Treatment of imprecision in data repositories with the aid of KNOLAP

Ermir Rogova

School of Electronics and Computer Science

TREATMENT OF IMPRECISION IN DATA REPOSITORIES WITH THE AID OF KNOLAP

by

ERMIR ROGOVA

A thesis submitted in partial fulfilment of the
requirements of the University of Westminster
for the degree of Doctor of Philosophy

2010

I dedicate this PhD thesis to my
parents Ferid and Nadire.

You give me strength and inspiration.
You made me who I am today.

I love you.

Këtë tezë Doktorature iu përkushtoj
prindërve të mi Feridit dhe Nadires.

Ju më jepni forcë dhe frymëzim.
Ju më bëtë ai që jam sot.

Ju dua.

Acknowledgments

First and foremost I would like to express my deepest gratitude to my supervisor and director of studies, Dr Panagiotis Chountas for the countless hours he invested in me and in this research. “Without you this thesis would have been very different. You were the best mentor a student can ever have. You are also a friend. Thank you for everything!”.

My great appreciation goes to my second supervisor, Dr Elia El-Darzi for the kind words of courage and advice. “Thank you for providing that very important different point of view!”

A big “Thank you!” to the University of Westminster for sponsoring my PhD studies.

Cordial thanks to my friend Mr Jonathan Balkind for free advice and a place to stay in London. “I will be forever in your debt”.

Last but not least, to my family: “Thank you for your moral support and for believing in me. I love you!”.

Declaration

The work included in this thesis is the author's own. It has not been submitted in support of an application for another degree or qualification of this or any other university or other institution of learning.

External publications directly related to this thesis

International Conferences

- E. Rogova, P. Chountas, On Imprecision Intuitionistic Fuzzy Sets & OLAP – The Case for KNOLAP, IFSA'07, Springer-Verlag GmBh , DBLP, Theoretical advances and application of fuzzy logic and soft computing, pp. 11-20
- E. Rogova, P. Chountas, K. Atanassov, Flexible Hierarchies and Fuzzy Knowledge-based OLAP, FSKD'07, IEEE Computer Society Press, Vol.2, pp. 7-11
- E. Rogova, P. Chountas, K. Atanassov, The Notion of H-IFS in Data Modelling, IEEE *International Conference on Fuzzy Systems,* IEEE FUZZ-WCCI'08, IEEE Computational Intelligence, pp.1397-1403
- P. Chountas, K. Atanassov, E. Rogova, (2008) H-IFS: Modelling & Querying over Hierarchical Universes, International Conference On Information Processing And Management of Uncertainty-IPMU'2008, ISBN 978-84-612-3061-7,pp 1628-1634
- P. Chountas, E. Rogova, K. Atanassov, S. Mohammed, "The Notion of H-IFS -An Approach for Enhancing Query Capabilities in Oracle10g", IEEE Intelligent Systems'08, IEEE Press ISBN: 978-1-4244-1739-1, 2008, pp.13-8 to13-13 . **Article was selected from the PC of IEEE-IS'08 for revised-**

extended post conf. publication in Journal of Intelligent Systems, Wiley InterScience

- P. Chountas, E. Rogova, K. Atanassov, "Enhancing OLAP Querying with the aid of H-IFS" IFSA-EUSFLAT, International Fuzzy Systems Association (IFSA) and the European Society for Fuzzy Logic and Technology (EUSFLAT). 20-24 July 2009, 1797-1802

Journals

- E. Rogova, P. Chountas, B. Kolev, Intuitionistic Fyzzy Knowledge-based OLAP, Notes on Intuitionistic Fuzzy sets, Vol. 13, No.2, ISSN 1310-4926, 2007, pp.88-100

Book Chapters

- P. Chountas, K. Atanassov, E. Rogova, E. El-Darzi, "Modelling the Vitis Vinifera Domain – The Case for H-IFS", Exit Publishers, ISBN:83-60434-01-8s, 2008, pp.11-20.
- P. Chountas, E. Rogova, K. Atanassov, "Expressing Hierarchical Preferences in OLAP Queries". Uncertainty Approaches for Spatial Data Modeling and Processing, Springer, ISBN: 978-3-642-10662-0, 2010, pp. 61-77.

Book

- P. Chountas, E. Rogova, B. Kolev, V. Tasseva, K. Atanassov, "Generalized Nets in Artificial Intelligence, Vol. Generalised Nets, Uncertain Data, and Knowledge Engineering", ISBN 978-954-322-255-1, pp. 1-149, Publishing House of Bulgarian Academy of Sciences.

Abstract

Traditional data repositories introduced for the needs of business processing, typically focus on the storage and querying of crisp domains of data. As a result, current commercial data repositories have no facilities for either storing or querying imprecise/ approximate data.

No significant attempt has been made for a generic and application-independent representation of value imprecision mainly as a property of axes of analysis and also as part of dynamic environment, where potential users may wish to define their "own" axes of analysis for querying either precise or imprecise facts. In such cases, measured values and facts are characterised by descriptive values drawn from a number of dimensions, whereas values of a dimension are organised as hierarchical levels.

A solution named H-IFS is presented that allows the representation of flexible hierarchies as part of the dimension structures. An extended multidimensional model named IF-Cube is put forward, which allows the representation of imprecision in facts and dimensions and answering of queries based on imprecise hierarchical preferences. Based on the H-IFS and IF-Cube concepts, a post relational OLAP environment is delivered, the implementation of which is DBMS independent and its performance solely dependent on the underlying DBMS engine.

Table of Contents

List of figures

List of tables

"Begin at the beginning and go on 'till you come to the end: then stop"

Lewis Carroll, Alice's Adventures in Wonderland.

Chapter One

1. Introduction

1.1 Research Objectives

The research objectives of this thesis can be defined with respect to levels of data and concepts.

- At the level of data, domain dimensions are generalized by utilising the proposed notion of H-IFS, a branch of Intuitionistic Fuzzy Logic and Sets that allows users to modify the axis of analysis/dimensions on the basis of its definition domain so different users can define the whole set of dimensions differently and not just to alter the linguistic definition of a dimension attribute.
- When it comes to query languages, a set of H-IFS based OLAP operators are proposed that allow users to identify the subgroup or data-cube that satisfies exactly a given request, while an enhanced subgroup or data-cube allows users to extract results previously unknown to them due to the enhancement of the query conditions and grouping statements.
- A the concept level, a meta-model is put forward that captures imprecision as part of the structural elements of a generic multidimensional OLAP architecture and it would be used as a guide for developing an ad-hoc utility that could be implemented on top of any of the commercially available Database Management Systems (DBMS).”

1.2 Treatment of Imprecision in Data Repositories

Traditional data repositories introduced for the needs of business processing, typically focus on the storage and querying of crisp/precise domains of data. As a result, current data repositories [1] have no facilities for either storing or querying imprecise/approximate data. However, when somebody considers scientific data (i.e. medical data, sensor data etc) value imprecision is inherited to scientific measurements.

At the same time, data repositories are expected to cope with the accommodation of amalgamated/integrated data. Such data is organised around several axes of analysis (dimensions). Dimensions are likely to be defined differently according to different groups of users, based either on rigid or flexible hierarchies.

Clearly, data repositories need to have build-in support for value imprecision with reference to their storage capabilities. When it comes to analysis/querying, support for dimensions defined on flexible hierarchies is expected. With respect to answering, responses could be a close approximation of the real value and the focus is on determining the accuracy of the answer, rather than finding the exact answer.

1.3 Data Imprecision in Perception

There have been many attempts to classify various possible kinds of imprecise information. The concern is mainly on the classification [2-4] of imprecise information coming either from the data engineering, or database/data management research. It is suggested that imprecision can be a property of data since:

- Data might be missing or unavailable (incomplete data).
- Data might be present but unreliable or ambiguous due to measurement errors, multiple conflicting errors and so forth.
- Data may simply be the user's best guess.
- Data may be based on defaults and the defaults may have exceptions.

A quantisation of a value; for example, the body temperature of a patient is quantised as being "normal", "a mild fever", "a fever", "a severe fever " and the limits are not precise.

Measurement errors can be interpreted only as human errors. The limits and tolerance of an instrument are well defined by the manufacturer. Furthermore it is recognised that imprecision maybe [5] related with the occurrence or the existence of an event. The reasons for this type of uncertainty are:

- Expert's belief in an event. One may have difficulty justifying such a belief.
- An exception to a general rule. The general rule may be unknown, or it may be known but too complex to be

efficiently implemented. A simpler rule is used, but it may be inexact or inefficient. In cases of artefacts or outliers, a pre-processing effort should be performed to remove them, as this will computationally and practically expensive to model them, assuming there is no loss of domain knowledge.

A diagrammatic synopsis of the different causes of uncertainty is presented in "Figure 1-1".

The belief that indicates the occurrence of an event, and the likelihood of when did an event occur may be not exactly known, thus introducing either "factual" or "temporal" imprecision. However the existence of either "factual" or "temporal" imprecision is emphasizing the need to consider multidimensionality as an information property of imprecise information [6].

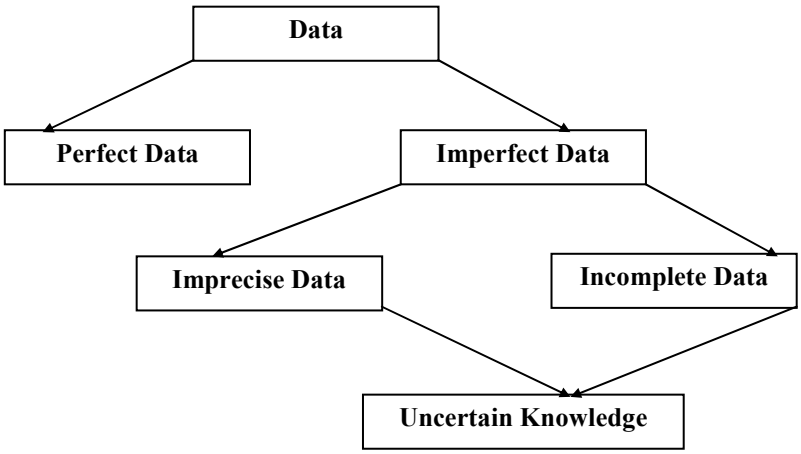


Figure 1-1: "Imprecision / Uncertainty"

A formal language for representing knowledge should be based on a conceptual formalism, otherwise knowledge representation may still

be ambiguous, since there is no description on specifying the ambiguous information elements found during the execution of the knowledge acquisition process. Treating imprecision at the data level without taking into account any conceptual perspective is meaningless, since an imprecise data value does not give any indication about the rationale of its generation.

Flexible/Intelligent information system must be able to represent imprecision at two distinct levels:

- the level of Knowledge/Concepts.
- the level of Data.

Ideally, intelligent/flexible information systems should contain models and information extraction mechanisms in representing or querying imprecise information both at the conceptual and the data level.

1.4 Imprecision in Databases

Imprecision is the partial knowledge of the true value of the real world. It is essentially an epistemic property caused by lack of information. Imprecision in a database context and at the instance or data level is synonymous to "objective ignorance" and it is "context dependent". In that sense, imprecision in databases covers the case where the value of an attribute is given, but not as a "single/atomic" value.

Considering the query mechanism of a DBMS, it can also be deduced that imprecision at the information extraction level is synonymous to "subjective ignorance". This occurs due to the fact that the query mechanism has to make a subjective opinion on a set of facts that is not definitely established.

With reference to [7], ignorance is divided into objective forms of ignorance and subjective forms of ignorance. The distinctions between these categories are vague and it could be argued that there may be an interrelationship between them, since an encoded form of objective ignorance may generate a subjective form of ignorance, as the information extraction system (i.e. a query mechanism) is not precise about the available information.

Objective ignorance states that imprecision related to randomness is an objective property, and the term "likes" qualifies an event that will probably occur. The occurrence of an event "that is likely", is independent of one's opinion about the occurrence of the event. It is an objective property of the cause that generates an event.

Subjective ignorance is related to an agent's belief about the true value of the information, as derived from existing information.

1.5 Definition of Ignorant Information

In this section will be defined what is meant by "Ignorant Information" [8-15]. The definition of ignorant is, perhaps, too concise; it is defined as "not complete". The definition of complete is of more help; complete means *entire* or *whole*. An object, then, is ignorant as opposed to an object that is entire or whole [16-42]. The ignorant object is missing something from its more complete partner. For instance, the fact "temperature 20º", is more complete from one which asserts the temperature to be $20^{o} \pm 4^{o}$, or a fact which states that it could be any temperature.

The informal definition of ignorant information is that information is ignorant because it is missing something from a more complete information, and adding the missing information – which may be unavailable – can ease the imprecision/ignorance property. It is very important to note that information is ignorant only with respect to more complete information (which in turn could be ignorant with respect to still other information).

Let it be assumed that a *fact* is represented by a *relation*. Each *fact instance (tuple)* corresponds to a *multiset* of values. A database, D, is a set of facts, $\{F_1, \ldots, F_n\}$. The meaning of a database D is denoted as $[D]$ and is represented as a set of facts instances $[f]$, where the meaning of each fact depends on the population of fact instances, which in general is a multiset. Each fact in the database models everything that is known about the modelled world.

Each fact instance in the database has two potential interpretations: a *definite* interpretation and a *possible* interpretation. The definite interpretation is all the information that that fact definitely represents, while the possible interpretation is everything that the fact possibly represents.

The fact instances related to a fact are denoted by $[f] = \{f_1, \ldots, f_n\}$, and d_f is the complete fact instance. Based on [15] the *definite* and *possible* interpretation of a fact instance is defined as follows:

Definition 1.4.1. The definite information of a fact instance $[f]$ written def_f, is

$$def_f \equiv d_f \cap f_i,$$

where $f_i \in [f]$, $\cap$ is set intersection.

Definition 1.4.2. The possible information, written $poss_f$,

$$poss_f \equiv d_f \cup f_i - def_f,$$

where $f_i \in [f]$, $\cup$ is set union.

The formal definition of ignorant/imprecise information in a fact instance is related to its possible and definite information. A fact instance can be ignorant/imprecise with respect to another fact instance.

Definition 1.4.3. A fact instance [*f*] is ignorant/imprecise with respect to a fact f' (written $f < f'$) if

$$((def_f \subset def_{f'}) \wedge (poss_f \subseteq poss_{f'})) \vee ((def_f \subseteq def_{f'}) \wedge (poss_f \subset poss_{f'}))$$

The definition states that *f* is ignorant/imprecise with respect to *f'* if it contains either less definite information or more possible information. The utility of these definitions is explained with the aid of an example. Consider the following fact instances of the sample fact "employment" abbreviated as:

[F = emp(Employee-name)],

f_1 = {emp (John)},

f_2 = {emp (John), emp (Anne)},

f_3 ={emp (John), emp (Anne), emp (Adam)}.

The definite and possible information in these facts is given below.

def_{f1} = {emp (John)}, $poss_{f1} = \varnothing$,

$def_{f2} = \varnothing$, $poss_{f2}$ = {emp (John), emp (Anne)},

$def_{f3} = \varnothing$, $poss_{f3}$ = {emp (John), emp (Anne), emp (Adam)}.

Thus, f_2 is ignorant/imprecise with respect to f_1 because it has both more possible information and less definite information. Similarly

f_3 is ignorant with respect to f_2 but only because it has more possible information.

Ignorant information is present at the database level as well as at the fact level (a database is a set of facts). For example, a database that contains two facts is imprecise with respect to another database, which stores those two facts plus a third. Ignorance at the database level is defined below.

Definition 1.4.4. A database D is ignorant/imprecise with respect to another database D' if

$$(\forall\ F_i \in D)\ ((\exists\ F_c \in D')(F_i < F_c))$$

This definition underlines that the incomplete database is missing information from its more complete partner. Imprecision at the schema level will not be explored further, since it is believed that the existence of the uncertainty as a property of data is independent of schema incompleteness.

1.6 Imprecision in OLAP & Data-Warehouses

The multi-dimensional view [43-46] of the data is an essential requirement of OLAP systems. The functionality requirements are divided into four categories:

Data Cube Operations: In this category, slice, dice, drill-down and roll-up operations are the most important. Slicing is the operation of selecting the dimensions used to view the cube. It is analogous to the selection operation in relational algebra. Dicing is the operation of selecting actual positions or values on a dimension. Roll-up is the operation of increasing the granularity along one or more dimensions. Drill-down is the converse operation of decreasing the granularity.

Aggregation Operations: functionality requirements refer to SQL type aggregate operators like (MIN, MAX, SUM, AVG, COUNT).

Handling of Imprecise data: This category includes well-defined mechanism for representing, modifying and transforming imprecise data consistently within the model as well as in associated operations.

Imprecision is persistent all over the real world. There are several sources of imprecision. In empirical situations, imprecision is a result of measurement errors and limits of measuring instruments. When measuring the distances between galaxies, for example, the limitation of instruments used will introduce imprecision. The same

applies to measurements of very small distances – the microscopic world. The natural language also contains ambiguity and vagueness.

A simple OLAP application for business competition analysis would need to handle imprecision in the data. For example, if a firm wants to construct a data-warehouse to store the estimated sales of competitor products over a period of time, the data will necessarily contain some imprecision. Similarly, weekly forecasts of sales normally contain imprecision and when collected for a period of time, the data-warehouse will contain uncertainty measures for each forecast made. Such data-warehouse can be used to analyse the accuracy of the models used to forecast and compare them with actual sales to tune the models in order to better forecast.

Consider the following aggregated wine-sales data in the form of a fact table, “Figure 1-2”.

- Fact-P1: represents the summarised sales of red wines (Bordeaux, Merlot) across the East Region (Dover, Canterbury).
- Fact-P2: represents the summarised sales of white Muscat in the town of Canterbury.
- Fact-P3: represents the summarised sales of White Bordeaux, across the East Region (Dover, Canterbury).
- Fact-P4: represents the summarised sales of white Muscat in the town of Bristol.
- Fact-P5: represents the summarised sales of red Merlot in the town of Cardiff.
- Fact-P6: represents the summarised sales of white Bordeaux in the town of Bristol.
- Fact-P7: represents the summarised sales of red Bordeaux in the town of Cardiff.

- Fact-P8: represents the summarised sales of white wines (Bordeaux, Muscat) in Cardiff.

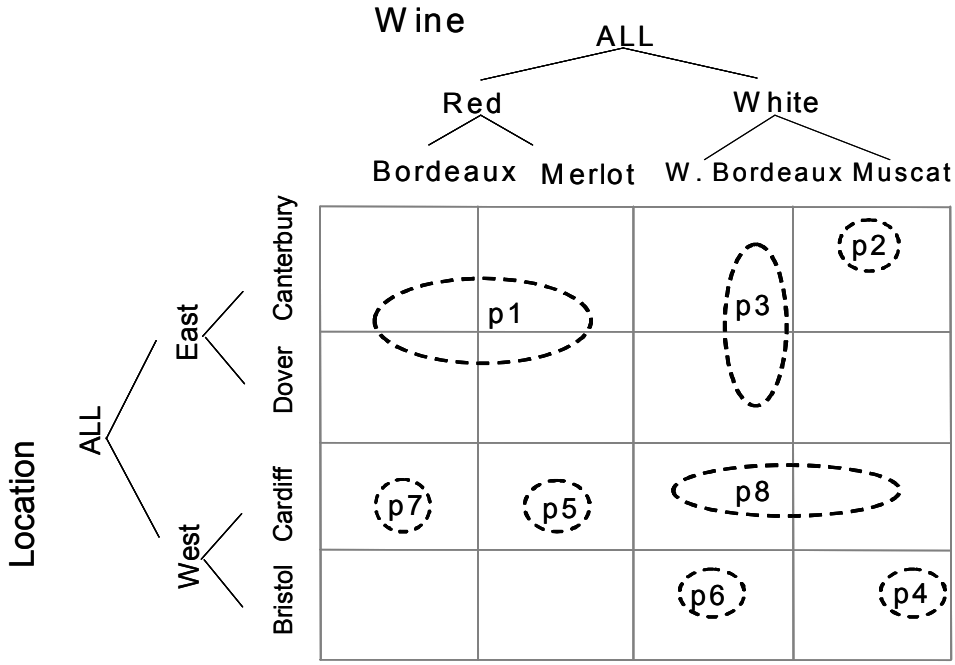


Figure 1-2: Aggregated Sales Data

Consider the query asking for the sales of red Bordeaux in the East Region, or the sales of white Bordeaux in Dover.

Existing OLAP tools cannot provide answers based on the semantics of possible worlds. Thus, they cannot provide possible answers, as part of a query request. In [47] an efficient QC-tree structure is proposed that allows the preservation of the cubic structure and its structural semantics while rolling or drilling down along the axes of analysis. One could possible expand the QC-tree structure to operate over imprecise dimensional domains but in its current form, it cannot be utilized immediately for defining imprecise OLAP

query conditions or grouping statements. In effect, the semantics of possible worlds cannot be accommodated directly by the QC structure in its current form as propose in [47].

Current OLAP database vendors can only deal with precise data domains and dimensions and are not designed to cope with the semantics of possible worlds or with change in dimensions. Research efforts were made by [48] and [49] trying to address the issue of designing an OLAP engine that can cope with the semantics of possible worlds and to that extent with imprecise data. Extensions of the OLAP operators that would be able to deal with imprecise data based on the semantics of possible worlds will also require the extension of the underlying data model, cube model and its dimensions.

Handling of "Kind-of" Relations: Based on the initiatory framework for hierarchical aggregation proposed by [50] new multidimensional models are proposed by [49], [51], [52] and others that can manage imprecision both in the dimensions and the facts as it is claimed. However, approaches proposed by [49], [51], [52] can only support linguistic imprecision at the levels of facts and dimensions. Linguistic imprecision is related usually to fuzzy functions defined behind a linguistic term i.e. high or low but such approaches do not support imprecision on the definition domain of a hierarchical concept/dimension. Moreover in terms of the cubic model [62] functions can be used only as measures while according to [49],[51],[52] same fuzzy linguistic functions can be used as measures as well as dimension attributes, obviously measures and dimensions are semantically different and should not be confused. Moreover [51] is ignoring the fact the OLAP is used to set the axis of analysis before applying data mining. It should not be ignored

that OLAP allows us to verify patterns of behaviour, while data-mining is focused on identifying patterns of behaviour.

This thesis is delivering an approach based on the notion of H-IFS that allows users to modify the axis of analysis/dimensions on the basis of its definition domain so different users can define the whole set of dimensions differently and not just to alter the linguistic definition of a dimension attribute as proposed by [49],[51],[50]. Moreover, the H-IFS OLAP approach and queries can be performed on top of existing OLAP servers without requiring new indexing schemes as in the case of [49]. In particular, the propagation of preference or possibility degrees in a hierarchy that is proposed here, is in adequacy with the object model, in which a query on a given class is also addressed to the subclasses of that class. To this extent, the concept of "Hierarchical Intuitionistic Fuzzy Sets, H-IFS" is proposed and investigated. In the work shown here, the "kind-of" relation defines the hierarchical links. The membership of an element in "H-IFS" has consequences on the membership of its sub elements in this Intuitionistic fuzzy set. "H-IFS" that have the same closure define equivalence classes, and each class has a unique particular representative, called "minimal IFS".

1.7 Conclusions and Chapter's Overview

In this chapter, the meaning of imprecise/ignorant information in relational databases was defined, and the proposed extensions to the relational data model that can represent and retrieve incomplete information were classified. There are many different kinds of ignorant information including information that is fuzzy, imprecise, indeterminate, indefinite, missing, partial, possible, probabilistic, unknown, uncertain, or vague. Each variety of ignorant information will be explored in detail below. Because there are so many different flavours of ignorant information, there have been many proposed extensions of the relational model to support ignorant/imprecise information. To make sense of the multitude, a taxonomy will be provided that brands each proposed extension in a general class of ignorant information models. Also a connection between the dual properties of uncertainty/imprecision will be provided.

First, "imprecise/ignorant" conflicting information is defined and a formal notation for describing databases that contain imperfect information will be build. Next, the formal definition of the kinds of ignorant information found in the literature will be applied.

In Chapter 2, the various kinds of value imprecision will be described and the meaning of each type given, in the context of the formal model semantics. The discussion will be focused on imprecise information stored as attribute values, which is called "*value imprecision*" in the context of data repositories and online

analytical processing (OLAP). It is pointed out that in the case of OLAP, value imprecision has a double-sided effect since it influences the representation needs for the facts as well as for the dimensions of the multidimensional paradigm.

Chapter 3 brings forward a solution named H-IFS that allows the representation of flexible hierarchies as part of the dimensional structures, allowing thus users to define the axis of analysis according to their requirements.

Chapter 4 delivers an extended multidimensional model named "IF-Cube" that allows for the representation of imprecise facts and the answering of queries based on user-defined hierarchical preferences with the aid of H-IFS.

Chapter 5 reveals the metadata of the model and delivers the IF-Oracle ad-hoc utility that is implemented on top of Oracle-10G. IF-Oracle utilizes the concepts of H-IFS and IF-Cube as part of the data definition and manipulation language allowing, thus, users to redefine the axis of analysis and retrieve answers closer to their intent or perception of the reality. During the implementation of IF-Oracle ad-hoc utility, emphasis will be given to the implementation of such scheme and its compatibility with the current SQL standard and data support offered by database and OLAP vendors. The performance of the IF-Oracle ad-hoc utility is solely dependable, due to lack of standardisation across database vendors, on the number of dimensions supported by the back-end Database/OLAP server employed on a particular organisation.

Overall, this thesis is about embedding precise or imprecise data and knowledge as part of an extended OLAP environment, christened Knowledge based OLAP or KNOLAP.

"It is the mark of an instructed mind to rest satisfied with the degree of precision which the nature of the subject admits and not to seek exactness when only an approximation of the truth is possible."

Aristotle

Chapter Two

2. Imprecision and OLAP Data Models

2.1 OLAP Data Models

Online analytical processing, or OLAP, is an approach to quickly answer multi-dimensional analytical queries. OLAP is part of a broader category of business intelligence, which also encompasses relational querying and data mining. OLAP offers a single subject-oriented source for analysing summary data based on various dimensions. It is assumed in multidimensional data analysis that a query needs summary data related to a specific subject and it must consider the data in respect of certain entities. Summary data is usually numerical and measurable. Therefore, the attributes representing them are often called measure attributes. The entities on the basis of which summary data is analysed are called dimensions, represented by dimension attributes. By selecting the specific dimensions through which summary data is analysed one can obtain a view into summary data.

When it comes to OLAP data models, there are two base models in use [53]. Those are: ROLAP – short for Relational On-Line Processing, and MOLAP, which stands for Multidimensional On-Line Processing. The combination of both models offers an alternative model known as HOLAP, which combines features from both basic models, and stands for Hybrid On-Line Processing. The HOLAP approach, allows the model designer to decide which portion of the data will be stored in MOLAP and which in ROLAP.

Both models have their own strengths and weaknesses. ROLAP was first introduced, mainly by commercial vendors like Oracle, DB2,

etc. and utilizes their powerful indexing systems and query optimisation algorithms developed for relational databases. Thus, ROLAP is considered to be more scalable in handling large data volumes, especially models with dimensions with very high cardinality. This is because the data itself is stored on relational databases and manipulated to give the appearance of OLAP functions like slicing and dicing.

The strength of ROLAP is inherited by the relational databases system, which lies on its solid relational database theory. Hence, it can cope with very large amounts of data and also makes use of the functionalities of the relational model. On the other hand, precisely because it is build on top of a relational model, it suffers from performance issues when it comes to aggregated tables. ROLAP relies on the general purpose database for querying and caching, and therefore several special techniques employed by MOLAP tools, such as special hierarchical indexing, were not available until recent improvements in OLAP server models.

The HQC [54] is an approach in which hierarchical dimensions are introduced and the size of data cube is reduced to minimal in order to deliver better query performance. By using hierarchical dimensions, the size of HQC can be reduced without information being lost. However, current HOLAP servers offered by the main database vendors have addressed the needs for special hierarchical indexing. Another weakness, again associated with SQL, would be the shortcoming of SQL operators when it comes to performing complex calculations.

The figure below shows a simple example of the ROLAP model in the form of a star schema. The star schema is a way of implementing multi-dimensional database (MDDB) functionality

using a mainstream relational database. Given the typical commitment to relational databases of most organisations, a specialised multidimensional DBMS is likely to be both expensive and inconvenient.

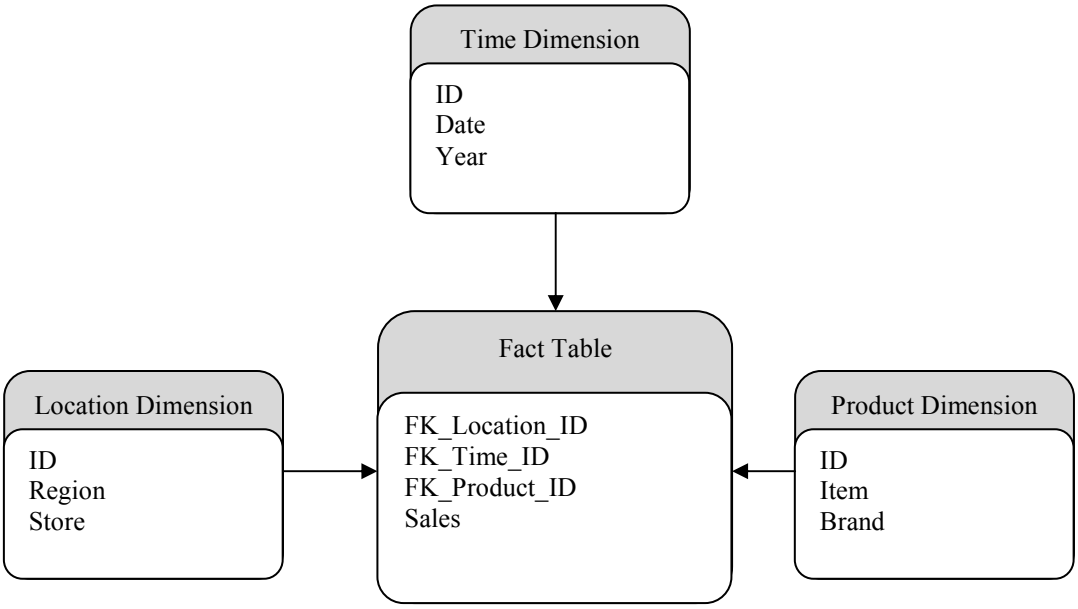


Figure 2-1: An example of a Star Schema

In MOLAP, the data is not stored in a relational database. Instead, it is stored in multidimensional cubes/arrays. This structure gives MOLAP the advantage of performance, when it comes to query execution – especially for queries containing aggregate operators – as the data-cube is build for the very purpose of being able to perform operations like slicing and dicing with ease, as well as the advantage of being able to perform complex calculations. This is because all of complex calculations are made when the cube is first generated, therefore, these complex calculations are not only doable, [43] but they also take a fraction of time to display.

A disadvantage of this model would be that it is limited in the amount of data it can handle. It is very compact for low dimension data sets. As all the calculations are performed when the data-cube is built, it is not possible to include a large amount of data in the cube itself. The cube could be derived from a large amount of data, but only a summary level of information will be included in the cube itself. MOLAP is an important application on multidimensional data-warehouse. For the cube with d dimensions, it can generate 2d cuboids. However, in a high-dimensional cube, it might not be practical to build all these cuboids.

In [55] a method is proposed that partition the high dimensional data cube into shell mini-cubes. The proposed data allocation and processing model also supports parallel I/O and parallel processing as well as load balancing for disks and processors, relevant to grid environments that utilize online analytical processing.

A MOLAP example with the same dimensions and measures as the previous one is shown in the picture below.

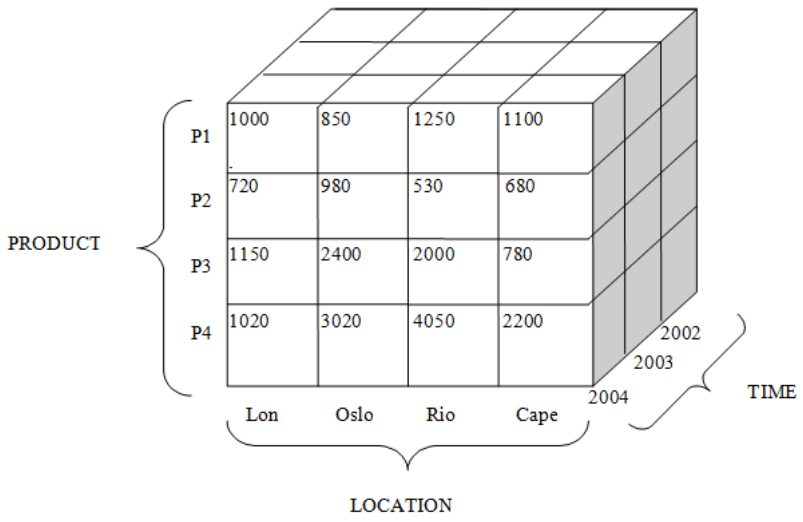


Figure 2-2: An example of the MOLAP data-cube

As both ROLAP and MOLAP models are being used in marketing, management reporting, business process management, budgeting and forecasting, financial reporting and similar areas, it is only natural that there has been research conducted in effort to enable these models to capture and deal with imprecision, when it comes to data analysis. In such environments, there is a need to allow decision makers to describe facts using abstract human concepts. Furthermore, query answers should not be restricted to one categorical association, simply because different fragments of the organisation may treat same data from different points of view when it comes to analysis.

In the next sections, various approaches for treating imprecision in ROLAP and MOLAP data environments will be reviewed.

2.2 Common models of imprecision in ROLAP

In any extended relational environment, one may distinguish three models for treating imprecision at the attribute level, as listed below:

- Probabilistic values
- Fuzzy Values
- Null values

2.2.1 Ignorance and Probabilistic values

Probabilistic information is a deviation of ignorant/imprecise information. A probabilistic data value is a set of alternatives. Each alternative has an associated probability that **it** is the attribute value [16, 17, 18]. For example, in a "wines" database, assume that Merlot's price is not known exactly, but there is 55%-65% certainty that it is £8 and 30%-40% certainty that it is £10. Merlot's price is a probabilistic value; the value exists, it is a value from a known subset of the attribute domain. It is exactly one value, and it is known that some alternatives are more likely than others. In some models, one of the members of the set of alternatives could be an unknown value [19], in which case the associated probability is distributed uniformly over the elements in the domain.

To represent a probabilistic fact using the multi-set (bag) notation, the probability of an alternative is proportional to the membership ratio of that alternative in the multi-set, adjusted by $\pm e$ to reflect a possible margin of error, or missing probability α.

The interpretation for the missing probabilities is that they could be distributed over the entire set of realisations of the attributes [20] including the ones that already appear in the relation. In that case, the imprecision associated with the attribute values for tuples that appear in the relation are represented by probability intervals, and not point estimates. So if f_i is an alternative, fact instance is F with probability p then:

$$[\max(0, p-e), \min(1, p+(e+\alpha)]\cdot \{ f_i \} \in F$$

For example if [F = Stock (Wine-Name, Price)]: f_2 is an instance or situation of F.

$$f_2 = \{\text{Merlot } (8, 10)\}$$

Situation f_2 could arise if, for instance, 20 people are polled to estimate Merlot's price for the upcoming year. Thus, the multiset or bag that represents the domain price consists of 20 members/values. Twelve people estimate Merlot's price of £10 corresponding to point estimate of $p=0.60$=60%. Seven people estimate Merlot's price of £8 corresponding to point estimate of $p=0.35$=35%. One person has not been contacted, thus presented as null value in a multi-set notation corresponding point-missing probability of α=0.05=5%. All people accept that there may be a marginal error in

their predictions of e=0.05=5%. Therefore, there are two different alternatives for the upcoming price of Merlot:

$$8/[\max(0, p-e), \min(1, p+(e+\alpha)] =$$

$$8/[\max(0, 0.35-0.05), \min(1, 0.35+(0.05+0.05)] = 8/[0.30, 0.45]$$

Similarly, the probability for the upcoming price of £ 10 is

10/ [0.55, 0.7].

Hence, f_2 implies possible information:

{[0.55, 0.7](Merlot, 10), [0.3, 0.45](Merlot, 8)}.

2.2.2 Imprecision and Fuzzy values

Another variety of ignorant/imprecise information is fuzzy set information. A fuzzy set is a set of possibilities. Each possibility is a *maybe* value, that *is*, *may belong to the set* or *it may not*. The possibility that it *does* belong is known as the *degree of membership*. The degree is a value between 0 and 1 inclusive. A fuzzy set can be an attribute value [21].

The meaning of a fuzzy set value in a multi-set notation is similar to the meaning of a probabilistic value [22, 23]. But *a possibility* is a subset of the attribute domain, rather than just an element of that domain. If the degree of membership is ignored, then the meaning of a fact instance f_i with a fuzzy set value, with N members is given by a multiset with 2^N members. For example:

f_2 = {Merlot (8, 10)}

For the moment let's set aside the degree of membership for 8 and 10. Then, the meaning of f_2 is {{Merlot(8, 10)}, {(Merlot(8)}, {(Merlot(10)}, {(Merlot( )}}. If it needs to be expressed that £8 is the price of Merlot's, with degree of membership 0.35 and that £10 is the price of Merlot's with degree of membership 0.65 then the meaning of f_2 is {0.35{Merlot (8, 10)}, 0.35{Merlot (8)}, 0.65{Merlot (10)}}

The distinction between probabilistic and fuzzy attribute-values is a controversial issue. However, it can be safely concluded that the meaning of a probabilistic value is a subset of the meaning of a fuzzy value. In the probabilistic meaning, only the singleton sets are retained as elements.

Much of the previous research on value imprecision is related to the semantics and issues of null values.

2.2.3 Imprecision and Null values

A null value represents an *unknown* attribute value. It is a value that is known to exist, but the actual value is unknown. The unknown value is assumed to be a valid attribute value, that is, some value in the domain of that attribute. This is a very common kind-of ignorant information. An ignorant value has various names in the literature

including *unknown null* [24], *missing null* [25], and *existential null* [26].

The meaning of a fact, F, with an unknown attribute value over an attribute domain of cardinality N is a multiset with N members; each member is a set containing an F instance, with the unknown value replacing a different value from the attribute domain. For example, assume f_2 = {Merlot (⊥)} where (⊥ represents an unknown value over a domain {8, 10}), then the meaning f_2 of is:

$$f_2 = \{\{\text{Merlot (8)}\}, \{\text{Merlot (10)}\}\}$$

This corresponds to the notion that a fact with an unknown value is incomplete, compared to a fact where that unknown value is no longer unknown, but is now known to be a specific value (i.e. f_1 = {{Merlot (8)}).

Another generalisation of an unknown fact is a *disjunctive* fact [27], also known as *indefinite* information [28]. A disjunctive fact is a logical "*OR"* applied to fact instances. Let F be an inclusive disjunctive fact with N disjuncts. The meaning of F is given by a multiset with N members; each member is a set, containing one disjunct. For example, the price of Merlot may be £8 or £10. (i.e." Merlot (8), Merlot (10)").

The disjunction could be *exclusive* [29] or *inclusive* [30]. If it is an exclusive disjunction, one and only one disjunct is true. The meaning of an exclusive disjunctive fact is the same as that of an imprecise value. Let f_2 = {{Merlot (8)}, {Merlot (10)}} be an

exclusive disjunctive, then the meaning of f_2 is f_2 = {Merlot (8)} $\vee$ {Merlot (10)}.

The meaning of an inclusive disjunctive fact is somewhat different than that of its exclusive complement, where at least one alternative may be true. Let F be an inclusive disjunctive fact with N disjuncts. The meaning of F is given by a multiset with 2^N-1 members; each member is a unique subset of disjuncts. For example, let's assume the inclusive disjunct f_2 = {Merlot (8) {Merlot (10)}, then the meaning of f_2, is f_2 = {{Merlot (8)}, {Merlot (10)}, {Merlot (8), Merlot (10)}}, excluding the fact, {{Merlot ($\perp$)}. The empty ($\perp$) attribute represents the situation where a fact instance exists, but does not have a particular attribute value.

A *maybe value* is an attribute value, which may or may not exist [31]. If it does exist, the value is known. A *maybe tuple* or *fact-instance* is similar to a *maybe value*, but the entire tuple might not be a part of the relation. *Maybe tuples* are produced when one disjunct of an inclusive disjunctive fact-instance is found to be true.

A combination of *inclusive disjunctive fact instance* and a *maybe fact instance* can determine the semantics of *open information* or *nulls* [32]. The denotation of an *open null* is exact to inclusive disjunctive information with the addition of the empty set as a possible value. That is, the attribute value may not exist, could be exactly one value, or could be many values. For example, in the "wines" database, an open value could be used to present Merlot prices. This value means that Merlot price possibly had a past record, (this could be the first appearance in the market); Merlot price may be one or many. The open value covers all these possibilities. A generalisation of open information is *possible* information [33] (this differs from the use of the term "possible"

here). *Possible information* is an attribute value whose existence is undetermined, but if it does exist, it could be multiple values from a *subset* of the attribute domain.

A *no information* value is a combination of an open value and an unknown value [34]. The *no information* value restricts an open value to resemble an unknown value. A *no information* value may not exist, but if it does, then it is a single value, which is unknown, rather than possibly many values. The meaning of a no information value is similar to that of an *unknown* value with the inclusion of the addition of the empty set as a possible value.

Unknown, partially known, open, no information, and *maybe* null values are different interpretations of a null value. There are other null value interpretations, but none of these is a kind-of well cognisant information.

An *inapplicable* or *does not exist* null is a very common null value. An inapplicable null, appearing as an attribute value, means that an attribute does not have a value [35]. An inapplicable value neither contains nor represents any ignorance; it is known that the attribute value does not exist. Inapplicable values indicate that the schema, usually for reasons of efficiency or clarity, does not adequately model the data. The relation containing the inapplicable value can always be decomposed into an equivalent set of relations that do not contain it. Hence the presence of inapplicable values indicates inadequacies in the schema, but does not imply that information is being incompletely encoded. A single null value is often semantically overloaded to mean either an unknown value or an inapplicable value [36] in which case it is no information null.

In summary, post relational database environments are capturing imprecision with the aid of a set of alternatives or possibilities. Alternatives or possibilities can be expressed with the aid of weights or non-weights.

Weights play an important role in some data models. They are typically normalised values in the range [0, 1]. Weights are assigned to individual alternatives or individual possibilities of an incomplete value. For an alternative, a weight gives the chance or probability that the alternative is *the* actual value. For a possibility, a weight indicates the likelihood that the possibility is *an* actual value.

The un-weighted school is an important special case of the weighted school. The un-weighted school is mainly expressed by the different value semantics that are expressed with the aid of different type of "*null*" values [14]. An appropriate weighting scheme can usually be used to encode un-weighted information using uniform weights. However, the query evaluation semantics with weighted imprecise information differs substantially from that with un-weighted information since queries need to be able to utilize the weighted information. Un-weighted, means not only that weights are not present, but also that query evaluation semantics make no use of weights.

Un-weighted imprecise information may be either *unrestricted* or *restricted*. By restricted it is meant that the value of each possibility or alternative is restricted to a subset of the attribute domain. The subset to which it is restricted must be encoded as part of the imprecise information in the value (if it were kept in the schema, the restriction would simply be to the domain of the attribute). OLAP queries must take into account the restriction

during query evaluation. An unrestricted value has no such constraints.

The three different models for representing ignorance at the data level have been presented with reference to the semantics of definite and possible information.

2.3 Common Models of Imprecision in MOLAP

The need for flexible systems to manage value imprecision has been the focus for database researchers, mainly at theoretical level and in the context of the relational model. There have been two approaches when it comes to modelling value imprecision. One approach uses the probabilistic model to capture, define and process value imprecision. The other approach uses fuzzy modelling for achieving the same results.

New multidimensional models are proposed by [49], [51], [52] and [56] but can only support linguistic imprecision at the levels of facts and dimensions. Such approaches do not support imprecision on the definition domain of a hierarchical concept/dimension. [49], [51], [52] and [56] do not allow users to modify/change the axis of analysis/dimensions on the basis of its definition domain so different users can define the whole set of dimensions differently and not just alter the linguistic definition of a dimension attribute. Moreover, OLAP query execution cannot be performed on top of commercial OLAP servers without requiring new indexing schemes.

At the same time OLAP, technology required [62] the extension of the relational systems with the inclusion of the data-cube and operators to operate over it. Alternatively, new models [46] were proposed to support OLAP based querying on top of multidimensional views.

In OLAP based systems, when it comes to the model level, support for value imprecision will be required at the fact level as well at the level of dimensions with the support of flexible hierarchies.

An interesting approach was used by [57] to address the issue of dealing with imprecise data in OLAP. A "deputy mechanism" is employed that was first introduced by [58] for use in an Object-Oriented model, and then extended by the inclusion of two new concepts: deputy classes and deputy objects. In this model, any real world entity is abstracted as an object, with a unique identifier and with attributes and methods that are used to describe its properties and behaviours. Objects with common attributes and methods are then grouped into separate classes. An object can have one or more deputy objects, and the deputy objects can have their own deputy objects as well. Deputy objects can selectively inherit attributes and methods from their source object(s) as well as extend their own attributes and methods. The deputy objects sharing the same schema are defined by a deputy class. Then, the standard OLAP data model is extended to represent imprecise data and composite measures.

In [57], probabilistic weights are assigned to objects in a process they call "deputy allocation operation". A range of extended query operators (computation join, union, and match join) is defined. The implementation of the above functions in object deputy database was done by: defining the grammar of the statements in parser, so the system can understand the meaning of the statements and extending the analyser, which can transform the commands into standard query structures. The metadata of classes is stored in system catalogues. The process of defining OLAP was done in 4 steps:

- Store basic information into od_class (a catalogue that holds basic schema information) and get a unique class OID,
- Check up the deputy operator rule. There are some deputy rule (allocation, match join, union, etc.) It must follow certain restrictions according to the declared deputy mode. For example, sub-queries shouldn't appear in the deputy rule of the group operator; the join operator needs at least two classes to join.
- Extract the schema of deputy class. Each target expression in the operator rule defines the schema, whose name and data type is the same as that of the expression result. This information will be stored in catalog od_attribute.
- Creating deputy objects. Finally, system will execute the operator rule to create deputy objects for the created deputy class.

By doing this, it is clearly shown that the deputy mechanism is more flexible than the traditional inheritance mechanism, and that it can also improve query performance.

While the allocation of probabilistic weights was proven somewhat effective, the fact remains that they [57] expect the user to offer the domain knowledge and participate in the formulation of the OLAP query. This knowledge is not incorporated anywhere in the multidimensional structure. As a matter of fact, there was no structure in place that could accommodate the knowledge required to perform the weight distribution. Furthermore, the list of operators is very short and does not include all the basic operators that is required in order to process the imprecise data in the repository.

In [59], the OLAP model is extended to represent imprecision and uncertainty by introducing an allocation-based approach to the semantics of aggregation queries. They employ the following *"criterions":*

- *consistency* – that accounts for the relationship between similar queries issued at related nodes in a domain hierarchy, and is supposed to act as a intuition to users' expectation as they navigate the hierarchy.
- *faithfulness* – that captures the intuition that more precise data should lead to better results, and
- *correlation-preservation* – as a requirement that statistical properties of the data should not be affected by the allocation of ambiguous records.

In [59], the OLAP model is extended as following: in order to model imprecision, dimension attributes are assigned as leaf-level values from the underlying domain hierarchy and then introduce a new "kind-of" measure attribute that represents uncertainty. It sees an uncertain value as a range of possible values together with their belief in the likelihood of each possible measure. Hence, a value for an uncertain measure is represented as a "probability distribution function over values from an associated 'base' domain". Furthermore, the extended operators (SUM, COUNT, AVERAGE and LINOP-linear operator) are defined in order to cope with alternative values.

Even though this methodology yields some interesting results, it fails to address the issues of imprecision and user preference at the hierarchical level.

Motivated by the increasing use of OLAP technology for medical applications, [60] investigate how to solve one of the most common problems with medical data, imprecision generated due to the use of multiple hierarchies with different data granularities. The approach described in [60] generally uses the concept of data granularity to handle imprecision in the data. A multidimensional data model is presented and an associated algebraic query language that facilitates the formal dentition of the imprecise concepts. Data imprecision is handled by testing if the data is precise enough to answer a query precisely. If this is not the case, an alternative query that might be answered precisely is suggested. If the user chooses to proceed with the original query, the imprecision is reflected in the grouping of data, as well as in the computation of the aggregate functions. The user is presented with the three different results. The conservative answer includes only what is known to be true, the liberal answer includes everything that might be true, while the weighted answer includes everything that might be true, but gives precise data higher weights than imprecise data. Along with the computation of aggregates, a separate computation of the precision of the result is carried out. As part of the imprecise query execution, the non-definite answer is presented to the user.

Compared to previous approaches to handling imprecision, [60] provides a different point of view as it shows how existing concepts and techniques from multidimensional databases, such as granularities and pre-aggregation, can be maximally re-used to also support imprecision. This yields an interesting approach that can be implemented using current MOLAP technology. As MOLAP involves pre-aggregated data, it would be interesting to see a more theoretical investigation of implementation of this technique.

The presented technique is applicable for the common cases where the data has a degree of imprecision that cannot be ignored, but data precision in any given dimension is expected to be higher than the precision requested by the queries. If the data is highly imprecise, this technique will not be so helpful, as ill defined data can only produce vague answers. This proposal, [60], fails to address imprecision at the conceptual level and provide "single-value" aggregation functions like MIN and MAX, which unlike other aggregation functions, are not readily sensitive to weighting. A single-value function would return only one value as a result of the query, like the highest, the lowest, the average or the sum of values of a specific domain, unlike multi-value functions that can return multiple answers. Another issue, not addressed, would be to allow the user to reformulate the query according to their axis of preference or to seek and obtain more precise data from outside sources.

In [61] a Probabilistic Multidimensional data model similar to the multidimensional data model in [62] and the probabilistic relational model in [63, 64] has been introduced. This model addresses the shortcomings in the OLAP model proposed by [62] by incorporating probability into the model. The probabilistic model in [64] provides a guideline in incorporating this, which was followed by [61]. In this model, each cell in the cube is stamped with probability measure *pS*. The probability stamp represents strength of the belief that there exists a real world object with given cell values. This is in contrast to the [62] model where a cell with a set of values represents existence of a real world object with certainty. As probability is being used to indicate the measure of the strength of belief, its domain is [0,1].When pS is 0, the real world object does not exist and when it is 1, it exists for certain. When it is 0, the

cell values for that object are not represented. When it is 1, the pS is not written explicitly in the cube cell. In this case, the model reduces to the [62] model. The authors also extend the algebra from the algebra provided for data cubes by [62]. The main shortcoming of [64] is that it does not allow for user involvement when it comes to flexibility on query preference or axis of analysis.

2.4 Remarks on OLAP and Imprecision

2.4.1 Initial Approaches to OLAP and Imprecision

Databases are one form of modelling the aspects of the real world. The specific segment of the real world, which a specific database models, is called the enterprise. Nearly all present databases model enterprises that are crisp. A crisp enterprise is one that is highly quantifiable - all relationships are fixed and all attributes have one value. The case of precise-enterprise and precise-data includes virtually all database systems in widespread use. In query language, the issue is whether a particular data item matches a query term when it is not identical to the term. There is a need for a simple query language, in which a user can indicate the degree of relaxation permitted to achieve a match. Even with data having no imprecision, such a query language would be useful.

However, it was and still is the precise enterprise and imprecise data that inspired one of the earliest seminal papers on uncertainty in relational databases [8]. The key notion is that while only one value applies to the enterprise, the database extension may contain a set. The classical approach is to reduce retrieval to 3-valuelogic [9] whether the query language is crisp or not. Each database object is *surely*, *maybe*, or *surely not* response to the query.

The appropriate branch of post relational database systems that deals with problems of this nature is possibility or probability theory. The application to the precise-enterprise and imprecise-data

is obvious. The value in the database is a possibility or probability distribution that is taken to mean the limits of knowledge concerning the actual value as correctly pointed by [10]. The ultimate goal of the flexible-based model is to provide more information about the data retrieved in the *maybe* category.

The relational database model [11] has been proved to be a fruitful paradigm for database research and commercial applications. The relational model blends conceptual simplicity with a solid theoretical foundation, and has an efficient implementation [12]. But, as originally formulated, the relational model could not represent or query *value imprecision*.

Considering all the problems and architectural issues related with the representation of value imprecision at the database level, it could be concluded that up to now value imprecision representation was mainly an implementation issue, therefore solutions were domain-problem oriented and dependent rather than trying to capture imprecision as a direct mapping between the perceived real world and its computer representation. The emphasis was on constructing either *weighted* or *unweighted* models to capture imprecise information.

2.4.2 Current Issues to OLAP and Imprecision

Current research issues for OLAP systems can be summarised as follows:

- Flexible models are required to support value imprecision at fact/data level as well as at the dimension level with the provision of flexible dimensions.
- Flexibility should not be eliminated at the structural level. It should be allowed also at the query level. Users should be allowed to synthesise their own model of dimensions for analysis purposes based on existing structure. Dimensions may be based in either rigid or flexible hierarchies.

The answer to the issues explained above is an enhanced OLAP model that was christened KNOLAP, short for Knowledge-based OLAP. This model will be introduced next but will be thoroughly discussed throughout this thesis.

2.5 Introducing KNOLAP

In the bibliography concerning the introduction of fuzzy methods for replacing unknown values with the aid of background knowledge, several issues have been dealt with, but are quite distant from what is proposed here. There are be two main categories of papers noted, especially in recent research.

In studies about possibilistic ontologies [65], each term of the ontology is considered as a linguistic label and has an associated fuzzy description. Fuzzy pattern matching between different ontologies is then computed using these fuzzy descriptions. This approach is related to those concerning the introduction of fuzzy attribute values in the object relational model [66].

Also, studies about fuzzy thesauri have discussed different natures of relations between concepts. Fuzzy thesauri have been considered, for instance, in [67].

Work reported in [68, 69] in parallel to the framework in this thesis, considers the problem of obtaining a family of fuzzy clusters with clear overlapping by allowing objects to fully belong to several classes. In this framework, the hesitation margin [70, 71, 72] denoting to what extent the overlapping occurs was not considered and cannot be represented directly in the fuzzy hierarchies, classes/clusters. As a result, the ordering and ranking of the query results will differ. Furthermore, different types of background knowledge will be put in use in order to restrict the scope of the query and the number of answers.

Realising a flexible OLAP environment where value imprecision is accommodated at the level of models, give users much more flexibility when queries are imposed and at the same time expands the range of answers obtained in respond to those queries. Thus, main issues to be resolved in this thesis are:

- *Imprecision at the level of multidimensional models*: the semantics of value imprecision have been defined with regard to the main structures of multidimensional modelling (dimensions, hierarchies, facts) and the interrelationships between them.
- Users or different applications should be able to define their own axes of analysis. This is achieved in two parts. On one hand the model has to deal with imprecise data. By "dealing with", it is meant to not only provide premises for storing the imprecise values, but also to process them, perform various OLAP operations on these sets of data and try to get a meaningful answer back. In order to do so, it was required to extend the normal OLAP operators, as they cannot cope with imprecise data. The detailed work can be seen on chapter four. On the other hand, the model devised here should be able to handle imprecise hierarchies and dimensions. This would make the model flexible and able to deal with perceptions and concepts, as well as being much better at incorporating hierarchies from multiple sources. This was achieved by employing H-IFS (Hierarchical Intuitionistic Fuzzy Sets) which will be covered on chapter three. In summary the model should be able to operate on various situations. These would cover cases having:
    - Well defined hierarchies/dimensions and precise data
    - Well defined hierarchies/dimensions and imprecise data
    - Concept based hierarchies/dimensions and precise data
    - Concept based hierarchies/dimensions and imprecise data

- *Flexible Non-Deterministic Query System*: this will allow the querying at the fact level with the assistance of OLAP operators after being re-defined with the aid of Intuitionistic fuzzy logic [73]. More specifically, the introduction of the automatic analysis of queries according to concepts defined as part of a knowledge-based hierarchy in order to guide the query answering as part of an integrated database environment with the aid of Hierarchical Intuitionistic fuzzy sets, H-IFS.

Overall, a unique ontological approach for the treatment of value imprecision is proposed, with regards to multidimensional modelling and flexible structuring of user defined versions of measures based on rigid or flexible hierarchies.

"The hierarchy of relations, from the molecular structure of carbon to the equilibrium of the species and ecological whole, will perhaps be the leading idea of the future."

Joseph Needham

Chapter Three

3. Hierarchical Intuitionistic Fuzzy Sets (H-IFS)

3.1 The Case for H-IFS

Over the past years there has been an increasing interest in expressing user or domain preferences [74] inside database queries. First, it appeared to be desirable property of a query system to offer more expressive query languages that can be closer to user's intent. Second, a classical query in the sense of relational paradigm may also have a restricted answer or sometimes an empty set of answers, while a relaxed version of the query enhanced with background or domain knowledge might be matched by some items in the database.

Frequently, integrated DBMS's contain incomplete data which can be represented using hierarchical background knowledge in order to declare support contained in subsets of the domain. These subsets may be represented in the database as partial values, which are derived from background knowledge using conceptual modelling to re-engineer the integrated DBMS.

Concerning query enhancement, several works such as [75] use a lattice of concepts to generalize unsolvable queries. An extended relational model for assigning possible values to an attribute value has been proposed by [76]. This approach [76], may be used either to answer queries for decision making or for the extraction of answers and knowledge from relational databases. It is therefore important that appropriate functionality is provided for database systems to handle such information.

In studies about possibilistic ontologies [77], each term of an ontology is considered as a linguistic label and has an associated

fuzzy description. Fuzzy pattern matching between different ontologies is then computed using these fuzzy descriptions.

Studies about fuzzy thesauri have discussed different natures of relations between concepts. Fuzzy thesauri have been considered, for instance, in [78].

Recently, in OLAP systems, a need has been identified for enhancing the query scope with the aid of "kind-of" relations that describe knowledge as well as ordering of the elements of a domain or a hierarchical universe. However, no significant attempt has been made for a generic representation of value imprecision, mainly as a property of axes of analysis and also as part of a dynamic environment, where potential users may wish to define their "own" axes of analysis for querying either precise or imprecise facts. To put it differently, various users may wish to define their own dimensions of analysis based on a multidimensional model [79, 80, 81].

In such cases as [79, 80, 81] measured values and facts are characterised by descriptive values drawn from a number of dimensions, whereas values of a dimension are organised in a containment type hierarchy. This need is even more obvious with the move from the classical DBMS environments to multi-source integrated environments, where concept based OLAP is the main query answering system.

3.2 Multidimensional schemas

A multidimensional model is made of the business process that one wishes to analyse, the involved dimensions/entities or axis of analysis [43] which are constructed in a hierarchical fashion, and of measures that hold the quantifying attributes. Multidimensional models can be represented in the form of a star or a snowflake; therefore they are called a star and snowflake multidimensional schema, respectively.

An example of the star schema is shown below, where it can be seen how this model is constructed. In the middle of the star is the fact table or process named as sale, which contains all the measures in the model, in this case, *Units* and *Amount* as well as foreign keys to *dimension/entities tables,* which form the axis of analysis of the star, in this case *Store, Product* and *Time.*

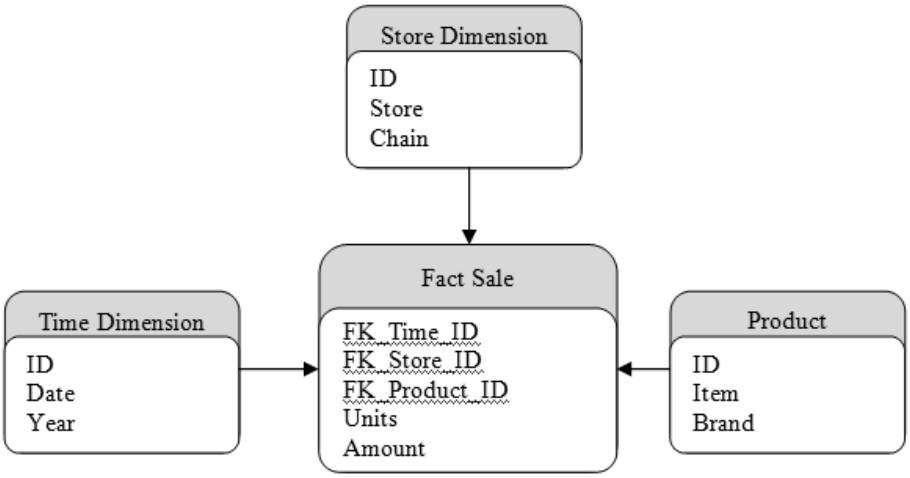


Figure 3-1: An example of a Multidimensional Star Schema

All four elements in this structure are relations connected by means of 1-to-many relationship types. The Fact relation contains a composite primary key, which is composed of several foreign keys - one for each dimension relation - and an attribute for each measure that uses these dimensions. This is where the process to be analysed is contained.

The Dimension relations are where the information for each dimension is contained. Each level of a hierarchy is represented by an attribute on the respective relation.

Dimensions allow the user to modify the axis of analysis when it comes to OLAP queries with the aid of hierarchies. In standard OLAP environments, hierarchies are fixed, meaning that the users are not allowed to modify the axis of analysis by incorporating external domain knowledge [82, 83, 84] relevant to the analysis. If the users were to be allowed to import external domain knowledge in the form of hierarchies, then imprecision is bound to be an issue mainly because:

- Merging multiple hierarchies from different sources cannot be achieved, in most cases, by direct mapping. Instead, an approximation is required.
- Even if there are no multiple hierarchies, users may wish to modify the axis of analysis by incorporating their own concepts.

From the above example, it can be seen that the dimension relations are not on the third normal form (3NF) as described in [43], thus, hierarchies are not levelled directly. Queries would have to specify the level.

When the 3NF is applied to dimensions, the hierarchies now branch out into several tables – because of the normalization process –, each level representing aggregates. The structure now resembles a snowflake, as it can be seen in the example below.

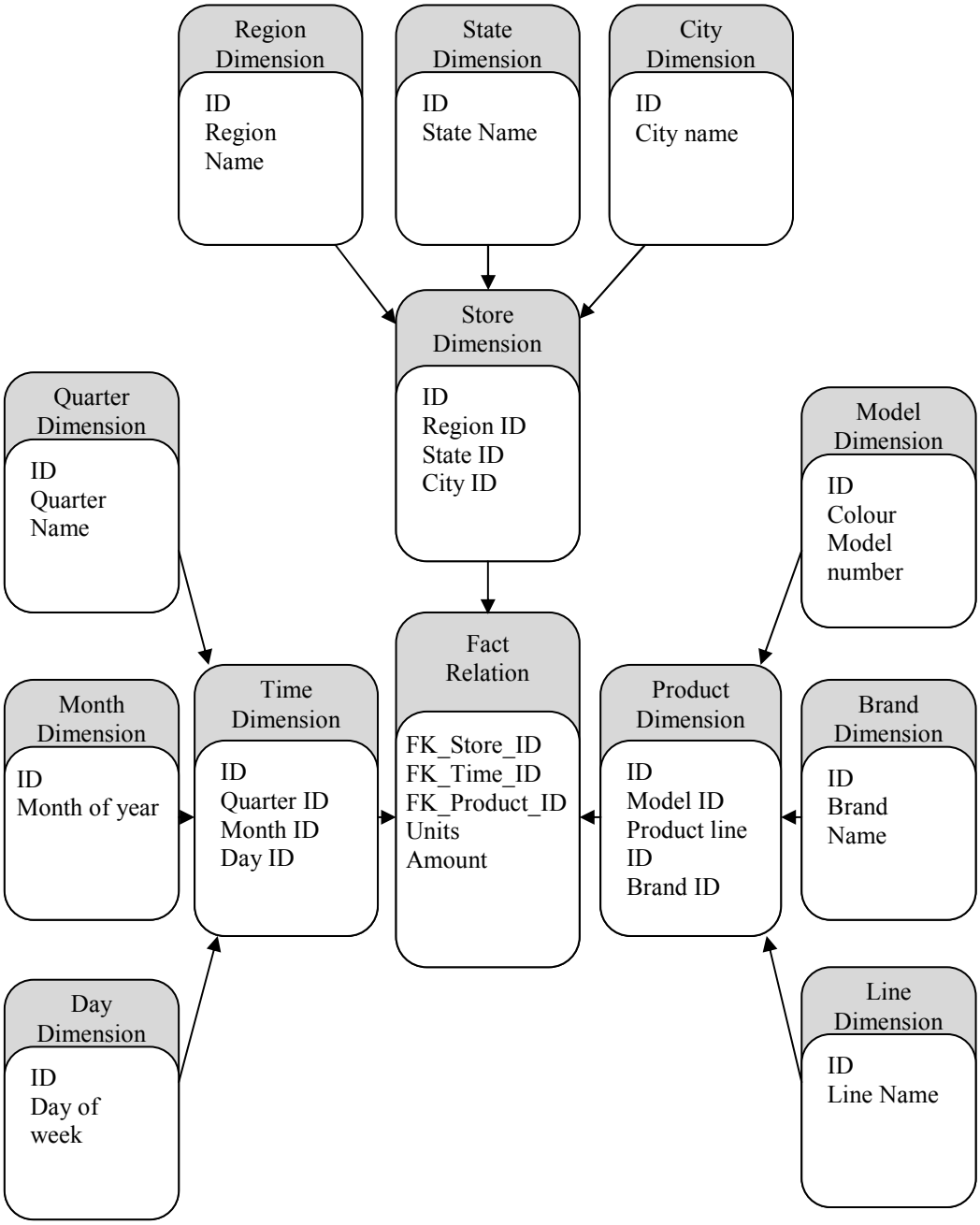


Figure 3-2: An example of a Multidimensional Snowflake Schema

These two schemas differ in the way they represent hierarchies of dimensions. It is this particular bit of information that is of interest. This is because the evidence suggests that there has been no research made on the presumption that the hierarchy of the dimensions may involve imprecision. Yet, if one would take a moment to think about cases of merging hierarchies from different sources, or the fact that hierarchies could be based on the user's concepts of what that hierarchy is, or even the simplest case of taking into account user's preferences on the axis of analysis, one would come into conclusion that imprecision in hierarchies of dimensions is unavoidable.

Currently, when using available OLAP tools in the cases of merging two or more sources, the data is first cleansed, reformatted and then imported into the new structure. However, during this process, the data that doesn't fit the pre-agreed dimensional structures is discarded [43].

In the context of this thesis, the terms of the hierarchy are not fuzzy. These observations led us to introduce the concept of closure of the H-IFS, which is a developed form defined on the whole hierarchy. The definition domains of the Hierarchical Intuitionistic Fuzzy sets (H-IFS) proposed below are subsets of hierarchies composed of elements partially-ordered by the "kind-of" relation. Intuitively, in the closure of the H-IFS, the "kind-of" relation is taken into account by propagating the degree associated with an element to its sub-elements, as more specific elements in the hierarchy.

Based on the above observations, in this chapter, the focus is on incorporating hierarchical preferences in OLAP systems, expressed in the form of background domain-knowledge with the aim on enhancing the OLAP query scope and in return getting an answer that is closer to user's intent.

The rest of this chapter is organized as follows:

- Section 3.3 outlines the principles of the IFS and delivers the basic definitions and properties of the H-IFS.
- Section 3.4 delivers the representation of a H-IFS with the aid of concept based relations.

3.3 Intuitionistic Fuzzy Sets and Hierarchical-IFS

3.3.1 IFS – Atanassov's Sets

Each element of an Intuitionistic fuzzy [85, 86] set has degrees of membership or truth (μ) and non-membership or falsity (ν), which don't sum up to 1.0 thus leaving a degree of hesitation margin (π). As opposed to the classical definition of a fuzzy set given by

$$A' = \widetilde{A} = \{< x, \mu_A(x) > | \, x \in X\}$$

where $\mu_A(x) \in [0,1]$ is the membership function of the fuzzy set A′, an Intuitionistic fuzzy set A is given by:

$$A = \{< x, \mu_A(x), \nu_A(x) > | \, x \in X\}$$

$$\mu_A : X \to [0;1] \text{ and } \nu_A : X \to [0;1]$$

such that $0 \le \mu_A(x) + \nu_A(x) \le 1$ and $\mu_A : X \to [0;1], \nu_A : X \to [0;1]$ denote a degree of membership and a degree of non-membership of $x \in A$, respectively. Obviously, each fuzzy set may be represented by the following Intuitionistic fuzzy set:

$$A = \{< x, \mu_A(x), 1 - \mu_A(x) > | \, x \in X\}$$

For each Intuitionistic fuzzy set in X, let's call $\pi_A(x) = 1 - \mu_A(x) - \nu_A(x)$ an Intuitionistic fuzzy index (or a hesitation margin) of x ∈ A which expresses a lack of knowledge of whether x belongs to A or not. For each $x \in A \; 0 \le \pi_A(x) \le 1$

Definition1. Let A and B be two Intuitionistic fuzzy sets defined on a domain X. A is included in B (denoted A ⊆ B) if and only if their membership functions and non-membership functions satisfy the condition:

$$(\forall X \in X) \; (\mu_A(x) \le \mu_B(x) \; \& \; \nu_A(x) \ge \nu_B(x))$$

Two scalar measures are classically used in classical fuzzy pattern matching to evaluate the compatibility between an ill-known datum and a flexible query, known as:

a possibility degree of matching, Π (Q/D)
a necessity degree of matching, N (Q/D)

Definition 2. Let Q and D be two Intuitionistic Fuzzy Sets defined on a domain X and representing, respectively, a flexible query and an ill-known datum.

The possibility degree of matching between Q and D, denoted Π(Q/D), is an "optimistic" degree of overlapping that measures the maximum compatibility between Q and D, and is defined by:

$$\Pi(Q/D) = \left\langle \sup_{x \in X} \min(1 - \nu_Q(x), \nu_Q(x)), \inf_{x \in X} \max(1 - \nu_D(x), \nu_D(x)) \right\rangle$$

The necessity degree of matching between Q and D, denoted N(Q/D), is a "pessimistic" degree of inclusion that estimates the

extent to which it is certain that D is compatible with Q, and is defined by:

$$N(Q/D) = \left\langle \inf_{x \in X} \max(\mu_Q(x), 1-\mu_Q(x)), \sup_{x \in X} \min(\mu_D(x), 1-\mu_D(x)) \right\rangle$$

The problem occurring from defining Intuitionistic Fuzzy Sets based on the "kind-of" relation is that two different Intuitionistic Fuzzy Sets on the same hierarchy do not necessarily have the same definition domain, which means they cannot be compared using the classic comparison operations Π(Q/D), N(Q/D), for this reason the notion of H-IFS is provided below.

3.3.2 The Notion of H-IFS

The definition domains of the hierarchical fuzzy sets [79, 80, 81, 84] that are proposed below are subsets of hierarchies composed of elements partially ordered by the "kind-of" relation. An element l_i is more general than an element l_j (denoted $l_i \sim l_j$), if l_i is a predecessor of I in the partial order induced by the "kind-of" relation of the hierarchy. An example of such a hierarchy is given in "Figure 3-3". An Hierarchical Intuitionistic Fuzzy Set is then defined as follows:

"An H-IFS is an IFS whose definition domain is a subset of the elements of a finite hierarchy partially ordered by the kind-of relation".

Definition 3. Let F be a H-IFS defined on a subset D of the elements of a hierarchy L. Its degree is denoted as <μ, ν>. The

closure of F, denoted clos(F), is a H-IFS defined on the whole set of elements of L and its degree $<\mu, \nu>_{clos(F)}$ is defined as follows:

For each element **I** of L, let $S_L = \{l_1, \ldots, l_n\}$ be the set of the smallest super-elements in D.

If S_L is not empty,

$$< \mu, \nu >_{clos(F)} (S_L) = < \max_{1 \le i \le n} \mu(L_i), \min_{1 \le i \le n} \nu(L_i) >$$

else

$$< \mu, \nu >_{clos(F)} (S_L) = < 0,0 >$$

In other words, the closure of a H-IFS F is built according to the following rules. For each element l_1 of L:

- If l_I belongs to F, then l_I keeps the same degree in the closure of F (case where $S_L = \{ l_I \}$).
- If l_I has a unique smallest super-element l_I in F, then the degree associated with l_I is propagated to L in the closure of F, $S_L = \{ l_I \}$ with $l_I > l_I$)
- If L has several smallest super-elements $\{l_1, \ldots, l_n\}$ in F, with different degrees, a choice has to be made concerning the degree that will be associated with l_I in the closure. The proposition put forward in definition 3, consists of choosing the maximum degree of validity {μ} and minimum degree of non validity {v} associated with $\{l_1, \ldots, l_n\}$. It is referred to as the *Optimistic strategy*.

A *Pessimistic strategy* can also be utilized, which consists of choosing the minimum degree of validity {μ} and maximum degree of non validity {v} associated with $\{l_1, \ldots, l_n\}$.

Alternatively, an *Average strategy* could be utilized, which consists of calculating the IF-Average and applying it to the degrees of validity {μ} and non-validity {v}.

It has been observed that two different H-IFSs, defined on the same hierarchy, can have the same closure, as in the following example:

The H-IFSs: Q={Wine<1,0>, Red Wine<0.7,0.1>, Rose Wine<1,0>, White Wine <0.4,0.3>} and

R ={Wine<1,0>, Red Wine<0.7,0.1>, Rose Wine<1,0>, Pinot <0.4,0.3>} have the same closure, represented in "Figure 3-3" below. Such H-IFSs form equivalence classes with respect to their closures.

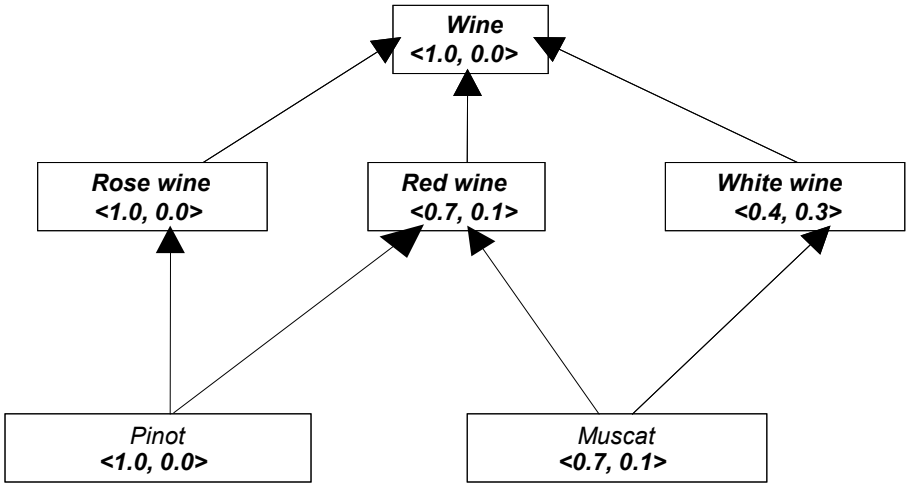


Figure 3-3: Common closure of the H-IFS's Q and R

Definition 4. Two H-IFSs Q and R, defined on the same hierarchy, are said to be equivalent Q≡R if and only if they have the same closure.

Property: Let Q and R be two equivalent Intuitionistic hierarchical fuzzy sets. If $l_I \in dom(Q) \cap dom(R)$, then $<\mu,\nu>(Q.l_I) = <\mu,\nu>(R.l_I)$

Proof: According to the definition of the closure of a H-IFS F, (*definition 3)*, the closure of F preserves the degrees that are specified in F. As Q and R have the same closure (by definition of the equivalence), an element that belongs to Q and R necessarily has the same degree $<\mu,\nu>$ in both.

It can be noted that R contains the same element as Q with the same $<\mu,\nu>$, and also one more element Pinot<1,0>. The $<\mu,\nu>$ associated with this additional element is the same as in the closure of Q. Then it can be said that the element, Pinot<1,0> is derivable in R through Q. The same conclusions can be drawn in the case of Muscat <0.7, 0.1>

Definition 5. Let F be a hierarchical fuzzy set, with $dom(F) = \{l_1, ...,l_n\}$, and $F\text{-}_k$ the H-IFS resulting from the restriction of F to the domain $dom(F) \setminus \{l_k\}$, l_k is deducible in F if:

$$<\mu, \nu>clos_{(F-k)}(l_k) = <\mu, \nu>clos_{(F)}(l_k)$$

As a first intuition, it can be said that removing a derivable element from a hierarchical fuzzy set allows one to eliminate redundant information. But, an element being derivable in F does not necessarily mean that removing it from F will have no consequence on the closure: removing k from F will not impact the degree associated with k itself in the closure, but it may impact the degrees of the sub-elements of k in the closure.

For instance, if the element Rose Wine is derivable in Q, according to *definition 5*, removing Rose Wine <1,0> from Q would not modify the degree of Rose Wine itself in the resulting closure, but it could modify the degree of its sub-element Pinot. Thus, Rose Wine <1,0> cannot be derived or removed. This remark leads us to the following definition of a minimal hierarchical fuzzy set.

Definition 6. In a given equivalence class (that is, for a given closure C), a hierarchical fuzzy set is said to be **minimal** if its closure is C and if none of the elements of its domain is derivable**.**

3.3.3 Obtaining the Minimal H-IFS

Step 1: Assign Min-H-IFS ← ∅. Establish an order so that the sub-elements $\{l_1,\ldots,l_n\}$ of the hierarchy L are examined after its super-elements.

Step 2: Let l_1 be the first element and $\mathbf{(l_1)/<\mu, \nu> \neq (l_1)/<0, 0>}$ then add $\mathbf{l_1}$ to Min-H-IFS and $\mathbf{<\mu, \nu>_{clos(Min\text{-}HIFS)}\ (l_1)= (l_1)/<\mu, \nu>}$.

Step 3: Let us assume that K elements of the hierarchy L satisfy the condition $\mathbf{<\mu, \nu>_{clos(Min\text{-}HIFS)}\ (l_i)=(l_i)/<\mu, \nu>}$**.** In this case the Min-H-IFS do not change. Otherwise go to next element $\mathbf{l_{k+1}}$ and execute Step 4.

Step 4: The $\mathbf{l_{k+1}/<\mu_{k+1}, \nu_{k+1}>}$ associated with $\mathbf{l_{k+1}}$**.** In this case $\mathbf{l_{k+1}}$ is added to Min-H-IFS with the corresponding $\mathbf{<\mu_{k+1}, \nu_{k+1}>}$**.**

Step 5: Repeat steps three and four until $\mathbf{clos_{(Min\text{-}HIFS)}=C}$.

For instance the H-IFSs S_1 and S_2 are **minimal** (none of their elements is derivable). They cannot be reduced further.

S_1= Wine<1,0>

S_2={Wine<1,0>, Red Wine<0.7,0.1>, Pinot<1,0>, White Wine <0.4, 0.3>}

3.4 Representing H-IFS as Concept Relations

The structure of any H-IFS can be described by a domain concept relation DCR = (Concept, Element), where each tuple describes a relation between elements of the domain on different levels. The DCR can be used in calculating recursively [26] the different summarisation or selection paths as follows:

$$PATH \leftarrow DCR_{\{x=1\ldots(n-2)|n>2\}} \triangleright\triangleleft DCR_x$$

If n≤2, then DCR becomes the Path table as it describes all summarisation and selection paths. These are entries to a knowledge table that holds the metadata on parent-child relationships. An example is presented below.

DCR	
Concept	**Element**
Wine <1.0, 0.0>	Rose Wine <1.0, 0.0>
Wine <1.0, 0.0>	Red Wine <0.7, 0.1>
Wine <1.0, 0.0>	White Wine <0.4, 0.3>
Rose Wine <1.0, 0.0>	Pinot <1.0, 0.0>
Red Wine <0.7, 0.1>	Pinot <1.0, 0.0>
Red Wine <0.7, 0.1>	Muscat <0.7, 0.1>
White Wine <0.4, 0.3>	Muscat <0.7, 0.1>

Table 3-1: Domain Concept Relation

Table 3.1 shows how the Wine hierarchy knowledge table is kept. Paths are created by running a recursive query that reflects the 'PATH' algebraic statement. The sample hierarchical IFS comprises of 3 levels, thus calling for the SQL-like query as below:

SELECT A.Concept as Grand-concept, b.concept, b.element
FROM DCR as A, DCR as B
WHERE A.child=B.parent;

This query will produce the paths shown in 'Table 3-2", which presents a pictorial view of the four distinct summarisation and selection paths. These paths will be used in fuzzy queries to extract answers that could be either definite or possible. This will be realised with the aid of the predicate (θ).A predicate (θ) involves a set of atomic predicates (θ_1, …, θ_n) associated with the aid of logical operators *p (* i.e. ∧, ∨, etc.). Consider a predicate θ that takes the value "Red Wine", θ = "Red Wine".

Path			
Grand-concept	**Concept**	**Element**	**Path Colour**
Wine <1.0, 0.0>	Rose Wine <1.0, 0.0>	Pinot <1.0, 0.0>	**Red**
Wine <1.0, 0.0>	Red Wine <0.7, 0.1>	Pinot <1.0, 0.0>	**Blue**
Wine <1.0, 0.0>	Red Wine <0.7, 0.1>	Muscat <0.7, 0.1>	**Green**
Wine <1.0, 0.0>	White Wine <1.0, 0.0>	Muscat <0.7, 0.1>	**Brown**

Table 3-2: Path table

After utilizing the IFS hierarchy presented in "Figure 3-4", this predicate can be reconstructed as follows:

$$\theta = \theta_1 \vee \theta_2 \vee \ldots \vee \theta_n$$

In the example at hand, θ_1="Red Wine", θ_2="Pinot" and θ_n="Muscat". The reconstructed predicate θ = (Red Wine $\vee$ Pinot $\vee$ Muscat) allows the query mechanism to not only definite answers, but also possible answers [84].

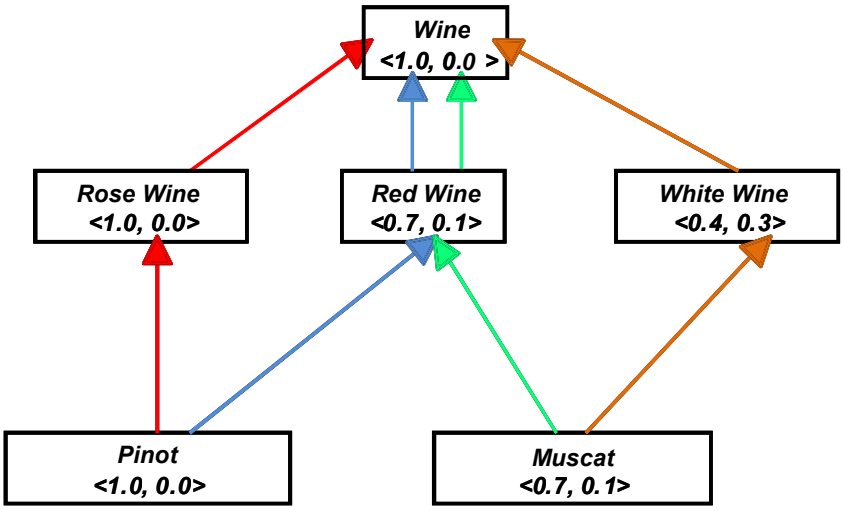


Figure 3-4: Pictorial representation of paths

In terms a query retrieving data from a summary table, the output contains not only records that match the initial condition, but also those that satisfy the reconstructed predicate. Consider the case where no records satisfy the initial selection condition (Red Wine). Traditional aggregation query would have returned no answer, however, based on the approach put forward in this thesis, the extended query would even in this case, return an answer, though

only a possible one, with a specific belief and disbelief $<\mu, \nu>$. It will point to those records that satisfy the reconstructed predicate θ, more specifically, “Pinot and Muscat”.

Following the representation of H-IFS as concept relations and the definition of summarisation paths, there is still a need to extend the traditional aggregation operators in order to cope with flexible hierarchies of data organisations, as part of the standard OLAP operators (*see Chapter 4*).

3.5 Generalised IFS Comparison to H-IFS

The concept of H-IFS was further extended in [87, 88] to allow the representation of hierarchical orderings between different universes.

Let E be a fixed universe and let A be an Intuitionistic Fuzzy Set IFS over E. Let F be another universe and let the set E be an IFS over F having the form:

$$E = \{\langle y, \mu_E(y), \nu_E(y)\rangle \mid y \in F\}$$

Therefore the element $X \in$ E has the form (*see [88]*):

$$\begin{gathered} x = \langle y, \mu_E(y), \nu_E(y)\rangle, \\ x \in F \times [0,1] \times [0,1], \\ A = \{\langle\langle y, \mu_E(y), \nu_E(y)\rangle, \mu_A(\langle y, \mu_E(y), \nu_E(y)\rangle), \\ \nu_A(\langle y, \mu_E(y), \nu_E(y)\rangle)\rangle \mid \langle y, \mu_E(y), \nu_E(y)\rangle \in E\} \end{gathered}$$

Let *A/E* stands for "A is an IFS over E". If the degrees of membership and non-membership of an element y to a set A in the frames of a universe E are $\mu_A(y)$ and $\nu_A(y)$ and the element $\langle y, \mu_A(y), \nu_A(y)\rangle$, has degrees of membership and non membership to the set E within the universe F are $\mu_E(y)$ and $\nu_E(y)$, then the following can be defined:

$$A = \{\langle y, \mu_E(y).\mu_A(y), \nu_E(y).\nu_A(y)\rangle \mid y \in F\}$$

When the universe is ordered, e.g., by relation ≤ the set A is called [79] an *"IFS over a universe with hierarchical structure (H-IFS)"*. In [88] generalised IFS over Hierarchical Universes IFS transform some ideas and results from [85], [86].

Based on [87] Let E be a finite universe defined as follows:

$$E = \{e_1, e_2, e_3, \{e_1, e_2\}, \{e_1, e_3\}, \{e_1, e_2, \{\{e_1, e_3\}\}\}$$

Therefore, the IFS A over E will have the form:

$$A = \{\langle e_1, \mu_A(e_1), \nu_A(e_1)\rangle, \langle e_2, \mu_A(e_2), \nu_A(e_2)\rangle, \langle e_3, \mu_A(e_3), \nu_A(e_3)\rangle,$$
$$\langle \{e_1, e_2\} \mu_A(\{e_1, e_2\}), \nu_A(\{e_1, e_2\})\rangle, \langle \{e_1, e_3\} \mu_A(\{e_1, e_3\}), \nu_A(\{e_1, e_3\})\rangle,$$
$$\langle \{e_1, e_2, \{e_1, e_3\} \mu_A(\{e_1, e_2, \{e_1, e_3\}), \nu_A(\{e_1, e_2, \{e_1, e_3\}\})\rangle\}$$

There is an order between some of the elements of E, e.g. for *i = 1, 2, 3*: e_i = i, this order is (≤ or <) but it cannot be extend over the rest E-elements. If the order is ⊂, it will be valid for fourth and sixth elements of E, but will not be possible for the rest E-elements.

Based on [87], for H-IFS E that has *n* levels and for every natural number $i \leq n$ one can introduce set $support_i$(E) that contains all E-elements that are from *i-th* level and that are not sets of elements of (*i+1*) level.

Let E be a finite or infinite set and let for each its element e: $\mu_A(e)$ and $\nu_A(e)$ exist. By analogy with [85] the set A/E/support(E) can be reconstructed, as shown below:

$$A/E[optimistic]\text{support(E)} = \{\langle \text{e}_{1,}, \mu_A(e_1), \nu_A(e_1)\rangle, \langle \text{e}_2, \mu_A(e_2), \nu_A(e_2)\rangle,$$
$$\langle \text{e}_{3,}, \mu_A(e_3), \nu_A(e_3)\rangle, \langle \{\text{e}_{1,}, e_2\}, \max(\mu_A(e_1), \mu_A(e_2)), \min(\nu_A(e_1), \nu_A(e_2))\rangle,$$
$$\langle \{\text{e}_{1,}, e_3\}, \max(\mu_A(e_1), \mu_A(e_3)), \min(\nu_A(e_1), \nu_A(e_3))\rangle,$$
$$\langle \{\text{e}_{1,}, e_2, \{\text{e}_{1,}, e_3\}\}, \max(\mu_A(e_1), \mu_A(e_2), \mu_A(e_3)), \min(\nu_A(e_1), \nu_A(e_2), \nu_A(e_3))\rangle\}$$

$$A/E[pessimistic]\text{support(E)} = \{\langle \text{e}_{1,}, \mu_A(e_1), \nu_A(e_1)\rangle, \langle \text{e}_2, \mu_A(e_2), \nu_A(e_2)\rangle,$$
$$\langle \text{e}_{3,}, \mu_A(e_3), \nu_A(e_3)\rangle, \langle \{\text{e}_{1,}, e_2\}, \min(\mu_A(e_1), \mu_A(e_2)), \max(\nu_A(e_1), \nu_A(e_2))\rangle,$$
$$\langle \{\text{e}_{1,}, e_3\}, \min(\mu_A(e_1), \mu_A(e_3)), \max(\nu_A(e_1), \nu_A(e_3))\rangle,$$
$$\langle \{\text{e}_{1,}, e_2, \{\text{e}_{1,}, e_3\}\}, \min(\mu_A(e_1), \mu_A(e_2), \mu_A(e_3)), \max(\nu_A(e_1), \nu_A(e_2), \nu_A(e_3))\rangle\}$$

$$A/E[average]\text{support(E)} = \{\langle \text{e}_{1,}, \mu_A(e_1), \nu_A(e_1)\rangle, \langle \text{e}_2, \mu_A(e_2), \nu_A(e_2)\rangle,$$
$$\langle \text{e}_{3,}, \mu_A(e_3), \nu_A(e_3)\rangle, \langle \{\text{e}_{1,}, e_2\}, \frac{\mu_A(e_1)+\mu_A(e_2)}{2}, \frac{\nu_A(e_1)+\nu_A(e_2)}{2}\rangle,$$
$$\langle \{\text{e}_{1,}, e_3\}, \frac{\mu_A(e_1)+\mu_A(e_3)}{2}, \frac{\nu_A(e_1)+\nu_A(e_3)}{2}\rangle, \langle \{\text{e}_{1,}, e_2, \{\text{e}_{1,}, e_3\}\},$$
$$\frac{\mu_A(e_1)+\mu_A(e_2)+\frac{\mu_A(e_1)+\mu_A(e_3)}{2}}{2}, \frac{\nu_A(e_1)+\nu_A(e_2)+\frac{\nu_A(e_1)+\nu_A(e_3)}{2}}{2}\rangle\}$$
$$= \{\langle \text{e}_{1,}, \mu_A(e_1), \nu_A(e_1)\rangle, \langle \text{e}_2, \mu_A(e_2), \nu_A(e_2)\rangle, \langle \text{e}_{3,}, \mu_A(e_3), \nu_A(e_3)\rangle,$$
$$\langle \{e_1, e_2\}, \frac{\mu_A(e_1)+\mu_A(e_2)}{2}, \frac{\nu_A(e_1)+\nu_A(e_2)}{2}\rangle,$$
$$\langle \{e_1, e_3\}, \frac{\mu_A(e_1)+\mu_A(e_3)}{2}, \frac{\nu_A(e_1)+\nu_A(e_3)}{2}\rangle, \langle \{\text{e}_{1,}, e_2, \{\text{e}_{1,}, e_3\}\},$$
$$\frac{3\mu_A(e_1)+\mu_A(e_2)+\frac{\mu_A(e_1)+\mu_A(e_3)}{2}}{2}, \frac{\nu_A(e_1)+\nu_A(e_2)+\frac{\nu_A(e_1)+\nu_A(e_3)}{2}}{2}\rangle\}$$

The following assertion is valid: for each IFS A over E:

$$\text{A/E[pessimistic]support(E)} \subset \text{A/E[average]support(E)} \subset$$
$$\text{A/E[optimistic]support(E)}$$

Generalised IFS offer a broader extension in terms of representing richer domain or user defined hierarchical preferences. However it has to be noted that Generalised IFS may be proven too demanding to be represented and handled as part of existing MOLAP tools, when it is known that some of the current MOLAP tools can handle only up to ten dimensions.

H-IFS are better suited for use in OLAP environments since queries are defined on top of "kind-of" hierarchies. Furthermore, queries utilize the concept of minimal H-IFS, ensuring thus successful query execution.

Generalises IFS are more appropriate for Mediator integration environments consisting of hierarchies of Mediators where there is a need [87] for more complex domain hierarchical preferences.

3.6 Conclusions

In this chapter, the concept of closure of an Intuitionistic fuzzy set over a universe that has a hierarchical structure was defined. Intuitively, in the closure of this Intuitionistic fuzzy set, the "kind-of" relation is taken into account by propagating the degree associated with an element to its sub-elements in the hierarchy. The automatic recommendation of analysis is introduced according to the concepts defined as part of domain description, in order to guide query answering with the aid of hierarchical Intuitionistic fuzzy sets.

The proposed methodology aims at expanding user preferences, expressed when defining a query, in order to obtain related and complementary answers. This is likely to be a useful tool for OLAP and knowledge discovery in, for example, data mediators or data-warehouses, where queries are expressing hierarchical perception-based user preferences.

In term of data representation, the new proposed structure would resemble a snowflake schema. However, there would be a significant change. The expanded axis of analysis may not be normalised dimensions, instead, every dimension would have linked to it the knowledge tables involved in their definition.

Accommodating imprecision or user preferences at the level of the dimensions or axis of analysis as part of a multidimensional model can be achieved with the aid of H-IFS. Overall, in terms of data representation, the following cases can be accommodated:

- Crisp dimensions and precise data
- H-IFS based hierarchies/dimensions and precise data

However, in order to have full representation of imprecision as part of multidimensional model, it should also allow the modelling of the following cases:

- Well defined hierarchies/dimensions and imprecise data
- H-IFS based hierarchies/dimensions and imprecise data

In order to achieve the accommodation of imprecision at the level of facts/data, it becomes clear that there is a need to extend the standard cubic model and the related OLAP operators. In Chapter four, a cubic model christened the IF-Cube is delivered. This allows the representation of imprecision at the data level. At the same time, the basic cubic operators are extended with the aid of Intuitionistic Fuzzy Logic. The delivery of the IF-Cube model in conjunction with the utilisation of the H-IFS caters for a complete treatment of imprecision, both at the level of dimensions and that of the data as part of an extended OLAP environment.

"Knowledge, a rude unprofitable mass, the mere materials with which wisdom builds, till smoothed and squared and fitted to its place, does but encumber whom it seems to enrich."

William Cowper

Chapter Four

4. KNOLAP – The IF-Cube and IF-Operators

Since the emergence of the OLAP technology, [43] different proposals have been made to give support to different types of data and application purposes. One of these is to extend the relational model (ROLAP) to support the structures and operations typical of OLAP. Further approaches [89, 90] were based on extended relational systems to represent data-cubes and operate over them. Another approach would be to develop new models using a multidimensional-cubic view of the data [62].

Nowadays, information and knowledge-based systems need to manage imprecision in the data, and more flexible structures are needed to represent the analysis domain. Models have been proposed for managing imprecision, as part of an incomplete data-cube [91], in the facts and the definition of facts using different levels in the dimensions [92].

Nevertheless, these models continue to use inflexible hierarchies, thus making it difficult to merge reconcilable data from different sources with some incompatibilities in their schemata. These incompatibilities arise due to different perceptions/views about a particular modelling reality.

In addressing the problem of representing flexible hierarchies, here is proposed a new multidimensional model that is able to deal with imprecision over conceptual hierarchies utilising the concept of H-IFS (*see Chapter 3)*.

The use of conceptual hierarchies or H-IFS enables one to:

- define the structures of a dimension in a more perceptive way to the final user, thus allowing a more perceptive use of the system.
- query information from different sources or even utilize domain preferences and enhance the description of hierarchies, thereby getting more knowledgeable query results. H-IFS is a unique way for incorporating "kind-of" relations, or conceptual hierarchies as part of a Knowledge based OLAP analysis (KNOLAP).

In the following sections, OLAP foundations are reviewed and a model aimed at resolving imprecision at the "Cube" or data level is proposed. The semantics of the Intuitionistic fuzzy cubic representation are introduced in contrast to the basic multidimensional-cubic structures. Overall, the introduced Intuitionistic Fuzzy cubic representation [82], [83] allows users to deal with imprecision not only at the level of dimensions with the aid of H-IFS but also at the level of facts or data. The basic cubic operators are extended and enhanced [85], [86], with the aid of Intuitionistic Fuzzy Logic.

4.1 Semantics of the IF-Cube vs. Crisp Cube

In this section the semantics of Multidimensional modelling and Intuitionistic Fuzzy Logic are reviewed, and based on these a unique concept named Intuitionistic Fuzzy Cube (IF-Cube) is proposed. The IF-Cube, in conjunction with the utilisation of H-IFS, allows users to model the following cases:

- Well defined hierarchies/dimensions and imprecise data
- H-IFS based hierarchies/dimensions and imprecise data

4.1.1 Overview of the Cube Model

A logical model that influences both the database design and the query engines is the *multidimensional-cubic* view of data in a warehouse. In a multidimensional data model, there is a set of *numeric measures* that are the objects of analysis. Examples of such measures are total sales, available budget, etc. Each of the numeric measures depends on a set of *dimensions,* which provide the context for the measure. The attributes of a dimension may be related via a hierarchy of relationships. In the above example, the product name is related to its category and the industry attribute through a hierarchical relationship, (*see "Figure 4-1"*).

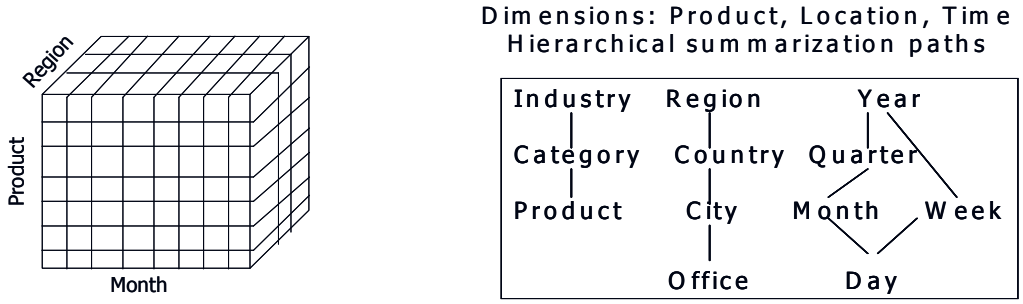


Figure 4-1: Cube 'Sales' - Rigid Hierarchies

A cubic structure [62] is defined as a 4-tuple <D, M, A, F> where the four components indicate the characteristics of the cube. These characteristics are:

- a set of n dimensions $D = \{d_1, d_2, \ldots, d_n\}$ where each d_i is a dimension name, extracted from a domain $dom_{dim(i)}$.
- a set of k measures $M = \{m_1, m_2, \ldots, m_k\}$ where each m_i is a measure name, extracted from a domain $dom_{measure(i)}$.
- The set of dimension names and measures names are disjoint; i.e. $D \cap M = 0$.
- A set of t attributes $A = \{a_1, a_2, \ldots, a_t\}$ where each a_i is an attribute name, extracted from a domain $dom_{attr(i)}$.
- A one-to-many mapping $F: D \rightarrow A$, i.e. there exists, corresponding to each dimension, a set of attributes.

4.1.2 Semantics of the IF-Cube

In contrast, an **IF-Cube** is an abstract structure that serves as the foundation for the multidimensional data cube model. Cube C is defined as a five-tuple *(D, l, F, O, H)* where:

- *D* is a set of dimensions
- *l* is a set of levels $l_1, \ldots, l_n$,
- A dimension $d_i = (l \leq O, l\perp, l_\top)\ dom(d_i)$ where $l = l_i\ i=1\ldots n$.
- l_i is a set of values and $l_i \cap l_j = \{\}$,
- $\leq O$ is a partial order between the elements of *l*.
- To identify the level *l* of a dimension, *dl* is used as part of a hierarchy.

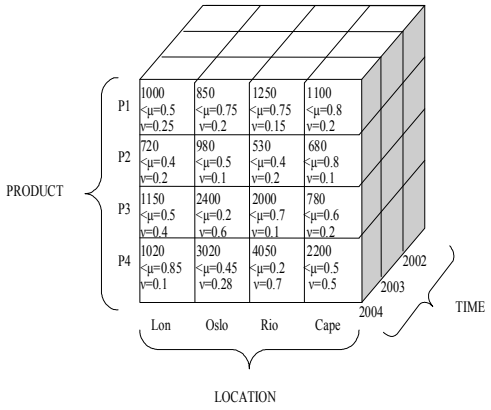
  $l\perp$: base level $l_\top$: top level

  for each pair of levels l_i and l_j there exist the relation:

  $$\mu_{ij} : l_i \times l_j \rightarrow [0,1] \quad v_{ij} : l_i \times l_j \rightarrow [0,1] \quad 0 < \mu_{ij} + v_{ij} < 1$$

- *F* is a set of fact instances with schema: $F = \{<x, \mu_F(x), v_F(x)> \mid x \in X\}$, where $x=<att_1, \ldots, att_n>$ is an ordered tuple belonging to a given universe *X*, $\mu_F(x)$ *and* $v_F(x)$ are the degree of membership and non-membership of *x* in the fact table *F* respectively.
- *H* is an object type history that corresponds to a cubic structure*(l, F, O, H')* which allows the tracing back the evolution of a cubic structure after performing a set of operators i.e. aggregation.

The example below provides a sample imprecise cube (*D, l, F, O, H) i.e. sales* and a conceptual flexible hierarchy product with reference to wine consisting of $l_i, \ldots, l_n$ levels with respective levels of membership and non membership $< \mu_{ij}\ v_{ij}, >$.



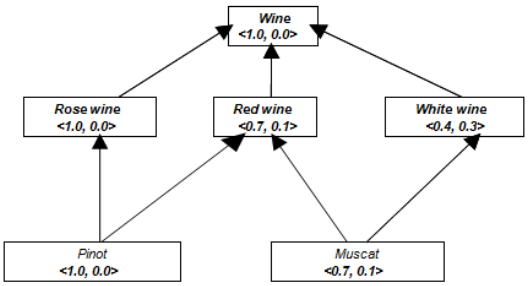


Figure 4-2: Imprecise cube 'Sales' ***Figure 4-3: H- IFS Hierarchy 'Wine'***

The defined IF OLAP Cube and the proposed OLAP operators make it possible to do the following:

- accommodate imprecise facts.
- utilize *conceptual hierarchies defined as H-IFS* used for aggregation purposes in the cases of roll-up and roll-down operations.
- offer a unique feature such as keeping track of the history when there is movement between different levels of a hierarchical order.

In the next section, first the current cubic operators are reviewed and then the IF-Operators are explained. These operators have been extended and redefined in order to cope with or multidimensional model.

4.2 IF-cubic operators vs. normal cubic operators

In the previous section, it was shown how the proposed IF-cube differs from the original cube and that it can be made to accommodate imprecision, both on the data level and on the conceptual level. However, the ability to store the data is only a small part of the problem. The difficulty stands with the ability to process such data, as the original cubical operators have not been designed to process imprecise information. In the subsections below, first the original operators will be shown and then the new IF-Operators presented, which will be able to deal with the new multidimensional structure.

4.2.1 Overview of the cubic operators

The cubic model proposed in [62], which is considered by many OLAP experts to be the fundamental one when it comes to the cubic model, also describes the algebraic operators necessary for the functioning of the multidimensional cube that have been adopted widely. Below is shown a brief description of these operators, the full descriptions of which can be found on [62].

***Restriction (σ)*:** This operator restricts the values on one or more dimensions. It has an *atomic predicate*, denoted by *p*, that is a logical expression involving a single dimension or a *compound*

predicate, denoted by *P* the is an expression involving a set of atomic predicates.

Mathematical notation: $\sigma_p(C_i)=C_o$

Example: $\sigma_{(year=2009)}$ *(Sales)*

Aggregation (α): This operator performs aggregation on one or more dimensions. This operator is based on relational aggregate functions (e.g. SUM AVG MAX) and allows these functions to be applied to cubes with one or more dimensions specified as *grouping attributes*.

Mathematical notation: $\alpha_{h,m,S}(C_I)=C_O$

Example: $\alpha_{[SUM(amount),\{product\_name,\ year\}]}(Sales)$

Cartesian product (×): This is a binary operator that can be used to relate two cubes.

Mathematical notation: $C_{I1} \times C_{I2} = C_O$

Join (|×|): The join operator is a special case of the *Cartesian product* operator that is used to relate two cubes having one or more dimensions in common and having identical mapping from the common dimensions to the respective attribute sets of these dimensions.

Mathematical notation:: $C_1 \ |\times|\ C_2 = \sigma p(C_1 \times C_2)$ where *p is the predicate* and $C_1$and C_2 are the two cubes.

Union (∪): This operator finds the union of two input cubes. If, for example, two cubes *Sales_Engand* and *Sales_Wales* contain the sales figures corresponding to the respective regions, and the user would like to consolidate the data for both regions into a single cube. This would be achieved by using the union operator.

Mathematical notation: $C_{I1} \cup C_{I2} = C_O$

Difference (-): This operator finds the difference of two cubes. If, for example, two cubes *Sales_England* and *Sales_London* contain sales figures corresponding to the England and London, and the user would wish to remove London figures from the England cube. This would be achieved using the difference operator.

Mathematical notation: $C_{I1} - C_{I2} = C_O$

4.2.2 The IF-cubic operators

In this section the IF-Cubic operators are defined and explained. Each operator is presented in the following format: the operator's name, symbol, textual description, input, output, mathematical description and an example of the operator.

Basic operators

Selection (Σ): The selection operator selects a set of fact-instances from a cubic structure that satisfy a predicate (θ). A predicate (θ) involves a set of atomic predicates (θ_1, …, θ_n) associated with the

aid of logical operators p (i.e. $\wedge$, $\vee$, etc.). Only the cells that satisfy the predicate p are captured into the result cube. If θ' is an Intuitionistic fuzzy predicate, then the set of possible facts that satisfy the θ should carry a degree of membership μ and non-membership v expressed as follows:

$$F = \{<x, \min(\mu_F(x), \mu(\theta(x))), \max(v_F(x), v(\theta(x)))> \mid x \in X\}$$

Thus the resulting cube populated with fact instances that either satisfy the predicate (θ) completely or to some degree of certainty. Where $\pi = 1 - (\mu + v)$ and acts as an index of the uncertainty, i.e. the higher the value of π, the more uncertain the fact instance is, even though it may entail the same level of membership μ.

Input: $C_i = (D, l, F, O, H)$ and the predicate θ.

Output: $C_o = (D, l, F_o, O, H)$, where

$F_o \subseteq F$ and $F_o = \{f \mid (f \in F) \wedge (f \text{ satisfies } \theta)\}$.

Mathematical notation: $\sum_{\theta}(C_i) = C_o$.

Example: Find the sales amount of 1000 with membership of greater than 0.4 and non-membership of less than 0.3 for all products in all cities during 2004:

$$\Sigma_{(\text{amount}=1000 \wedge (\mu>0.4 \ \wedge \ v<0.3) \ \wedge \ \text{year}=2004)}(\text{Sales}) = C_{\text{Result}}$$

Cubic Projection (Π): In cubic instances that hold non-deterministic facts, there can be no projecting-out of any of individual domains. The reason behind this statement is that unlike deterministic cubes, in non-deterministic ones the membership and

non-membership of a fact instance determines the likelihood of all domains involved in that cube/fact instance. Hence, projecting out a domain, would result in loss of information.

Input: $C_i = (D, l, F, O, H)$.

Output: $C_o = (D, l, F, O, H)$.

Mathematical notation: $\Pi_F (C_i) = C_o$.

Example: Project the cube from the previous example:

$$\Pi_{(\text{Sales})} (\Sigma_{(\text{amount}=1000 \wedge (\mu=0.4 \wedge \nu=0.3) \wedge \text{year}=2004)}(\text{Sales}))=C_{\text{Result}}$$

Basic Cubic Product (⊗): This is a binary operator $C_{i1} \otimes C_{i2}$. It is used to relate two cubes C_{i1} and C_{i2} assuming that $D_1 \subseteq D_2$ and O_1, O_2 are reconcilable partial orders. Thus, l_1, l_2 could lead to l_o being a ragged hierarchy.

Input: $C_{i1} = (D_1, l_1, F_1, O_1, H_1)$ and $C_{i2} = (D_2, l_2, F_2, O_2, H_2)$.

Output: $C_o = (D_o, l_o, F_o, O_o, H_o)$, where

$D_o = D_1 \cup D_2$, $l_o = l_1 \cup l_2$, $O_o = O_1 \cup O_2$, $H_o = H_1 \cup H_2$,

$F_o = F_1 \times F_2 =$

$= \{<<x, y>, \min(\mu_{f1}(x), \mu_{f2}(y)), \max(\nu_{f1}(x), \nu_{f2}(y))> | <x, y> \in X \times Y\}$.

Mathematical notation: $C_{i1} \otimes C_{i2} = C_o$.

Example: Consider the two cubes one wants to relate,
C_{i1}: C_{Sales} and C_{i2}: $C_{\text{Discounts}}$.

$C_{\text{Discounts}}$ has the same dimensions as C_{Sales} except the measure amount is not sale but is a discount. In that case the cubic product of these two, would be:

$$C_{\text{Sales}} \otimes C_{\text{Discounts}} = C_{\text{Result}}$$

ProdID	StoreID	Amount	<μ, ν>
P1	S1	10	0.7, 0.2
P2	S2	15	0.5, 0.5

⊗

ProdID	StoreID	Discount	<μ, ν>
P2	S1	2	0.5, 0.5
P3	S3	5	0.3, 0.3

⇓

S.Prod ID	S.Store ID	S.Amount	D.Prod ID	D.Store ID	Discount	<μ, ν>
P1	S1	10	P2	S1	2	0.5, 0.5
P1	S1	10	P3	S3	5	0.3, 0.3
P2	S2	15	P2	S1	2	0.5, 0.5
P2	S2	15	P3	S3	5	0.3, 0.5

Table 4-1: Cubic product

Union (∪): The union operator is a binary operator that finds the union of two cubes. C_{i1} and C_{i2} have to be union compatible. The operator also coalesces the value-equivalent facts using the minimum membership and maximum non-membership.

Input: $C_{i1} = (D_1, l_1, F_1, O_1, H_1)$ and $C_{i2} = (D_2, l_2, F_2, O_2, H_2)$.

Output: $C_o = (D_o, l_o, F_o, O_o, H_o)$, where

$D_o = D_1 = D_2$, $l_o = l_1 = l_2$,

$O_o = O_1 = O_2$, $H_o = H_1 = H_2$,

$F_o = F_1 \cup F_2 =$

$= \{<x, \max(\mu F_1(x), \mu F_2(x)), \min(\nu F_1(x), \nu F_2(x))> \mid x \in X\}$.

Mathematical notation: $C_{i1} \cup C_{i2} = C_o$.

Example: Consider the two cubes one want to relate,

C_{i1}: $C_{\text{Sales\_North}}$ and C_{i2}: $C_{\text{Sales\_South}}$,

in that case the union of these two cubes would be:

$$C_{\text{Sales\_North}} \cup C_{\text{Sales\_South}} = C_{\text{Result}}$$

ProdID	StoreID	Amount	<μ, ν>
P1	S1	10	0.7, 0.2
P2	S2	15	0.5, 0.5

∪

ProdID	StoreID	Amount	<μ, ν>
P1	S1	10	0.5, 0.5
P3	S3	5	0.3, 0.3

⇓

S.ProdID	S.Store ID	S.Amount	<μ, ν>
P1	S1	10	0.7, 0.2
P2	S2	15	0.5, 0.5
P1	S1	10	0.5, 0.5
P3	S3	5	0.3, 0.3

⇓

S.ProdID	S.StoreID	S.Amount	<μ, ν>
P1	S1	10	0.7, 0.2
P2	S2	15	0.5, 0.5
P3	S3	5	0.3, 0.3

Table 4-2: Union operator example

Difference (-): The difference operator is a binary operator that the difference of two cubes. It is similar to the difference operator in relational algebra. C_{i1} and C_{i2} have to be union compatible. The difference operator removes the portion of the cube C_{i1} that is common to both cubes.

Input: $C_{i1} = (D_1, l_1, F_1, O_1, H_1)$ and $C_{i2} = (D_2, l_2, F_2, O_2, H_2)$.

Output: $C_o = (D_o, l_o, F_o, O_o, H_o)$, where

$D_o = D_1 = D_2$, $l_o = l_1 = l_2$, $O_o = O_1 = O_2$,

$H_o = H_1 = H_2$,

$F_o = F_1 \cap F_2 = \{<x, \min(\mu F_1(x), \mu F_2(x)), \max(\nu F_1(x), \nu F_2(x))> \mid x \in X\}$.

Mathematical notation: $C_{i1} - C_{i2} = C_o$.

Example: Consider the two cubes one wants to relate,

C_{i1}: $C_{\text{Sales\_North}}$ and C_{i2}: $C_{\text{Sales\_South}}$,

in that case the difference between North and South sale cubes would be:

$$C_{\text{Sales\_North}} - C_{\text{Sales\_South}} = C_{\text{Result}}$$

ProdID	StoreID	Amount	<μ, ν>
P1	S1	10	0.7, 0.2
P2	S2	15	0.5, 0.5

–

ProdID	StoreID	Amount	<μ, ν>
P1	S1	10	0.5, 0.5
P3	S3	5	0.3, 0.3

⇓

S.ProdID	S.StoreID	S.Amount	<μ, ν>
P1	S1	10	0.7, 0.2
P2	S2	15	0.5, 0.5
P1	S1	10	0.5, 0.5

⇓

S.ProdID	S.StoreID	S.Amount	<μ, ν>
P1	S1	10	0.5, 0.5
P2	S2	15	0.5, 0.5

Table 4-3: Difference operator example

Extended Operators

Join (Θ): The join operator relates two cubes having one or more dimensions in common, and having identical mappings from common dimensions to the respective attribute sets of these dimensions. This operation can be expressed using Cubic Product operation.

$C_{i1} = (D_1, l_1, F_1, O_1, H_1)$ and $C_{i2} = (D_2, l_2, F_2, O_2, H_2)$ are candidates to join if $D_1 \cap D_2 \neq 0$.

Input: $C_{i1} = (D_1, l_1, F_1, O_1, H_1)$ and $C_{i2} = (D_2, l_2, F_2, O_2, H_2)$.

Output: $C_o = (D_o, l_o, F_o, O_o, H_o)$.

Mathematical notation: $C_{i1} \Theta C_{i2} = \sigma_p(C_{i1} \otimes C_{i2})$.

Example: Consider the two cubes one wants to relate, C_{i1}: C_{Sales} and C_{i2}: $C_{\text{Discounts}}$. $C_{\text{Discounts}}$ has the same dimensions as C_{Sales} except the measure amount is not sale but is a discount.
Also there is a predicate p= ($S.ProdID = D.ProdID \wedge S.StoreID = D.StoreID$). In that case the join of these two, would be:

$$C_{\text{Sales}} \Theta C_{\text{Discounts}} = C_{\text{Result}}.$$

ProdID	StoreID	Amount	<μ, ν>
P1	S1	10	0.7, 0.2
P2	S2	15	0.5, 0.5

Θ

ProdID	StoreID	Discount	<μ, ν>
P1	S1	2	0.5, 0.5
P3	S3	5	0.3, 0.3

⇓

S.ProdID	S.StoreID	S.Amount	D.Discount	<μ, ν>
P1	S1	10	2	0.5, 0.5

Table 4-4: Join operator example

Aggregation (A): The aggregation operator performs aggregation on one or more dimensional attributes utilizing Intuitionistic Fuzzy functions such as IFS_{SUM}, IFS_{AVG}, IFS_{MIN}, IFS_{MAX}. An aggregation operator A is a function $A(G)$, where $G = \{<x, \mu_F(x), \nu_F(x)> | x \in X\}$, where $x=<att_1, \ldots, att_n>$ is an ordered tuple, belonging to a given universe X, $\{att_1, \ldots, att_n\}$ is the set of attributes of the elements of X, $\mu_F(x)$ and $\nu_F(x)$ are the degree of membership and non-membership of x.

The result is a bag of the type $\{<x', \mu_F(x'), \nu_F(x')> \mid x' \in X\}$. To this extent, the bag is a group of elements that can be duplicated and each one has a degree of μ and ν.

Input: $C_i = (D, l, F, O, H)$ and the function $A(G)$.
Output: $C_o = (D, l_o, F_o, O_o, H_o)$.

The definition of aggregation operator points to the need of defining the IFS extensions for traditional group operators such as SUM, AVG, MIN and MAX.

Group Operations & Operators

In this section an investigation is made on how traditional group operations can be redefined to cope with the IFS representation of data. Note that the introduction of the IF facts influence the evaluation of aggregates at different levels:

- Will the result over which the aggregate is performed be either crisp or Intuitionistic Fuzzy?
- What is the meaning of the result after the IF aggregation is performed?

Using the standard definitions for the group operators (SUM, AVG, MIN and MAX) as foundations, their IF extensions and meaning is provided.

***IFS*$_{SUM}$:** The IFS_{sum} aggregate, like its standard counterpart, is only defined for numeric domains. Given a fact F defined on the schema $X(att_1, \ldots, att_n)$, let $att_{n\text{-}1}$ defined on the domain $U=\{u_1, \ldots, u_n\}$. The fact F consists of fact instances f_i *with* $1 \le i \le m$. The fact instances f_i are assumed to take Intuitionistic fuzzy values for the attribute $att_{n\text{-}1}$ for $i = 1$ to m

$$f_i\,[att_{n\text{-}1}] = \{<\mu_i(u_{ki}),\ v_i(u_{ki})>/\ u_{ki} \mid 1 \le k_i \le n\}.$$

The IFS_{sum} of the attribute $att_{n\text{-}1}$ of the fact table F is defined by:

$$IFS_{SUM}((att_{n-1})(F)) =$$

$$= \left\{ < u > / y \mid ((u = \min_{i=1}^{m}(\mu_i(u_{ki}), \nu_i(u_{ki})) \wedge \left(y = \sum_{k_i = k_1}^{km} u_{ki} \right) (\forall k_1, \ldots, k_m : 1 \le k_1, \ldots, k_m \le n)) \right\}$$

Example: $IFS_{SUM}((Amount)(ProdID))$

= {<0.8, 0.1>/ 100} + {(<0.4, 0.2>/110), (<0.3, 0.2>/120)} +
+{(<0.5, 0.3>/130), (<0.5, 0.1>/120)}
={(<0.8∧0.4,0.1∧0.2>/100+110),(<0.8∧0.3,0.1∧0.2>/100+120)}+
+{<0.5, 0.3>/130, <0.5, 0.1>/120}
={(<0.4, 0.2>/210),(<0.3, 0.2>/220)}+{<0.5, 0.3>/130,
<0.5, 0.1>/120}
={(<0.4∧0.5,0.2∧0.3>/210+130),(<0.4∧0.5,0.2∧0.1>/210+120),
(<0.3∧0.5,0.2∧0.3>/220+130),(<0.3∧0.5,0.2∧0.1>/220+120)
={(<0.4, 0.3>/340), (<0.4, 0.2>/330), (<0.3, 0.3>/350),
(<0.3, 0.2>/340)}
={(<0.3, 0.3>/340), (<0.4, 0.2>/330), (<0.3, 0.3>/350)}.

IFS$_{AVG}$: The IFS_{AVG} aggregate, like its standard counterpart, is only defined for numeric domains. This aggregate makes use of the IFS_{SUM} that was discussed previously and the standard *COUNT*. The IFS_{AVG} can be defined as:

$$IFS_{AVG}((att_{n-1})(F) = IFS_{SUM}((att_{n-1})(F)) \,/\, COUNT((att_{n-1})(F)).$$

Example: $IFS_{AVG}((Amount)(ProdID))$
$=IFS_{SUM}((Amount)(ProdID))/\ COUNT((Amount)(ProdID))$
$=\{(<0.3, 0.3>/340), (<0.4, 0.2>/330), (<0.3, 0.3>/350)\}/3$
$=\{(<0.3, 0.3>/113.33), (<0.4, 0.2>/110), (<0.3, 0.3>/\ 116.66)\}.$

IFS$_{MAX}$: The IFS_{MAX} aggregate, like its standard counterpart, is only defined for numeric domains. Given a fact F defined on the schema $X(att_1, \ldots, att_n)$, let at_{n-1} defined on the domain $U=\{u_1, \ldots, u_n\}$. The fact F consists of fact instances f_i *with* $1 \le i \le m$. The fact instances f_i are assumed to take intuitionistic fuzzy values for the attribute att_{n-1} for $i = 1$ to m

$$f_i[att_{n-1}] = \{<\mu_i(u_{ki}),\ \nu_i(u_{ki})>/\ u_{ki} \mid 1 \le k_i \le n\}.$$

The IFS_{sum} of the attribute att_{n-1} of the fact table F is defined by:

$$IFS_{MAX}((att_{n-1})(F)) +$$
$$= \left\{< u > / y \mid ((u = \min_{i=1}^{m}(\mu_i(u_{ki}), \nu_i(u_{ki})) \wedge \left(y = \max_{i=1}^{m}(\mu_i(u_{ki}), \nu_i(u_{ki}))\right)(\forall k_1, \ldots, k_m : 1 \le k_1, \ldots, k_m \le n))\right\}$$

Example: $IFS_{MAX}((Amount)(ProdID))$

IFS_{MAX} =
={<0.8, 0.1>/100}, {(<0.4, 0.2>/110), (<0.3, 0.2>/120)},
{(<0.5, 0.3>/130), (<0.5, 0.1>/120)};
= {(<0.8∧0.4, 0.1∧0.2>/max(100,110)),
(<0.8∧0.3,0.1∧0.2>/max(100,120)},
{<0.5, 0.3>/130, <0.5, 0.1>/120};
= {(<0.4, 0.2>/110), (<0.3, 0.2>/120)}, {<0.5, 0.3>/130,
<0.5, 0.1>/120};
= {(<0.4∧0.5, 0.2∧0.3>/max(110,130)),
(<0.4∧0.5, 0.2∧0.1>/max(110,120)),
(<0.3∧0.5,0.2∧0.3>/max(120,130)),
(<0.3∧0.5,0.2∧0.1>/max(120, 120))
= {(<0.4, 0.3>/130), (<0.4, 0.2>/120), (<0.3, 0.3>/130),
(<0.3, 0.2>/120)}
= {(<0.3, 0.3>/130), (<0.3, 0.2>/120)}.

IFS_{MIN}: The IFS_{MIN} aggregate, like its standard counterpart, is only defined for numeric domains. Given a fact F defined on the schema $X(att_1, \ldots, att_n)$, let $att_{n\text{-}1}$ defined on the domain $U=\{u_1, \ldots, u_n)$. The fact F consists of fact instances f_i *with* $1 \leq i \leq m$. The fact instances f_i are assumed to take intuitionistic fuzzy values for the attribute $att_{n\text{-}1}$ for $i = 1$ to m therefore $f_i[att_{n\text{-}1}] = \{<\mu_i(u_{ki}), \nu_i(u_{ki})>/ u_{ki} \mid 1 \leq k_i \leq n\}$. The IFS_{sum} of the attribute $att_{n\text{-}1}$ of the fact table F is defined by:

$$IFS_{MIN}((att_{n-1})(F)) =$$
$$= \left\{ < u > / y \mid ((u = \min_{i=1}^{m}(\mu_i(u_{ki}), \nu_i(u_{ki})) \wedge \left(y = \min_{i=1}^{m}(\mu_i(u_{ki}), \nu_i(u_{ki}))\right)(\forall k_1,\ldots,k_m : 1 \leq k_1,\ldots,k_m \leq n)) \right\}$$

It can be observed that the IFS_{MIN} is extended in the same manner as IFS_{MAX} aggregate except for replacing the symbol **max** in the IFS_{MAX} definition with **min**.

The definition of the extended group operations makes it possible to define the extended group operators ***Roll up (Δ), and Roll Down (Ω).***

Roll up (Δ): The result of applying *Roll up* over dimension d_i at level dl_r *using the* aggregation operator A over a datacube $C_i = (D_i, l_i, F_i, O, H_i)$ *is another datacube* $C_o = (D_o, l_o, F_o, O, H_o)$.

Input: $C_i = (D_i, l_i, F_i, O, H_i)$.
Output: $C_o = (D_o, l_o, F_o, O, H_o)$.

An object of type history is a recursive structure:

$$H = \begin{cases} \omega \text{ is the initial state of the cube} \\ (l, D, A, H') \text{ is the state of the cube after performing an operation on the cube} \end{cases}$$

The structured history of the datacube allows the storing of all the information when applying *Roll up* and the recall of it back when *Roll Down* is performed. In order to be able to apply the operation of *Roll Up* the IFS_{SUM} aggregation operator needs to be put to use.

Roll Down (Ω): This operator performs the opposite function of the *Roll Up* operator. It is used to roll down from the higher levels of the hierarchy with a greater degree of generalization, to the leaves with the greater degree of precision. The result of applying *Roll Down* over a datacube $C_i = (D, l, F, O, H)$ having $H = (l', D', A', H')$ is another datacube $C_o = (D', l', F', O, H')$.

Input: $C_i = (D, l, F, O, H)$.
Output: $C_o = (D', l', F', O, H')$ where $F' \rightarrow$ *set of fact instances defined by operator A.*

To this extent, the *Roll Down* operative makes use of the recursive history structure previously created after performing the *Roll Up* operator.

4.3 Conclusions

In this chapter the context of value imprecision was revised, as part of an MOLAP based environment. A new approach for extending the MOLAP model was presented, so that it can include treatment of value uncertainty as part of a multidimensional model, inhabited by concepts and flexible hierarchical structures of organization. A new multidimensional-cubic model named the IF-Cube was introduced, which is able to operate over data with imprecision either in the facts or in the dimensional hierarchies.

The main contribution of this new multidimensional-cubic model is that is able to operate over data with imprecision in the facts and the summarisation hierarchies. Classical models imposed a rigid structure that made the models present difficulties when merging information from different but still reconcilable sources.

These features are inexistent in current OLAP tools. Furthermore, it has been noticed that the IF-Cube can be used for the representation of Intuitionistic fuzzy linguistic terms.

In order to further clarify the structure of the model presented here, chapter five will describe the metadata of the model and also introduce a case study, the *Vitis Vinifera* (the ontology of the common grape vine) domain. This case of multidimensional modelling is put through an H-IFS based ad-hoc utility christened "IF-Oracle".

"Burgundy for Kings, Champagne for Duchesses, and claret for Gentlemen."

French Proverb

Chapter Five

5. Implementing the KNOLAP architecture

5.1 Conceptual Modelling

An approach is presented towards the problem of encoding value and imprecision in OLAP systems. The nature of value imprecision is specified and a conceptual model is built that can capture the semantics of value imprecision.

This section presents the properties for a conceptual formalism that will capture value imprecision. A conceptual model should present value imprecision at three levels of abstraction, which are:

- The instance level, which contains the actual facts that will populate the data repository.
- The specification level which still is application dependent but of a higher level; it describes classes of objects rather than individual data items.
- The concept level or meta-model level, where the basic concepts and their inter-relationship is described. The meta-model level is application independent.

A conceptual model for representing value imprecision must perceive the following properties:

- **Powerful, Simple and Formal:** How much expressive power a conceptual model should support? High expressive power may lead to complexity and eventually to rejection of the model. Simplicity implies that the number of concepts should be kept small. Further on, the definitions of the concepts (primitive notions) should be as close as possible to the real

world concepts. For specifications to be ambiguous they have to rely on a sound formal definition. This is the point where meta-modelling is useful.

With regards to the multidimensional property of imprecision, two more features can be added, named belief and knowledge representation. However, it has to be emphasised that knowledge representation is related to the instance level only;

- **Belief:** is expressed formally as the measure related with the might happen ability or tendency of things to occur. In this thesis belief indicates possible ways to interpret and map the semantics of the kind-of relation (*see chapter 3*).
- **Knowledge Representation:** The issue here is whether the model can represent the "kind-of" relations as summarisation paths that will allow users to define their own axes of analysis when it comes to OLAP analysis (*see chapter 4*).

At this point the first considerations for a conceptual formalism will be defined that will enable a common framework for value imprecision in OLAP based environments or tools.

5.2 Meta-model of the KNOLAP approach

Modelling for OLAP could be viewed [93] from a conceptual and a logical perspective. To this extent, there is no accepted conceptual modelling language for OLAP semantics. Instead, the debate is dominated by a broad variety of models for multidimensional structures [93]. Actually, no in-depth analysis has been carried out which relates syntax and semantics of the multidimensional query languages regarding the underlying universe of discourse.

Multidimensional models consist of quantifying and qualifying data. The former, often referred to as measures, represent values of relevant objects of an application domain (e.g. sales/turnover). Measures are qualified through dimensions, describing the selected viewpoints (e.g. time, region, and customer), leading to concrete information. Combining quantifying and qualifying data results in a cube-fact, which represents both.

Dimensions consist of dimensional nodes which are regularly organised into hierarchies, following the mechanism of defining set memberships. A hierarchy results from creating sets of elements. The grouping of elements or sets should be guided by some type of set definition. Hierarchies model granular elements which are on the same level of the hierarchy. A repetitive application of the grouping leads to three possible types of hierarchical levels. First, a level could contain all elements. As it contains all available elements, this level is called elementary level. Second, a level root could contain only one set which all other elements are subordinated to.

And last, there can be intermediate levels, residing in-between. Hierarchies enable the changing of the level of detail of a business object that is being represented, adapting views according to the actual information requirements.

Then, it has to be considered whether the concepts of the business process/facts may benefit from H-IFS labelling. An Intuitionistic fuzzy dimension is actually an extension of a crisp dimension. For each crisp dimensional instance, Intuitionistic Fuzzy weights are associated with the aid of H-IFS closure (*see section 3.3.2*). For example, consider the wine dimension. A typical wine dimension may have the product level, the category level, and the subcategory level. In many cases, a wine subcategory can belong to several product categories. For example, Muscat can be considered either red or white wine. Therefore, the user can associate the "wine" product subcategory to several product categories and assign different membership, non-membership and hesitation degrees to each wine category.

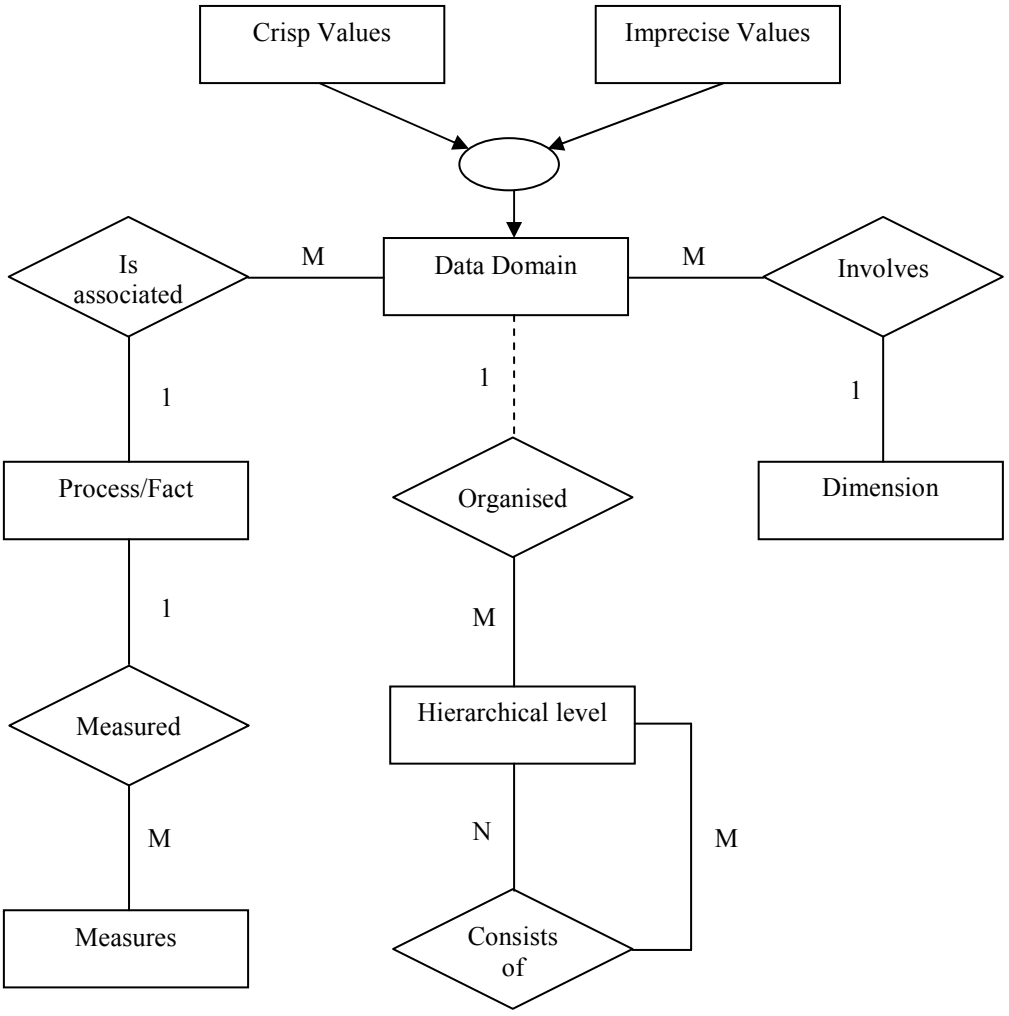


Figure 5-1: KNOLAP Meta-model

The meaning behind the Intuitionistic fuzzy hierarchy is that the child and parent levels will have a many-to-many relationship as seen in “Figure 5-1”. This kind-of relationship is not common in the classical DW but [43] did mention the feasibility of such a relationship in a DW. The solution for this problem is adding a "summarisation paths" object between the parent and child levels. The summarisation table holds combinations of the parent and the

child level and, in addition, the different membership, non-membership and hesitations degrees are added between the different level items. The specification of an Intuitionistic fuzzy hierarchy is based on the utilisation of the minimal H-IFS (*see section 3.3*).

In the next sections the proposed KNOLAP meta-model and its architecture are utilized for the purposes of analysis of the Vitis Vinifera domain and for implementing an ad-hoc utility called ”IF-Oracle” that allows expression of value imprecision at the level of domain hierarchies.

With respect to the KNOLAP architecture as presented in “Figure 5-2”, OLAP query execution occurs as follows:

- The user provides an OLAP query to be answered and a set of dimensions related to the submitted query. Dimensions are defined according to user’s perceptions given a set of entities involved in the analysis.

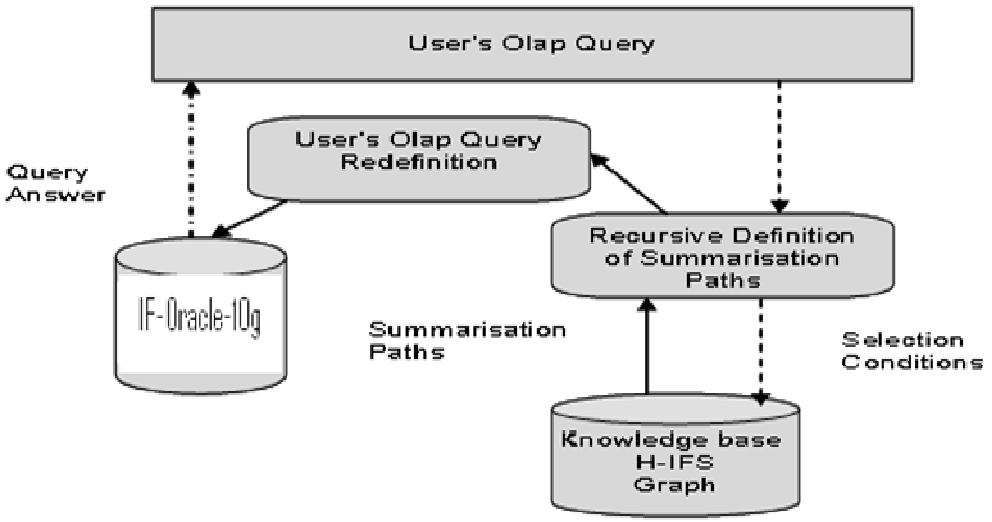


Figure 5-2: KNOLAP architecture

- Conditions and grouping sets in the OLAP query are transformed into H-IFS generalized conditions and grouping sets. H-IFS generalized condition point to possible summarization paths involved in the definition of a dimension.
- Summarization paths express hierarchical weighted knowledge (H-IFS) involved in the definition of a dimension and thus in the query conditions and grouping statements.
- Generalized H-IFS conditions and grouping statements as part of an OLAP query are submitted as weighted conjunctive queries to the underlying OLAP database server.
- As soon as the query execution is over, the user will receive an answer to its initial query that consists of two parts; the definite and the possible. The definite part identifies the subgroup or data-cube that satisfies exactly the given request, while the possible part identifies an enhanced subgroup or data-cube that allows users to extract results previously unknown to them due to the enhancement of the query conditions and grouping statements.

5.3 Introduction of the Vitis Vinifera Domain

The Vitis Vinifera domain is a case of multidimensional modelling, according to Multidimensional paradigm [43]. Further analysis of the Vitis Vinifera domain will require operations to aggregate based on levels of aggregation alternatively known as dimension hierarchies. So, improving query answers process involves well defined and rich hierarchies [94]. Then the main task is on addressing the following question/issue, "How to define dimension hierarchies"?

There are several possible approaches in developing a hierarchy:
A top-down development process starts with the definition of the most general concepts in the domain and subsequent specialization of the concepts.
A bottom-up development process starts with the definition of the most specific elements, the leaves of the hierarchy, with subsequent grouping of these classes into more general concepts.

A combination development process is a combination of the top-down and bottom-up approaches: first the more significant concepts are defined and then they are generalised or specialised appropriately. To start with, a few top-level concepts such as Wine, and a few specific concepts, such as Syrah are considered. Then they are related to a middle-level concept, such as Rhone.

This is not a simple task for the following reasons:

- Hierarchies could not be specified as many terms and data required by users are not included in the operational sources, i.e. consider a wine-sales database.

- Some kind of guidance is needed to enrich hierarchies by adding levels of aggregation, when referring to complex modelling domains like Vitis Vinifera.
- There is a need to include knowledge provided by the Vitis Vinifera domain in order to improve the quality of dimension hierarchies. This will allow the inclusion of new hierarchy aggregation levels, which in return will allow DWH's (Data-Warehouse – a collection of integrated multisource data) users to achieve their analysis information needs and better support the query answering process.

5.3.1 A proposed solution

Automatically complete hierarchies using relationships among concepts provided by an H-IFS [94] for the following reasons:

- Dimension hierarchies represent semantic relations between values. i.e. Red Bordeaux are Red wines.
- H-IFS can express generalisation of two important properties: "is-a-kind-of" and aggregation or "is-a-part-of". For example, Cabernet Sauvignon, Cabernet Franc and Merlot are kind-of Bordeaux Grapes and they are part of Red Bordeaux wines.
- These semantic relations allow us to organize concepts into hierarchical structures. The "kind-of" and "is part of" relations between concepts are of interest here, as they are the most useful relationships in a dimension hierarchy, and could be used to extend dimension hierarchies.
- In this context, H-IFS, which are more generally used to represent concepts whose borders are not strictly delimited, can be used to define flexible selection criteria, by

associating a preference with every candidate value. The hierarchical structure can be used to enhance the users' queries in case of empty answers, while respecting the preference order defined by the users in their selection criteria.

In the Vitis Vinifera domain, the following are possible competency questions:

- Is Muscat a Red or White wine?
- Is Bordeaux a Red or White wine?
- Which wines, red or white are increasing in popularity?

In providing an answer to these questions/queries one has to recognise that off-the-shelf products cannot answer the above questions simply because hierarchies provide only levels of summarisation but not any knowledge about the domain. On the other hand, H-IFS provide an ontological view of the modelled domain as well as efficient ways of summarising operational data as part of data-warehouse.

The H-IFS structure for Vitis Vinifera Domain has been constructed as follows: applying elementary generalisation of the initial set of an H-IFS structure into H-IFS of extended structure to create a new hierarchy level. The process is repeated until the required level of aggregation is achieved.

Thus, the first elementary generalisation for the H-IFS structure for Vitis Vinifera domain starts at level n-3. This elementary generalisation allows us to relate grape varieties as "kind-of" regional wine types. This corresponds to the level n-2 of the H-IFS. The second elementary generalisation starts at level n-2. This consecutive elementary generalisation allows us to relate regional

wine types as kind-of the general type wines (Black, Rose, White, Red). This corresponds to the level n-1 of the H-IFS. If one wishes to further generalise the n-1 H-IFS, then the next elementary generalisation, level n, will produce the whole production for all types of wines.

The next section presents an OLAP querying mechanism that utilizes knowledge in the form of H-IFS for the Vitis Vinifera domain.

5.4 Querying the Vitis Vinifera Domain

Let's consider a sample multidimensional model, depicted in "Figure 5-3" in the form of a star schema that describes sales of the Vitis Vinifera type wines.

Considering the Wine Sales star schema and the product dimension, the attribute H-Name corresponds to "Figure 5-5", H-IFS structure for Vitis Vinifera domain. So far no fuzziness with respect to data displayed in "Figure 5-3".

Product	Price	*Name*
	€ 50.00	Red Bordeaux
	€ 20.00	Medit.Muscat
	€ 45.00	Merlot
	€ 50.00	Sauvignon
	€ 51.00	Friuli
	€ 52.00	White Bordeux
	€ 48.00	Chateau d'Yquem

Store	*Store-Id*	*City*
	C1	Rome
	C2	Paris
	C3	Moscow

Sale	*Sale-Id*	*Bottle-Id*	*Store-Id*	*Quantity*	*Date*
	S1	Red Bordeaux	C1	20	09-Dec-99
	S2	Medit.Muscat	C1	14	09-Dec-99
	S2	Medit.Muscat	C1	16	09-Dec-99
	S3	Merlot	C1	40	09-Dec-99
	S4	Merlot	C2	100	09-Dec-99
	S5	Sauvignon	C2	120	09-Dec-99
	S5	Sauvignon	C2	80	09-Dec-99
	S6	Friuli	C2	200	09-Dec-99
	S7	White Bordeux	C3	600	12-Dec-07
	S8	Merlot	C3	1000	12-Dec-07
	S9	Medit.Muscat	C3	440	12-Dec-07
	S9	Medit.Muscat	C3	360	12-Dec-07
	S10	Chateau d'Yquem	C3	400	13-Dec-07

Figure 5-3: Sample of a Star schema

At the same time let's recall the main focus of the multi-dimensional approach which is the subject area that is most

important for analysis in this case sales of bottled wines. To this extent let us consider the following questions:

- Which wines, red or white are increasing in popularity?
- Is Muscat a Red or White wine?

Traditional OLAP tools like Oracle Express, etc. are currently not capable of answering this query for the following reason:

- By observing the Name attribute in "Figure 5-3", it can be seen that there are no direct matches for red or white wine. So a traditional OLAP query will return no answers for question I. Similarly, question II cannot be answered by traditional OLAP tools because Muscat type wines can either be classified as red or white.

The following diagram, "Figure 5-4", represents the query dilemmas for a traditional OLAP Tools.

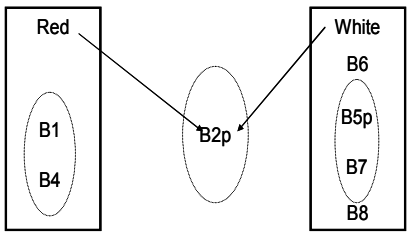


Figure 5-4: Allocation strategies

- Bottles B1 Red Bordeaux, B4 Merlot can be classified as Red Bordeaux or Red wines with absolute certainty.
- Bottles B6 Pinot Gris, B5p Sauvignon and B7 are White Wine Types, for sure. Only bottle B7 is White Bordeaux. Bottle B5p, Sauvignon can be either classified as White Bordeaux or

as Chateau d'Yquem, B8. Chateau d'Yquem is a White wine but not a White Bordeaux

- Bottles B2p, known as Muscat can be either Red or White wines.

The above queries show the importance of H-IFS for two reasons, firstly they allow us to extend the scope of the query and secondly they permit us to consider mixed concepts i.e. Muscat when the queries above are answered.

At this point it is important to estimate the total confidence in B5p being White Bordeaux or as Chateau d'Yquem. Similarly, the total confidence in B2p needs to be estimated, being white or red wine.

The measure can be formulated as following:

$$C_{Ip,\,Region1} = \frac{Sum|Region1|}{Sum|Region1| + Sum|Region2|} \quad Ip \notin Sum|Region|$$

$$C_{B5p,\,Chateau\ d'Yquem} = \frac{|B8|}{|B8| + |B7|} = \frac{|400|}{|400| + |600|} = \frac{4}{10}$$

$$C_{B5p,\,White\ Bordeaux} = \frac{|B7|}{|B7| + |B8|} = \frac{|600|}{|600| + |400|} = \frac{6}{10}$$

$$C_{B2p,\,Red\ Wine} = \frac{|B1|+|B4|}{|B1|+|B4|+|B5|+|B6|+|B7|+|B8|} = \frac{|1160|}{|1160| + |1400|} = \frac{116}{256}$$

$$C_{B2p,\,White\ Wine} = \frac{|B5|+|B6|+|B7|+|B8|}{|B1|+|B4|+|B5|+|B6|+|B7|+|B8|} = \frac{|1400|}{|1160| + |1400|} = \frac{140}{256}$$

As an alternative for the Sum, the Count measure can be utilized. The measure can be formulated by using the following rationale:

Bottle B5p is disputed by two "regions": White Bordeaux and Chateau d'Yquem. So the confidence that B5p is a kind-of Chateau d'Yquem is the sum of sales for Chateau d'Yquem over the sum of all sales for White Bordeaux and Chateau d'Yquem. The item of dispute B5p is excluded from the sum. The same applies for Bottle B2p.

The estimations below are based on the sample data from "Figure 5-3".

When it comes to White Bordeaux and Chateau d'Yquem, the following stands true: it is known for certain that there are 600 bottles of White Bordeaux and possibly more, with a confidence of 6/10, out of 200 Sauvignon. Therefore:

$$\mu_{\text{White Bordeaux, White Wine}} = \frac{600}{1856} = 0.32 \qquad \pi_{\text{White Bordeaux, White Wine}} = \frac{120}{1856} = 0.07$$

As far as Chateau d'Yquem is concerned, there are 400 bottles there for certain and possibly more with a confidence of 4/10 out of 200 Sauvignon.

$$\mu_{\text{Chateau d'Yquem, White Wine}} = \frac{400}{1856} = 0.21 \qquad \pi_{\text{Chateau d'Yquem, White wine}} = \frac{80}{1856} = 0.04$$

Based on the above calculations, a weighted H-IFS sub-domain can be built, "Figure 5-5" suitable for modelling and querying needs of complex/mixed concepts and sample data of the star schema displayed in "Figure 5-3".

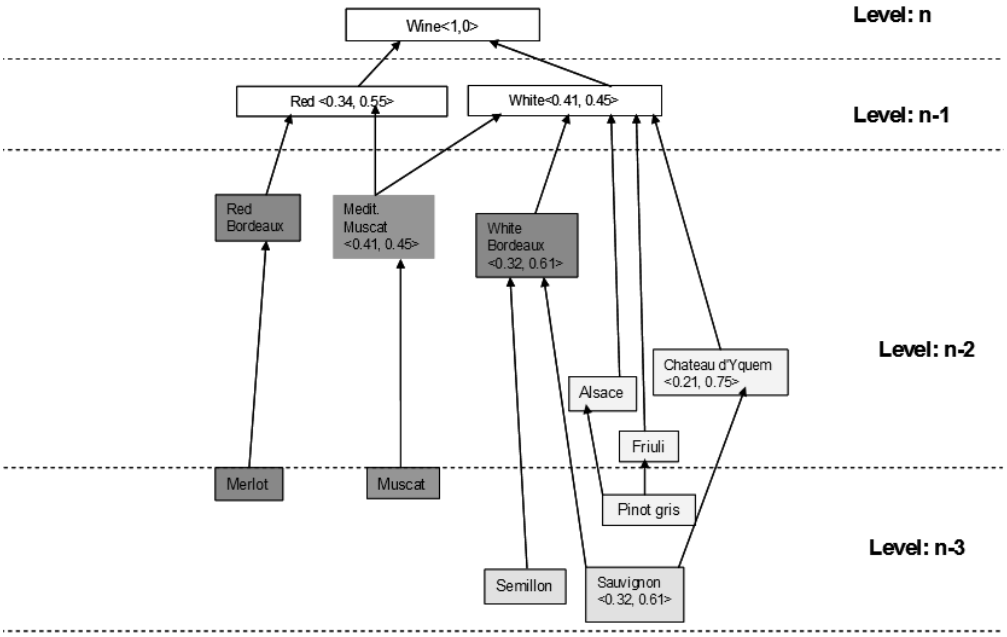


Figure 5-5: The Vitis Vinifera H-IFS

Based on the H-IFS domain presented above and data from the star schema of "Figure 5-3", an OLAP querying mechanism will be presented, capable of dealing with mixed concepts, knowledge and summarising data according to a specified level. The hierarchy used to represent the data, as well as to express queries in the retrieval system, is organized into a hierarchy of terms that corresponds to the taxonomy derived from the Vitis Vinifera Domain.

Further on, the main concepts involved in the designing and implementation the "IF-Oracle" ad-hoc utility will be presented and discussed, so that there can be a demonstration of the potential of

"IF-Oracle" utility when it comes to query answering that requires utilisation of the domain knowledge in order to receive answer close to the user's intent. Finally a conclusion is presented and future research aims and targets are put forward.

'IF-Oracle' has been implemented on top of Oracle10g and allows one to:

- Define an H-IFS hierarchy
- Incorporate hierarchical knowledge in the form of H-IFS as part of the standard SQL queries.
- Enhance the scope of query answers against the Oracle10g standard query answers.

5.4.1 The IF-Oracle, an H-IFS Based Ad-Hoc Utility

IF-Oracle [84] has been developed using Visual Studio.Net as an ad-hoc utility that is attached to and enhances Oracle10g DBMS query capabilities. For demonstrating the functionality of IF-Oracle let's consider a sample multidimensional model, "Figure 5-3", in the form of a star schema that describes sales of Vitis Vinifera type wines.

"Figure 5-6" shows a sub-hierarchy that has been derived from the Vitis Vinifera domain for testing purposes. On the left is shown the tree structure view as displayed in IF-Oracle, while on the right its tree representation.

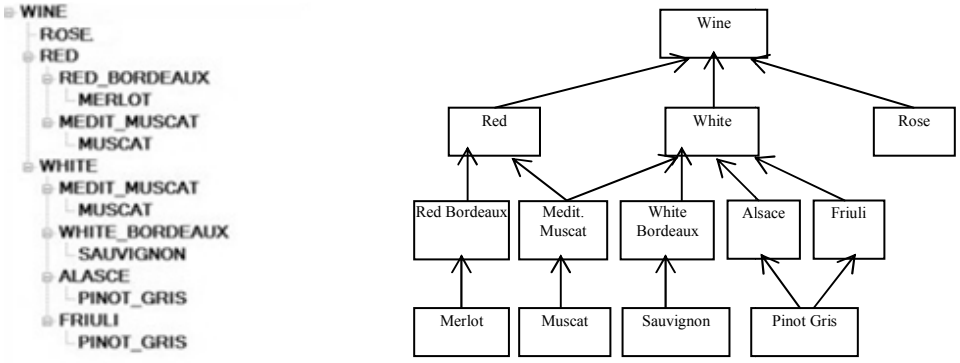


Figure 5-6: ***Vitis Vinifera sub-hierarchy views***

After forming the structure and storing it as a concept relation in Oracle10g, the calculation of the hierarchical closure of the H-IFS and its weights is performed.

The user now has the choice of selecting three different strategies: ***Optimistic, Pessimistic*** or ***Average*** as defined on section 3.3.2.

Let's assume that the user's interest lays on finding information about Red, White and Brown wines.

"Figure 5-7" below, shows the hierarchy after weights have been calculated and assigned reflecting the user's intent.

Figure 5-7: Vitis Vinifera sub-hierarchy view with weights

It can be observed that the principle of the H-IFS closure *(see definition 3)* has been preserved when propagating the degree of validity {μ} and non-validity {ν} from super-elements to sub-elements by using the optimistic strategy.

The degree of validity and non-validity <μ, ν> and π are calculated as follows:

$$\mu = \frac{|c_l|}{|c_{l-1}|} \qquad \nu = \frac{|\neg c_l|}{|c_{l-1}|} \qquad \pi = 1 - (\mu + \nu)$$

Where c_l corresponds to those elements from the fact table that absolutely satisfy the selection criteria with reference to a node in the hierarchy. C_{l-1} represents the children elements of that selection on a lower level that satisfy the selection condition to some extent. It is obvious that:

$$\pi = 1 - (\mu + \nu)$$

After adding the hierarchy into the repository and automatically calculating the weights for the requested nodes, the user can utilize the ad-hoc interface for execution of queries either in standard SQL or make use of the enhanced Select clause and features that IF-Oracle provides.

"Figure 5-8" shows the results of a user request for "Red" wine executed in standard SQL provided by Oracle10g.

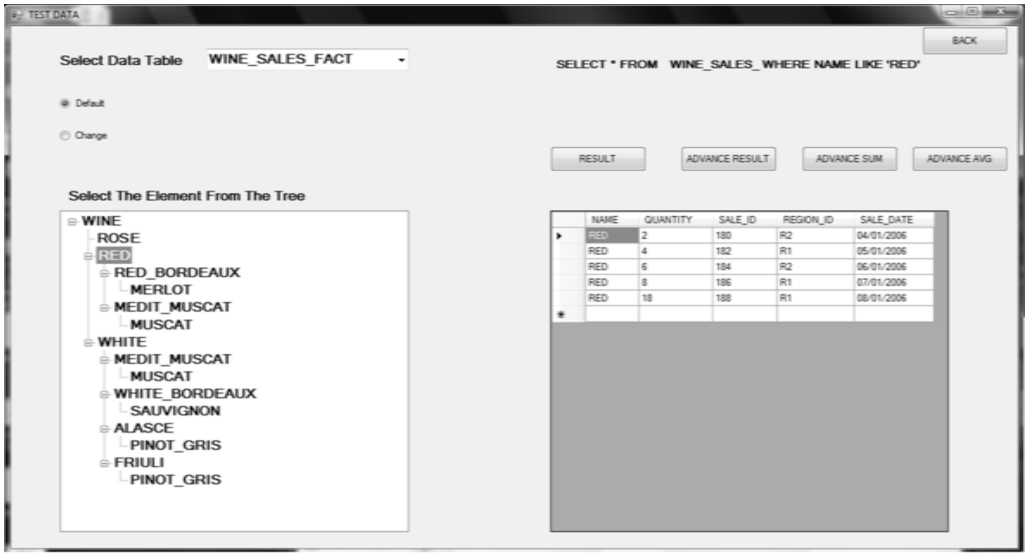


Figure 5-8: Standard SQL output for "Red" wine

In contrast, "Figure 5-9" shows the output after executing the same query, but this time using the IF-Oracle utility.

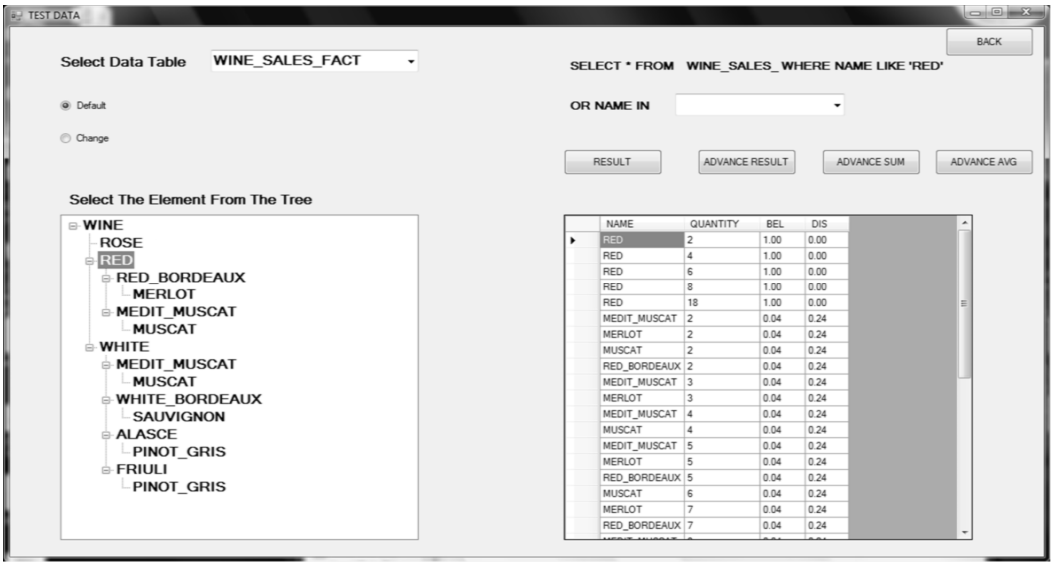


Figure 5-9: Enhanced SQL output for "Red" wine

By comparing the two figures, one can observe that IF-Oracle produces a knowledge-based answer instead of mindlessly matching the records against the word "Red".

The results show that IF-Oracle not only retrieves sales of "Red" bottles, but also sales of bottles that are classified as red wines by the knowledge represented in the H-IFS hierarchy as "Merlot", "Red Bordeaux", "Muscat", etc. with indicative degrees of $<\mu, \nu>$ relevant to the user's preference.

5.5 Conclusions

A context for capturing and representing the semantics of value imprecision under three levels of abstraction, the meta-model level, the specification level and the instance level has been delivered. In this way the properties of value imprecision can be captured and formally defined, verified at the meta-model level. A specification level has been delivered, enabling users or application designers to express the imprecision semantics in a simple and formal way. A post-relational environment for handling uncertainty has been defined.

The focus is on delivering an OLAP architecture that allows the integration of hierarchical preferences expressed in the form of background/domain-knowledge, with the aim on enhancing the query scope and in return receiving a richer answer, closer to user requests. This thesis provided means of using background knowledge to re-engineer query processing and answering with the aid of H-IFS and Intuitionistic Fuzzy relational representation. The hierarchical links defined on the basis of the H-IFS closure represent knowledge in the form of enhanced "kind-of, ≤" relation. The membership of an element in an H-IFS has consequences on the membership and non-membership of its sub-elements in this set.

This chapter has demonstrated the simplicity and implementation-ability of the H-IFS notion by adding an ad-hoc utility 'IF-Oracle' in Oracle10g that allowed the enrichment of the scope of query and the return of answers closer to user's intent and preferences, even when answers are not obvious when using the standard SQL provided by Oracle10g.

"I think and think for months and years. Ninety-nine times, the conclusion is false. The hundredth time I am right."

Albert Einstein

Chapter Six

6. Conclusions and further research

6.1 Short Summary

In Chapter 1 the meaning of imprecise/ignorant information in data repositories was defined. Proposed extensions to the relational data model so that it can represent and retrieve such information were examined. Various kinds of imprecise information were explored, such as: fuzzy, missing, partial, possible, probabilistic, unknown.

In Chapter 2 the various kinds of *value imprecision* were described along with their meaning in the context of a formal model. *Value imprecision* is positioned in the context of data repositories as part of an attribute-value and On-Line Analytical Processing (OLAP), as part of the axis of analysis/dimensions.

In Chapter 3 a unique solution was brought forward that allowed the representation of flexible user-defined hierarchies as part of the dimension structures. This solution was defined as "Hierarchical-Intuitionistic-Fuzzy-Sets" or "H-IFS", an extension of the Intuitionistic Fuzzy Sets.

In Chapter 4 the issue of representing imprecise facts and the answering of queries based on user-defined hierarchical preferences was dealt by delivering an extended multidimensional model and query operators named the "IF-Cube".

In Chapter 5 the KNOLAP conceptual formalism was revealed, which put forward a common framework for defining value imprecision with respect to multidimensional modelling properties. Based on the KNOLAP conceptual formalism, the "IF-Oracle" ad-hoc utility was delivered. IF-Oracle utilized the concepts of H-IFS

and IF-Cube as part of the data definition and manipulation language, allowing thus the encoding of imprecision either as part of the attribute values or the axis of analysis/dimensions.

6.2 Contributions

This thesis provided a theoretical framework of using background knowledge to re-engineer query processing and answering with the aid of H-IFS and Intuitionistic Fuzzy relational representation. The hierarchical links defined on the basis of the H-IFS closure represent knowledge in the form of enhanced "kind-of, ≤" relation.

The automatic recommendation of analysis was introduced according to the concepts defined as part of domain description in order to guide query answering with the aid of Hierarchical Intuitionistic Fuzzy Sets (H-IFS), an extension of the Intuitionistic Fuzzy Sets. Accommodating imprecision or user preferences at the level of the dimensions or axis of analysis as part of a multidimensional model can be achieved with the aid of H-IFS. Furthermore, based on the concept of minimal H-IFS it is ensured that queries utilize the minimum number of predicates when it comes to query formulation prior to execution. In terms of data representation when it comes to OLAP analysis, the following cases can be accommodated:

- Crisp dimensions and precise data
- H-IFS based hierarchies/dimensions and precise data
- Crisp hierarchies/dimensions and imprecise data
- H-IFS based hierarchies/dimensions and imprecise data

In order to achieve the accommodation of imprecision at the level of facts/data, it became clear that there is a need to extend the

standard cubic model and the related OLAP-query operators. A new approach for extending the MOLAP model was presented, so that it can include treatment of value imprecision as part of a multidimensional model inhabited by concepts and non-rigid hierarchical structures of organisation. A new multidimensional-cubic model named the IF-Cube was introduced, which is able to operate over data with imprecision either in the facts or in the dimensional hierarchies.

A conceptual formalism known as the KNOLAP meta-model for capturing and representing the semantics of value imprecision with the aid of three levels of abstraction, the meta-model level, the specification level and the instance level has been delivered. In this way the properties of value imprecision can be captured and formally defined, verified at the meta-model level. A specification level has been delivered, enabling users or application designers to express the imprecision semantics, in a simple and formal way. The delivered IF-Oracle ad-hoc utility is a realisation of the KNOLAP meta-model. The focus is on delivering an OLAP architecture that allows the integration of hierarchical preferences expressed in the form of background domain-knowledge with the aim on enhancing the query scope and in return receiving a richer answer, closer to user intents and preferences.

6.3 Limitations and Future Work

In order to implement the concept of H-IFS and the IF-Cube model, a platform was required which would provide a DW, OLAP browsing, based on the fundamental concept of possible worlds. Commercial tools did not fit the need for customised software that supports OLAP processing based on the concept of possible worlds.

The IF-Oracle ad-hoc utility allows OLAP processing with the retrieval of not only definite answers but also possible answers, so it complies with the fundamental concept of possible worlds when it comes to query answering. The IF-Oracle ad-hoc utility is currently implemented on top of Oracle-10g and in general can be used as an ad-hoc utility to any of the existing OLAP servers. However as an ad-hoc utility, IF-Oracle's modelling and query processing power is dependent on the underlying OLAP-database engine. Some MOLAP products have difficulty updating and querying models with more than ten dimensions. The possibility of building a dedicated OLAP engine customised to the semantics of possible worlds, although is semantically desirable, in practice will be proven to be a very costly effort in terms of data storage and indexing structures with little chances to compete successfully against existing commercial OLAP engines and Database vendors. The IF-Oracle utility currently ensures that queries utilize the minimum number of predicates, with the aid of minimal H-IFS, when it comes to query formulation prior to execution.

In terms of future work, one could further equip the IF-Oracle utility with a collection of global predicates to describe the overall Data-warehouse contents and validate user queries before formulation and execution. This is quite important for achieving better query performance since it will stop the formulation of meaningless queries. For this reason, the IF-Oracle architecture can be equipped further with a repository that contains various constraints (i.e. Intuitionistic Fuzzy Range Constraints, Intuitionistic Fuzzy Functional Dependencies, etc) that are related to the information sources that participate on the Data-warehouse environment. Furthermore, one could envisage the incorporation of online data sources to further enrich the quality of an H-IFS based environment.

Considering the problem of evolution in data-warehouses, most of the current OLAP systems report data in the most recent analysis structure. However, working only with the latest version hides the existence of evolution and information that may be critical for data analysis. H-IFS can potentially act as a suitable medium for building a structured version of dimensions, allowing thus users to have a view of the evolution of the enterprise performance or achievements over the time.

Overall it can be said that contributions of this thesis can be utilized in querying answering systems for encoding domain knowledge and then utilising it for further enhancing the query formulation and answers obtained from current querying tools.

References

1. A. Silberschatz, M. Stonebraker, J. D. Ullman. Database systems: Achievements and opportunities. – Commun. ACM, Vol. 34, 1991, No 10, pp.110-119.

2. V.S. Lakshmanan, F. Sadri. Modelling uncertainty in deductive databases. – In Proc. of the Conference on Database Expert Systems and Applications, Springer-LNCS, 1994, pp.724-733.

3. P. Haddawy, Believing Change and Changing Belief. – IEEE Transactions on Systems, Man and Cybernetics, - Vol.26, 1996, No.3, pp.385-409.

4. S. Iyengar, S. Prasad, L. H. Min. Advances in Distributed Sensor Technology, Prentice-Hall, 1995.

5. B. Buckles, F. Petry. Fuzzy databases in the new era. – In: Proc. of the 1995 ACM symposium on Applied computing, 1995, pp.497- 502.

6. D. Dubois, H. Prade. Certainty and uncertainty of vague knowledge and generalized dependencies in fuzzy databases. – In: Proc. of the International Fuzzy Engineering Symposium, 1988, pp.239-249.

7. P. Smets, Belief functions in Non standard logics for automated reasoning, P. Smets, A. Mamdani, D. Dubois, H. Prade, Eds., London, Academic Press, 1988, pp.253-286.

8. R. George, B. Buckles, F. Petry. Modeling class hierarchies in the fuzzy object-oriented model. – Fuzzy Sets and Systems, Vol. 60, 1993, No 3, pp.253-272.

9. J. Ullman, D. Principles of Database and Knowledge Base Systems, Computer Science Press, 1989.

10. L. Zadeh, Fuzzy sets as a basis for a theory of possibility. – Fuzzy Sets and Systems, Vol. 1, 1978, pp.3-28.

11. B. Buckles, F. Petry. Extension of the fuzzy database with fuzzy numbers. – Information Sciences, Vol. 34, 1984, No 4, pp.121-132.

12. P. Bosc, O. Pivert. Fuzzy querying in conventional databases, In: L. Zadeh and J. Kacprzyk, Eds. – In: Fuzzy Logic for the Management of Uncertainty. John Wiley and Sons, 1992, pp.645-671.

13. P. Chountas, I. Petrounias. Precise Enterprises and Imprecise Data. J., Barzdins, A. Caplinskas Eds. – In: Databases and Information Systems. Kluwer Academic Publishers, ACM- Portal, 2001, pp.57-68.

14. J. Grant, J. Minker. Answering Queries in Indefinite Databases and the Null Value Problem. – In: Advances in Computing Research, Vol. 3, 1986, pp.247-267.

15. P. Chountas, Review of uncertainty schools of thoughts & data representation. – Issues in Intuitionistic Fuzzy Sets and Generalized nets, 2007, Vol. 6, Wydawnictwo, WSISIZ

16. D. Barbará, H. García-Molina, D. Porter. A probabilistic relational data model. – In: Proceedings of the International Conference on Extending Database Technology, Advances in Database Technology, 1990, pp.60-74.

17. D. Barbará, H. García-Molina, D. Porter. The management of probabilistic data. – IEEE Transactions on Knowledge Data Engineering, Vol. 4, 1992, No 5, pp.487-502.

18. R. Cavallo, M. Pittarelli. The theory of probabilistic databases. – In: Proceedings of the International Conference on Very Large Data Bases, IEEE Computer Society Press, 1987, pp.71-81.

19. D. Dey, S. Sarkar. A probabilistic relational model and algebra. – ACM Trans. Database Systems, Vol. 21, 1996, No 3, pp.339-369.

20. V.S. Lakshmanan, N. Leone, R. Ross, V. S. Subrahmanian. ProbView: A flexible probabilistic database system. – ACM Trans. Database System, Vol. 22, 1997, No 3, pp.419-469.

21. M. Zemankova, A. Kandel. Implementing imprecision in Information systems. – Information Science, Vol. 37, 1985, pp.7-141.

22. D. Dubois, H. Prade, C. Testamale. Handling Incomplete or Uncertain Data and Vague Queries in Database Applications. Plenum Press, 1988.

23. H. Prade, Annotated bibliography on fuzzy information processing. – In: Readings on Fuzzy Sets in Intelligent Systems, H. Prade, D. Dubois, and R. Yager, Eds. Morgan Kaufmann Publishers Inc., 1993.

24. E. Codd, Extending the data base relational model to capture more meaning. – ACM Trans. Database Systems, Vol. 4, 1979, No 4, pp.397-434.

25. B.S. Goldstein, Constraints on null values in relational databases. – In: Proc. of the7-th Int. Conf. on VLDB, IEEE Press, 1981, pp.101-110.

26. J. Biskup, A Foundation of Codd's Relational Maybe-Operations, – In: XP2 Workshop on Relational Database Theory, 1981.

27. C. Liu, R. Sunderraman, "Indefinite and maybe information in relational databases" *ACM Trans. Database Syst.* Vol.15, No.1, 1990, pp.1–39.

28. C. Liu, R. Sunderraman On representing indefinite and maybe information in relational databases: A generalization. – In: International Conference on Data Engineering, IEEE Computer Society, 1990, pp.495-502.

29. A. Ola, Relational databases with exclusive disjunctions. – Data Engineering, 1992, pp.328-336.

30. W. Homenda, Databases with alternative information. – IEEE Trans. on Knowledge and Data Engineering, Vol. 3, 1991, No 3, pp.384-386.

31. G.H. Gessert, Handling missing data by using stored truth values. – SIGMOD Record, Vol. 20, 1991, No 1, pp.30-42.

32. R. Zicari, Closed world databases opened through null values. – In: Proc. of the 14-th Int. Conf. on VLDB, Morgan Kaufmann, 1988, pp.50-61.

33. J. Lipski, On semantic issues connected with incomplete information databases. – ACM Trans. Database System, Vol. 4, 1979, No 3, pp.262-296.

34. C. Zaniolo, Database relations with null values. – Journal of Comput. Syst. Sci., Vol. 28, 1984, pp.142-166.

35. J. Grant, Partial values in a tabular database model. – Information Processing Letters, Vol. 9, 1979, No 2, pp.97-99.

36. C. Zaniolo, S. Ceri, C. Faloutsos, R. Snodgrass, V. S. Subrahmanian, C. Zicari. Advanced Database Systems. Morgan Kaufman, 1997.

37. G. Gottlob, R. Zicari. Closed world assumption opened through null values. – In: Proc.of the Fourteenth International Conference on VLDB, Morgan Kaufman, 1988, pp.50-61.

38. J. Biskup, A foundation of Codd's relational maybe-opera-tions. – ACM Trans. on Database Systems, Vol. 8, 1983, No 4, pp.608-636.

39. A. Motro, Imprecision and incompleteness in relational databases. – Survey. Inf. Softw. Technol., Vol. 32, 1990, No 9, pp.579-588.

40. S. Abitebul, G. Grahne. Upadate semantics for incomplete databases. – In: Proc. of the 11-th International Conference on Very Large Databases, Morgan Kaufman, 1985, pp.1-12.

41. Y. Vassiliou, Null values in database management - a denotational semantics approach. – In Proc. of the ACM SIGMOD International Conference on the Management of Data, 1979, ACM Press, pp.162-169.

42. J. Melton, A. R. Simon. Understanding the New SQL: A Complete Guide. Morgan Kaufmann Publishers Inc., 1992.

43. R. Kimball, The Data Warehouse Toolkit. New York, John Wiley & Sons, 1996.

44. W. H. Inmon, Building the Data Warehouse. Second Edition, New York, John Wiley & Sons, 1996.

45. T. B. Pedersen, C. S. Jensen. Multidimensional data modelling for complex data. – In: Proc. of 15th ICDE, IEEE Computer Society, 1999, pp.336-345.

46. J. Gray, A. Bosworth, A. Layman, H. Pirahesh. Data cube: a relational aggregation operator generalizing group-by, cross-tabs and subtotals. – Journal of Data Mining and Knowledge Discovery, Springer Verlag, Vol. 1, 1997, No 1, pp.29-53.

47. L. Lakshmanan, J. Pei, Y. Zhao QC-Trees: An Efficient Summary Structure for Semantic OLAP, SIGMOD 2003

48. D. Burdick, A. Doan, R. Ramakrishnan, S. Vaithyanathan OLAP over imprecise data with domain constraints, Proceedings of the 33rd international conference on Very large data bases, 2007.

49. M. Delgado, C. Molina, D. Sanchez, A. Vila, L. Rodriguez-Ariza, A fuzzy multidimensional model for supporting imprecision in OLAP, 2004 IEEE International Conference on Fuzzy Systems, pp. 1331-1336

50. P. Bosc, O. Pivert An Approach for a Hierarchical Aggregation of Fuzzy Predicates, Second IEEE International Conference on Fuzzy Systems 1993, pp. 1231-1236

51. A. Laurent , B. Bouchon-meunier , A. Doucet , Towards Fuzzy-OLAP Mining, 2001

52. P. Kumar, P. Radha Krishna and S. K. De Fuzzy OLAP cube for Qualitative Analysis International Conference on Intelligent Sensing and Information Processing, ICISIP-2005, Chennai, India, 2005, pp. 290-295.

53. T. Pedersen, C. Jensen, Multidimensional Database Technology. Distributed Systems Online (IEEE): 2001, pp.40–46. ISSN 0018-9162.

54. X. Dong, H. Huang, H. Li HQC: An Efficient Method for ROLAP with Hierarchical Dimensions, Lecture Notes in Computer Science, Volume 3642/2005, pp. 211-220

55. K. Hu , C. Ling, S. Jie, G. Qi, X. Tang High Dimensional MOLAP with Parallel Shell Mini-cubes, Lecture Notes in Computer Science, Volume 3613/2005, pp. 1192-1196

56. M. Kaufmann Application of Fuzzy Classification to a Data Warehouse in E-Health, Master's Thesis, University of Fribourg, Switzerland, 2006

57. J. Liu, M. Luo, X. Yang, Deputy Mechanism for OLAP over Imprecise Data and Composite Measure, IEEE 7th International conference on Computer and Information Technology 2007, pp. 65-70.

58. Z. Peng, Y. Kambayashi, "Deputy Mechanism for Object-Oriented Database", In Proccessings of 11th ICDE Conference, 1995, pp. 476-493.

59. D. Burdick, P.M. Deshpande, T.S. Jayram, R. Ramakrishnan, S. Vaithyanathan, OLAP over Uncertain and Imprecise Data. *Proceedings of the 31st VLDB conference, 2005.*

60. T. B. Pedersen, C. S. Jensen, C. E. Dyreson, Supporting Imprecision in Multidimensional Databases Using Granularities, SSDBM 1999, pp. 90-101

61. B. R Moole, A Probabilistic Multidimensional Data Model and Algebra for OLAP in Decision Support Systems, Proceedings IEEE Southeastcon 2003, pp18-30

62. H. Thomas & A. Datta, A Conceptual Model and Algebra for On-Line Analytical Processing in Decision Support Databases. Information Systems Research 12: pp.83-102, 2001

63. D. Dey, S. Sarkar, "A probabilistic relational model and algebra" *ACM Trans. Database Syst.* Vol.*21*, No.3, 1996, pp 339–369

64. D. Dey, S. Sarkar, "Modifications of Uncertain Data: A Bayesian Framework for Belief Revision", Information Systems Research, Vol. 11, No. 1, March 2000, pp. 1-16

65. "*Evaluation of Term-Based Queries Using Possibilistic Ontologies*," Soft Computing for Information Retrieval on the Web, E. Herrera-Viedma, G. Pasi, F. Crestani, eds. Springer-Verlag,2005.

66. M. Koyuncu, A. Yazici, "IFOOD: *An Intelligent Fuzzy Object-Oriented Database Architecture"*. IEEE Trans. Knowl. Data Eng. 15(5): pp. 1137-1154, 2003

67. G. Bordogna, G. Pasi, "*Modeling Vagueness in Information Retrieval"*. ESSIR 2000, pp.207-241

68. R. Thomopoulos, P.Buche, O. Haemmerlé: *"Fuzzy Sets Defined on a Hierarchical Domain"*. IEEE Trans. Knowl. Data Eng. 18(10): pp.1397-1410 (2006)

69. A. Amo, J. Montero, G. Biging, V. Cutello: "*Fuzzy classification systems"* European Journal of Operational Research (2004) Volume 156, Issue 2, pp.495-507

70. E. Szmidt, J. Baldwin "*Intuitionistic Fuzzy Set Functions, Mass Assignment Theory, Possibility Theory and Histograms"*. 2006 IEEE World Congress on Computational Intelligence pp.237—243.

71. E. Szmidt, J. Kacprzyk "*Intuitionistic fuzzy sets in group decision making",* Notes on IFS, 2, pp.15—32. (1996)

72. E. Szmidt, J. Kacprzyk "*Remarks on some applications of intuitionistic fuzzy sets in decision making"*, Notes on IFS, 2(3), (1996) pp.22—31.

73. K. Attanasov, "Intuitionistic Fuzzy Sets: Theory and Applications", Springer-Verlag, 1999

74. S. Zadrozny, J. Kacprzyk: Bipolar Queries and Queries with Preferences. DEXA Workshops 2006: pp.415-419

75. S. Rice, J. F. Roddick, "Lattice-Structured Domains, Imperfect Data and Inductive Queries", LNCS, DEXA, 2000, pp. 664-674

76. D. Bell, J. Guan, S. Lee "Generalized union and project operations for pooling uncertain and imprecise information". DKE 18 (1996), pp.89-117

77. G. Pasi, F. Crestani, "Evaluation of Term-Based Queries Using Possibilistic Ontologies," Soft Computing for Information Retrieval on the Web, Springer-Verlag, 2005.

78. S. Miyamoto, K. Nakayama, "Fuzzy Information Retrieval Based on a Fuzzy Pseudothesaurus," IEEE Trans. Systems, Man and Cybernetics, 1986, Vol. 16, no. 2, pp. 278-282

79. E. Rogova, P. Chountas, K. Atanassov "Flexible hierarchies and fuzzy knowledge-based OLAP". – FSKD 2007, IEEE Computer Society Press Vol.2, pp. 7-11

80. E. Rogova, P. Chountas, K. Atanassov, The Notion of H-IFS in Data Modelling, Fuzz-IEEE WCCI'08, *International Conference on Fuzzy Systems,* IEEE Computational Intelligence, pp.1397-1403

81. B. Kolev, P. Chountas, E. Rogova, K. Atanassov, Representation of Value imperfection with the Aid of Background Knowledge – H-IFS, Intelligent Techniques and Tools for Novel System Architectures, 2008 Spriger-Verlag Series in Computational Intelligence pp. 473-492

82. E. Rogova, P. Chountas, On Imprecision Intuitionistic Fuzzy Sets & OLAP – The Case for KNOLAP, IFSA'07, Springer-Verlag GmBh , Theoretical advances and application of fuzzy logic and soft computing, pp. 11-20

83. E. Rogova, P. Chountas, B. Kolev, Intuitionistic Fyzzy Knowledge-based OLAP, Notes on Intuitionistic Fuzzy sets, Vol. 13, No.2, ISSN 1310-4926, 2007, pp.88-100

84. P. Chountas, E. Rogova, K. Atanassov, S. Mohammed, The Notion of H-IFS -An Approach for Enhancing Query Capabilities in Oracle10g, IEEE IS'08, IEEE Computer Society Press ISBN: 978-1-4244-1739-1, 2008, pp.13-8 to13-13

85. K. Atanassov (1999). Intuitionistic Fuzzy Sets, Springer-Verlag, Heidelberg

86. K. Atanassov Remarks on the Intuitionistic fuzzy sets. – Fuzzy Sets and Systems, Vol. 51, 1992, No 1, pp.117-118.

87. P. Chountas, K. Atanassov "On Intuitionistic Fuzzy Sets over Universes Hierarchical Structures", Notes on Intuitionistic Fuzzy Sets, 2007, Vol13, No1, pp. 52-56.

88. P. Chountas, E. Rogova, B. Kolev, V. Tasseva, K. Atanassov, "Generalized Nets in Artificial Intelligence, Vol. Generalised Nets, Uncertain Data, and Knowledge Engineering", ISBN 978-954-322-255-1, pp. 1-149, Publishing House of Bulgarian Academy of Sciences.

89. S. Chaudhuri, U. Dayal, V. Ganti Database Technology for Decision Support Systems. In: Computer, Vol. 34, p. 48-55, 2001

90. M. Jarke, Fundamentals of data warehouses. Springer, London, 2002

91. C. Dyreson, Information retrieval from an incomplete data cube, VLDB, Morgan Kaufman Publishers, pp. 532-543, 1996.

92. T. Pedersen, C. Jensen, and C. Dyreson, A foundation for capturing and querying complex multidimensional data, Information Systems, vol. 26, pp. 383-483, 2001.

93. A. Abelló, J. Samos, F. Saltor, YAM2 a multidimensional conceptual model extending UML Journal Inf. Systems, 2006, Vol.31(6) pp. 541-567 .

94. P. Chountas, K. Atanassov, E. Rogova H-IFS: Modelling & Querying over Hierarchical Universes, International Conference On Information Processing And Management of Uncertainty-IPMU'2008, ISBN 978-84-612-3061-7,pp 1628-1634