promotion to each sale. Therefore, we defined the *promotion impact* label as (*highImpact, lowImpact, mediumImpact*). We estimated the promotion impact from the ratio between *average sales* when the *promotion* was *on* and the *average sales* when there was no promotion. It will be computed at the staging area after we have populated the crisp data needed for the calculation of the formula. In Fig. 4 we present the fuzzy promotion Impact measure report. Notice that in this report we added a formula for calculating the row crisp data into the desired crisp ratios which will be used for the fuzzy measures calculations.

Now that we have designed the basic schema - defining fuzzy and crisp dimensions and measures - let's see how the fuzzy hierarchy defined in the third step affects the DW measures and fact tables. Previously we defined a fuzzy hierarchy and a similarity degree between the products category and subcategory. We added two measures related to the fuzzy hierarchy, defined in the previous step. We will use the similarity degree in the bridge table as a measure, "*fuzzy Categorical Similarity*". The membership degrees stored in the bridge tables can be used as actual measures in the fuzzy fact table, each value being assigned to the relevant fact. For example *Bob*'s *shirt* will be stored once as *accessories* with a membership degree of 0.3 and another time as *clothing* with a membership degree of 0.8. They can also be used as weights which will influence the facts (part of a hybrid measure). The second measure we define is the *"crisp Categorical Similarity"*. This degree will hold the original crisp degree between the product subcategory and category.

FUZZY PROMOTION IMPACT REPORT

SOURCE: SALESTBL.SALESAMOUNT, SALESTBL.PROMOTIONKEY

CALCULATION FORMULA :

promotionIncreaseRatio[i] =
promotionAverageProfits[i]/NoPromotionAverageProfits
i - Number of promotions

DESTINATION : FACTINTERNETSALES

PROMOTION IMPACT LABELS:

Label Name	Alpha	Beta	Gamma	Delta
HighImpact	1.5	5	100	null
MedImpact	1.1	2	3	5
LowImpact	null	1	1.1	2

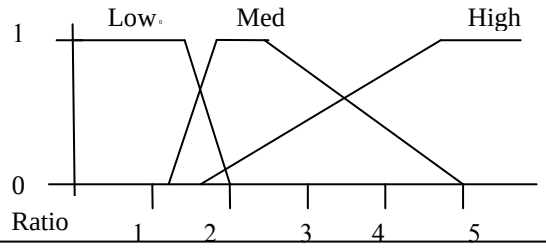

Fig. 4. The fuzzy promotion Impact measure report

Fig. 5 outlines a possible dimensional model for the Adventure Works Cycle internet sales. This schema presents the final outcome of the presented case study example. A box is drawn around the fuzzy measures and a circle around the fuzzy dimensions. The schema represents the fact table with the fuzzy product category hierarchy and the other dimensions.

We recommend that if the user decides to add fuzzy hierarchies to the DW then he should separate the fact tables with fuzzy hierarchies from their “twin” one (without the fuzzy hierarchies). This will help the DW users to avoid confusion and mistakes. Furthermore, the fuzzy measures related to the fuzzy hierarchy are removed. The facts in the crisp "sister" fact table are fewer and different due to the fuzzy hierarchy. The relationship between product category and subcategory is *1:n*. instead of *n:m*. Other fuzzy elements can remain in the "twin" fact table since they do not influence the crisp facts and dimensions.

The fuzzy DW offers several types of fuzzy attributes, all of which can be supported by the fuzzy cube [7]. The fuzzy attributes presented in the schema can be implemented via any type of fuzzy database.

These four steps cover the essentials for designing a

feasible and useful fuzzy DW. This fuzzy DW can be queried with queries such as "present the effect of *promotions* on the *profit*", "what is the *products popularity*", or "what is the *profitability of the sales*". These types of query are not supported in today's "classic" DW model.

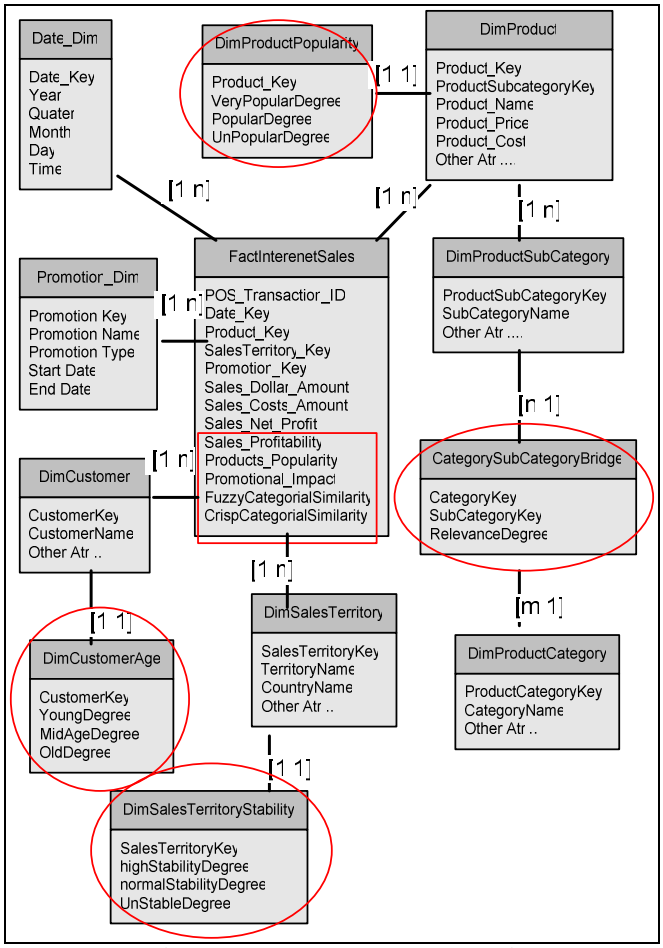

Fig. 5. The new fuzzy data warehouse schema including fuzzy hierarchy.

The correctness of our methodology is demonstrated in an outline of a realistic case study. The example was fully implemented using a customized program which provided the utilities for performing the ETL and a user interface OLAP browsing, of a fuzzy DW.

In the next section we suggest how to use the fuzzy hierarchy measures, fuzzy operators for aggregation, and dimension filtering. More information regarding the implementation issue can be found in [21].

III. Fuzzy Operators and Measure Types

A. Using Correctly the Fuzzy Hierarchy Measures

The *n:m* relationship between the fuzzy hierarchy objects affects the fact table directly. Suppose we have the fact (transaction) that *Bob* bought a *shirt* and the *shirt* belongs with some degree of membership to the *accessories* and to the *clothing* product categories. This fact will be stored (cloned) twice in the fact table even though it actually happened only once. The fuzzy hierarchy may consume excessive disk space if we create a full fuzzy fact table. Each fact in the crisp fact table can possibly clone itself '*m*' times where '*m*' is the number of children that has any degree of similarity to the fact property (the cardinality value). In our case study, it's the number of categories related to a subcategory. Furthermore, if we have several fuzzy levels, then the fact will clone itself recursively further according to the cardinality between the children and their grandchildren.

Fuzzy hierarchy measures can be very useful. Typically, the number of relationships between a child and its several parents is much smaller than the cardinality value ('*m*'). It is dependent on the designer's decision. In our comprehensive case study implementation the maximal enlargement factor of a fuzzy hierarchy was only 3.78 times the original fact table [21].

When using the bridge table technique (as for the product dimension in Fig. 5), we are not forced to actually create the full fuzzy fact table. We can store the information in the bridge tables instead of cloning the same fact again and again. Afterwards, we can create the full or partial fuzzy fact table *on demand*.

B. Fuzzy Aggregation

Fuzzy logic demands the fulfillment of several axioms from an aggregation operator [20], [4]. Computing the fuzzy measures with standard aggregation operations available in crisp DB or DW (e.g., *count*, *sum*, *average*, *num of Childs*) is not efficient. Though the *average*, *min*, *and max* operators are common fuzzy operators, many more fuzzy operators are mentioned in the literature. Therefore, the DW aggregation functions should be extended to support the appropriate fuzzy operators. Some fuzzy operators might be problematic, for example, the standard t-norm and t-conorm operators will reach 0 (t-norm) or 1 (t-conorm) when even one or a few of the aggregated elements equals 0 or 1.

A common aggregation operator family is the ordered weights average operators (OWA), first introduced by Yager [18]. The OWA operators take into account a weight vector where the sum of the weights vector element is equal to 1. The weights vector is multiplied by the actual elements values vector. This ensures that the aggregation result will not reach 1 or 0 easily, and that the result will be according to the semantic value of each item. The drawback of OWA is that it requires the definition of a weights vector and that the items vector needs to be ordered before starting the aggregation. The items' ordering is needed for fulfilling the symmetry axiom of the fuzzy operator. Nevertheless, [1], [17], [8], [19] define OWA operators and demonstrate solutions and applications.

Fuzzy logic experts should choose the aggregation measures. Attempting to define these measures with standard SQL or MDX querying languages without the customized fuzzy operators may be very complex: the DW user will first have to order the sets behind each cell in the fact table, define the weights vector, and compute the multiplication. This will be complex and even impossible for some DW systems.

Because of the ordering of the facts by dimensions, the computational complexity of the crisp DW is *O(nlogn)*. In our implementation of the case study we demonstrated that the fuzzy DW only doubles the computational complexity, since the fuzzy DW adds the extra ordering of the OWA operators. The empirical experiments verified the doubling of the computation time by comparing the processing times

of crisp and fuzzy cubes under the same conditions.

C. Using Fuzzy Dimensions for Filtering

Filters are often applied when querying the DW cubes. For example, the user can choose to see facts only from the last year. Fuzzy dimensions provide natural means to help the user filter the cube in order to identify trends and patterns. The user can use *alpha cuts* or fuzzy quantifiers such as *much, less than* [20] for filtering by a dimension label. The user can also filter the data using *t-norm* and *t-conorm* operators [13], [4] between the fuzzy dimension variable labels. For example, the user can filter the cube to show only facts related to customers which belong *both* to the "*young*" and "*mid-age*" groups. The intersection filter between fuzzy sets focuses the cube on customers which are between these two concepts and allow the user to identify their purchasing behavior.

IV. Discussion and Conclusions

The fuzzy DW has several advantages: Users can make more intuitive and easy to understand queries in a natural-like language. Furthermore, the user may receive several answers to his queries (with different degrees of relevance). Trying to elicit the answers from today's classical DW computed by the fuzzy attributes will usually require a lot of effort (complex SQL or MDX queries) and will not be feasible in some cases (e.g., fuzzy aggregation functions).

Defining fuzzy dimensions allows the user to describe the facts with abstract human concepts which are actually more realistic. In reality, things are typically not black or white but something in between. The fuzzy dimensions also allow more flexible and interesting filtering of the facts. The filters are defined by using simple human concepts (e.g., customers that are related both to the *Young and Mid-Age* groups).

Using fuzzy hierarchies in the cube may help the user to look at the facts from perspectives and points of view which are not possible in the crisp DW. For example, in our case study the DW user can understand better what kind of products the customers buy without being restricted to one categorical association. The user can look at products which belong to the "*accessories*" category as if they belong to the "*clothing*" or "*components*" categories.

The most significant added value of the fuzzy DW is the fuzzy measures. We have demonstrated that fuzzy measures used with fuzzy aggregation operators (e.g., OWA) allow the user to understand his business and the DW measures better. Now, the user can understand the semantics behind the facts which were hidden in the crisp DW. The user can understand, for example, the impact of the *promotions* on the *profits* when the products prices are economical, as well as many other interesting business questions.

Furthermore, there maybe some saving in disk-space due to integration of several crisp attributes into a single fuzzy attribute. For example, we can reduce all the columns that store *demographic* and other information about the customers into a fuzzy attribute "*customer class*." Each class may represent a collection of demographic property values. Instead of saving the number of children, salary, houses, and more we can store this information in a single fuzzy variable.

On the other hand, there are some disadvantages to using a fuzzy data warehouse: the preprocessing of the fuzzy measures may be time-consuming; the queries may be even more time-consuming due to the computation of some complex fuzzy hierarchies or complex inference methods. The fuzzy scheme of the data warehouse may be more complicated than the crisp one due to additional dimensions, fact tables, and relationships. As long as there are no commercial implementations of a fuzzy data warehouse, experts and users may need to be more involved in the design and construction of the system. Last but not least is the huge amount of disk space needed to store many fuzzy hierarchies. Too many fuzzy hierarchies may lead to an unusable or very slow data mart which will be more than the business users patience can take.

In this paper we outlined a new methodology for the design of a fuzzy data warehouse that extends the methodology of [12]. It is relatively simple to master and may ameliorate some of the deficiencies of the classical approach (e.g., the complexities in representing human concepts, representing the data in several perspectives, etc.). We have demonstrated that the fuzzy DW substantially extends the classical crisp DW functionality. The viability of the methodology was demonstrated in a case study. A comparative performance evaluation indicated that the fuzzy DW resulted in only doubling the computational complexity, and quadrupling the space complexity [21]. Future work will adapt the methodology to handle further other aspects of data warehouse design and implementation, such as data staging and ETL processes.

Appendix A: Implementation Issues

In order to implement our ideas on fuzzy DW we had to find a platform which would provide a fuzzy DW, OLAP browsing, and a fuzzy ETL tool. With a single platform we can evaluate the viability of the methodology. Commercial tools did not fit our need for a customized software that supports fuzzy OLAP processing and fuzzy ETL. One exception is the new fuzzy methods provided in Microsoft SQL server, which allows the user to retrieve or group data using fuzzy comparison methods between text format (fuzzy lookup and grouping) [16]. These utilities are very far from being a complete fuzzy ETL tool.

In order to develop a fuzzy DW tool we needed two new capabilities: a) Execution of fuzzy operators as aggregators of the measures; b) The ability to define fuzzy hierarchies which are not classical multidimensional relationships.

While the second capability could be achieved with some improvisations, the first capability could not be solved easily. We needed to use fuzzy operators which do not exist in standard SQL or even Microsoft's MDX language. Therefore, we decided to implement our own user interface and processing tool for analyzing the fuzzy DW.

The main functionality of our program is a customized OLAP browser that allows us to execute the fuzzy operators and handle the fuzzy dimensions and hierarchies. We used the Microsoft SQL server DB as the infrastructure. We used the C# language with its DB interface components (*Ado.Net*), and graphics components

of the *.Net* framework (data grids, etc.). Thus, we could access and process the raw data in the DB and aggregate or manipulate it in any manner we wished. Note that the DW is still stored in the Microsoft SQL Server and has a multidimensional structure. Using the *Ado.Net* technology allows our program to support other commercial DB tools (Oracle, DB2, etc.).

Another required functionality that has to be developed is a fuzzy DW ETL tool. In order to have a fuzzy ETL tool we needed the following: a) The possibility to use customized methods which can take the source data and transform them into fuzzy data and place them in the DW according to our methodology guidelines; b) The ability to define user forms which will collect the fuzzy attributes definitions for populating the fuzzy dimensions, hierarchies, and measures.

It is appropriate to use a commercial ETL tool for fuzzy ETL. Yet, we still need to create special methods for computing the fuzzy values of the destination columns. Another problem is providing these customized extraction methods with the fuzzy definitions they need to transform the crisp data into fuzzy data. Therefore, we preferred to implement an interface for collecting the fuzzy variables definitions.

We decided that it would be more convenient to add more screens to our OLAP browser program for the fuzzy ETL process. Though we defined a general methodology which supports all kinds of fuzzy attributes, we implemented only a small number of specific operators in order to evaluate our methodology. The other classical ETL operations were executed using Microsoft ETL package.

		United States			
F_MidAge	FiscalYear	SumNetProfit	AvgProfitability	CenteredOWAProf...	SemanticOWAProf...
(0-0.1) ...		47300.12	0.6	0.73	0.6
(0) < 30....		558946.21	0.62	0.72	0.58
(0.1-0.2) ...		73832.48	0.62	0.72	0.59
(0.2-0.3) ...		132532.78	0.62	0.76	0.6
(0.3-0.4) ...		133763.6	0.62	0.72	0.63
(0.4-0.5) ...	**2002**	53115.45	0.7	0.85	0.66
(0.4-0.5) ...	**2003**	50588.51	0.67	0.81	0.69
(0.4-0.5) ...	**2004**	125611.74	0.64	0.78	0.65
(0.4-0.5) ...	**2005**	9067.89	0.65	0.76	0.77
(0.5-0.6) ...		164236.07	0.6	0.69	0.59
(0.6-0.7) ...		293521.78	0.63	0.76	0.61
(0.7-0.8) ...		189134.9	0.64	0.78	0.64
(0.8-0.9) ...		356653.84	0.62	0.72	0.64
(0.9-1) 3...		109640.54	0.62	0.76	0.64
(1) 39.0 -...		851851.7	0.61	0.71	0.61

Fig. 6. A screenshot of a simple cube shown in the OLAP browser

In the OLAP browsing screen the user is able to view and analyze the DW he defined and populate it using the ETL screens. The user first defines the desired dimensions and measures he wishes to view. The browsing includes operations such as *roll-up*, *drill-down* and *filtering* the dimensions. Fig. 6 presents a screenshot of a cube shown in the OLAP browser. The cube is defined with the row dimensions fuzzy *Mid-Age* (in intervals 0, 0.1, 0.2…1), and the fiscal year. The column dimension is the *sales territory country*. The crisp measure is the *sum of net profits*. The fuzzy measures are the average of the "*normal profitability product*", the same fuzzy values aggregated using centered OWA and the semantic OWA where the leading variable is sum of net profits. The differences between these measures can be seen. The *Mid-Age* with the interval 0.4-0.5 was drilled to different years.

ACKNOWLEDGMENT

This work was partially funded by Deutsche Telekom Laboratories at Ben-Gurion University.

REFERENCES

[1] Beliakov G., "Learning weights in the generalized OWA operators", *Fuzzy Optimization and Decision Making*, Vol.4, 2005, pp. 119-130.

[2] Bordogna G., Passi G., (Eds.): *Recent Research Issues on the Management of Fuzziness in databases*, Phisica-Verlag, 2000.

[3] Burdick D., Deshpande P.M., Jayram T.S., Ramakrishnan R., Vaithyanathan S.: "OLAP Over Uncertain and Imprecise Data", *The International Journal on Very Large Data Bases (VLDB)*, Vol. 16, issue 1, 2006, pp. 123-144.

[4] Cox, E., *The Fuzzy Systems Handbook*. AP professionals, Boston, 1994.

[5] Dey D., Sarkar S., PSQL, A Query Language for Probabilistic Relational Data, *Data and Knowledge. Engineering*, Vol. 28, No. 1, 1998, pp. 107-120.

[6] Delgado M., Martín-Bautista M.J., Sánchez D., Vila M.A., "On a Characterization of Fuzzy Bags", *Fuzzy Sets and Systems – 10th International Fuzzy Systems Association World Congress*, Istanbul, Turkey, June 30 – July 2, 2003, pp. 119-126.

[7] Delgado M., Molina C., Sánchez D., Ariza L.R., Amparo Vila M, "A fuzzy multidimensional model for supporting imprecision in OLAP", *IEEE international conference on fuzzy systems*, Vol. 3, Budapest, Hungary, 2004, pp. 1331- 1336.

[8] Fodor J.C., Marichal J.-L. and Roubens M. , "Characterization of the ordered weighted averaging operators", *IEEE Transactions on Fuzzy Systems*, Vol. 3, No. 2, 1995, pp. 236–240.

[9] Galindo J., Uralian & Piattini, *Fuzzy databases modeling, Design and Implementation*, IDEA group publishing, 2005.

[10] Inmon W.H., *Building the Data Warehouse*, 4th edition, John Wiley & Sons, 2005

[11] Kimball R., Reevs L., *The Data Warehouse Lifecycle Toolkit*, Wiley & Sons, 1998.

[12] Kimball R., Ross M., *The Data Warehouse Toolkit*, Second Edition, Wiley Computer Publishing, 2002.

[13] Klir G.J., Folger T.A., *Fuzzy Sets, Uncertainty and Information*, Prentice Hall International Editions, New Jersey, 1992.

[14] Medina J. M., Vila M.A., Cubero J.C., Pons O., "Towards the Implementation of a Generalized Fuzzy Relational Database Model", *Fuzzy Sets & Systems*, Vol. 75, Issue 3, 1995, pp. 273 - 28.

[15] Medina J.M., Pons O., Cubero J.C, Vila M.A., Gefred, "A Generalized Model of Fuzzy Relational Databases", *Information Science*, 1994.

[16] Chaudhuri S., Ganjam K., Ganti V., Narasayya V., and Vassilakis T., *"Fuzzy Lookup and Fuzzy Grouping in SQL Server Integration Services 2005", Technical Articles Microsoft SQL Server 9.0 Microsoft Corporation* , September 2005

[17] Xu Z., "An overview of methods for determining OWA weights", *International Journal of Intelligent Systems*, Vo. 20, 2005, pp. 843–865.

[18] Yager, R. R.. "Families of OWA operators", *Fuzzy Sets and Systems*, Vol. 59, 1993, pp. 125–148.

[19] Yager R.R., "*Centered OWA operators*", *Soft Computing*, Vol. 11, 2007, pp. 631-639.

[20] Zimmerman, H., *Fuzzy Set Theory*, Kluwer, 1991.

[21] Sapir L. , *The Fuzzy Datawarehouse*, M.Sc. Dissertation submitted to the Dept. of Information Systems Engineering, Ben-Gurion University, Israel, 2008.