Probabilistic Data Modeling and Querying for Location-Based Data Warehouses

Igor Timko, Curtis E. Dyreson, and Torben Bach Pedersen

October 7, 2005

TR-13

A DB Technical Report

Title	Probabilistic Data Modeling and Querying for Location-Based Data Warehouses
	Copyright © 2005 Igor Timko, Curtis E. Dyreson, and Torben Bach Pedersen. All rights reserved.
Author(s)	Igor Timko, Curtis E. Dyreson, and Torben Bach Pedersen
Publication History	March 2005. A DB Technical Report

Abstract

Motivated by the increasing need to handle complex, dynamic, uncertain multidimensional data in location-based warehouses, this paper proposes a novel probabilistic data model that can address the complexities of such data. The model provides a foundation for handling complex hierarchical and uncertain data, e.g., data from the location-based services domain such as transportation infrastructures and the attached static and dynamic content such as speed limits and vehicle positions. The paper also presents algebraic operators that support querying of such data. Use of pre-aggregation for implementation of the operators is also discussed. The work is motivated with a real-world case study, based on our collaboration with a leading Danish vendor of location-based services.

1 Introduction

Corporate and personal use of location-based services (LBSs), e.g., traffic or tourist related services, is increasing. LBSs generate massive amounts of location-based data that must be analyzed in order to optimize and personalize the services. Of particular interest are aggregation queries that involve the transportation infrastructure and attached content, e.g., "How many users (in their cars) of age less that 21 will be in the eastbound lane of Main Street five minutes from now?". Current OLAP and data warehouse (DW) technology [12, 14, 19] supports aggregation queries based on a *multidimensional* data model capturing hierarchies of dimensional data. Unlike other types of data models, multidimensional models provide first-class support for interactive, investigative aggregate queries on complex data, e.g., roll-up and drill-down queries [20]. This creates a need for location-based data warehouses (LBDWs) that can offer the benefits of traditional DW technology for LBS data. Current DW technology, both in research and industry, can provide support for many kinds of LBDW queries, but LBDWs have additional complexities that are not well-supported by traditional DWs such as uncertain data. For example, in a transportation infrastructure, cars are moving dynamically, so the future location of a car is uncertain. Moreover, the current location is sometimes also uncertain (e.g., known only to a wireless phone grid). Since the problem domain is very complex, a formal foundation for LBDWs is needed.

The contributions of this paper are as follows. First, the paper presents a probabilistic multidimensional data model (extension of the deterministic model from [27]) that can manage uncertain LBS data. The probabilities appear both in dimension hierarchies (a dimension value may *partially* contain another value) and fact characterizations (facts are characterized by dimension values with certain probabilities). Second, the paper presents a set of algebraic operators for querying the modeled uncertain data (extension of the operators from [20]). Third, the paper defines different types of probabilistic fact groupings for aggregation. Fourth, the paper defines different types of probabilistic aggregation functions applied to the groups. Finally, implementation of the data model and the algebraic operators, including the use of pre-aggregation for query processing, is also discussed. The paper thus extends current OLAP/DW technology with means for supporting LBDWs. The concepts presented in the paper are illustrated using a real-world case study from the LBS domain. The work is based on an on-going collaboration with a leading Danish LBS vendor, Euman A/S [8].

Previous related work has generally fallen into three categories: probabilistic databases, spatio-temporal databases, and multidimensional OLAP databases. The work on probabilistic data management in general [1, 2, 6, 9] handles basic uncertainty in the data, but does not support dimensional data with hierarchies and LBS specifics such as transportation infrastructures and attached content. Research in general-purpose spatio-temporal data management considers "operational" queries on certain [22, 23, 25] or uncertain [4, 5, 28, 29] spatio-temporal data in 2D spaces or transportation infrastructures, but do not consider aggregation/analysis queries. Some papers [21, 26, 30, 31] have considered aggregation of spatial or spatio-temporal data, but have not considered transportation infrastructures.

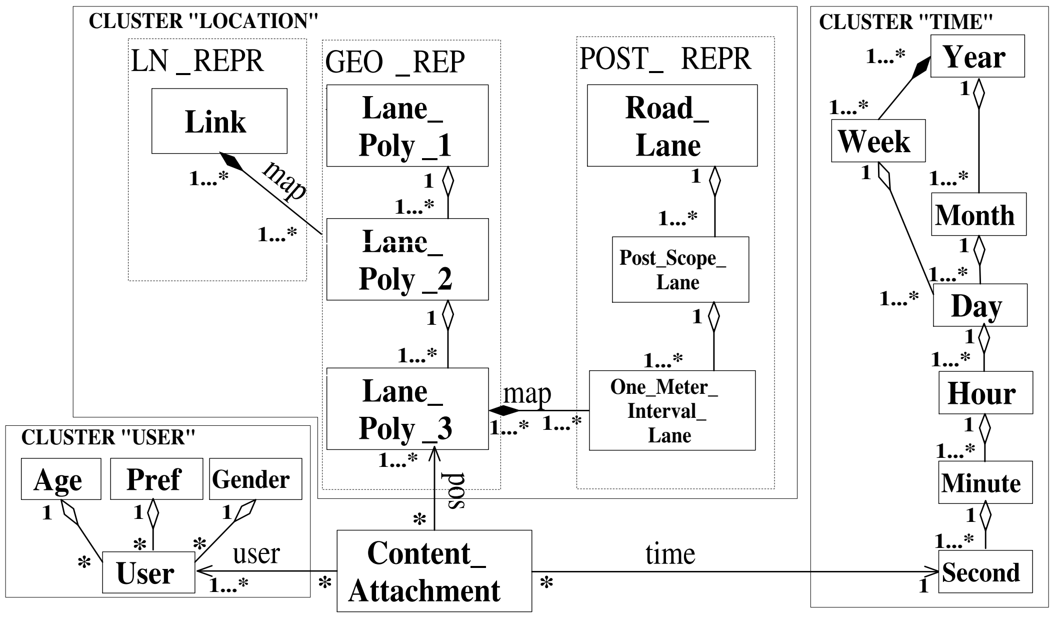


Figure 1: Case Study

Previous research has also covered the modeling of moving objects [10], transportation infrastructures [17, 18], or both [24], for "operational", i.e., non-analytical, purposes. The Euman data model [11] handles multiple infrastructure representation based on a *segment-based* model that is a generalization of the popular *linear referencing* technique [18]. However, none of this work captures data in a multidimensional framework, and thus does not provide optimal support for DW-like analytical querying, nor does it address the inherent uncertainties in LBS data.

Previous work on modeling multidimensional data, e.g., [20], does not handle the complexities of LBDWs. On the one hand, the data model and algebra presented in [13] support LBDW to a certain extent by allowing partial containment dimension hierarchies, while [27] improves on [13] by additionally handling transportation infrastructures and complex content. However, neither [13] nor [27] handles uncertainty in the data. On the other hand, a probabilistic multidimensional data model [16] does handle uncertain data (by assigning a *single* probability to a *whole* row of a fact table) but does not support the other mentioned features of LBDWs. Moreover, our data model takes a more general and more flexible approach to handling uncertainty (a probability is assigned to *each* attribute of a fact table row).

The remainder of the paper is structured as follows. Section 2 presents the case study, and describes content and queries. Section 3 briefly introduces the model we use as the foundation, namely the [OLAP$_{LBS}$] model from [27]. Section 4 describes our approach to handling spatial hierarchies using expected degrees of containment, while Section 5 deals with probabilistic fact characterizations. Section 6 describes the formal query algebra. Pre-aggregation is discussed in Section 7. Section 8 concludes the paper and points to future work.

2 Case Study

We now discuss the requirements for an LBDW by presenting a real-world case study for which a UML diagram can be seen in Figure 1.

2.1 Content

We start with discussing LBS content. LBDW have both *point* and *interval* content [11]. *Point content* concerns entities that are located at a specific geographic location, have no relevant spatial extent, and are attached to specific points in the transportation infrastructure, e.g., traffic accidents, museums, gas stations, and (users' and other's) vehicle positions. *Interval content* concerns data that is considered to relate to a *road section* and is thus attached to intervals of given roads. Examples include speed limits and road surfaces. Our model must capture both point and interval content.

Content can be further classified as *dynamic* (frequently evolving) or *static* (rarely evolving). *Static* content, e.g., gas stations or speed limits, remains attached to a point or an interval of a road for a relatively long period of time. In this paper, we focus on very dynamic (*hyper-dynamic*) content, e.g., vehicle positions and their predicted trajectories (which evolve continuously). Positions of static content are usually certain, while positions of dynamic content are usually uncertain, e.g., a vehicle position is approximated by a wireless phone cell. Furthermore, any position prediction algorithm will have some degree of uncertainty. Thus, we must capture content of any degree of dynamism, as well as uncertainty, in our model,

In Figure 1, hyper-dynamic content is modeled by the "USER" cluster, where the "User" class represents users and (implicitly) their vehicles, The "User" class participates in three *full containment relationships* capturing user age, preference, and gender. The users' (vehicle) positions in the infrastructure is modeled by the "LOCATION" cluster. The positions are captured at certain times, represented by the "TIME" cluster. This content positioning/attachment, is modeled as a "Content_Attachment" class which is linked to users, positions, and times. In an OLAP multidimensional model, the "Content_Attachment" class would be a *fact* characterized by "USER", "LOCATION", and "TIME" *dimensions*.

2.2 Transportation Infrastructure

We now discuss the aspects of the transportation infrastructure relevant to data aggregation. Different, purpose-dedicated infrastructure representations, may be used, but most modern types of infrastructure representations, e.g., kilometer-post and geographic, are (1) *segment-based* and (2) *hierarchical* [11]. Thus, our data model must capture different types of segment-based and, possibly, hierarchical representations.

The "LOCATION" cluster from the UML diagram in Figure 1 contains three segment-based representations, "LN_REPR", "GEO_REPR", and "POST_REPR", which are link-node, geographic, and kilometer post representations, respectively. All three are a refinement of real-world representations used by the LBS company Euman A/S [8], obtained by representing *lanes* instead of *roads*. Often, lanes of the same road have different characteristics, e.g., different traffic density, so lanes must be captured separately [24]. We refer to segments that capture individual lanes, as *lane segments*. Lane segments may be further subdivided into subsegments to obtain more precise positioning (see "Content"). In the "LOCATION" cluster, each such lane segment level is a separate class. "LN_REPR" has only one level, "Link", that contain segments where the characteristics such as the speed limit remain constant. "POST_REPR" has three levels: 1) the "Road_Lane" class which captures particular lanes, e.g., a lane on an exit from a highway; 2) the "Post_Scope_Lane" class which captures segments between two kilometer posts, i.e, subdivisions of the road lanes above; 3) the "One_Meter_Interval_Lane" class which captures one-meter intervals (of the post scope segments above). The "GEO_REPR" also has three levels. Here, a segment is a two-dimensional polyline representing (part of) a lane. Thus, a segment level is a geographical map. A sequence of segments from the "Lane_Poly_3" class (finest scale map), is approximated by (contained in) a segment from the "Lane_Poly_2" class (medium scale map), and similarly for "Lane_Poly_2" and "Lane_Poly_1" (coarsest scale map), see [11] for details. The levels define a hierarchy of *full containment (aggregation) relationships* between segments, which is denoted by empty rhombus-headed arrows in our model.

Finally, relationships between the representations must be captured, to allow content attached to one representation to be accessible from another. In the diagram, the relationships between the representations are modeled as aggregation relationships labeled "map". Due to differences in how and from what data the representations are built, these mappings are *partial containment relationships*, i.e., segments from the "One_Meter_Interval_Lane" class are *partially* contained (fully contained is a special case) in segments from the "Lane_Poly_3" class. The "position" association captures attachments of user content to level-three segments of "GEO_REPR", making content mapped to "GEO_REPR" accessible from "POST_REPR" . Further aspects such as road segments, traffic directions, lane change prohibitions, and traffic exchange directions, are discussed in [27].

2.3 Time

Time We now discuss the temporal characteristics of content. As mentioned above, content positions are captured at certain *time intervals*, which are organized in a containment hierarchy of temporal granularities, see the "TIME" cluster in Figure 1. Note that our time hierarchy consists both of *full* and *partial* containment relationships between temporal granularities, e.g., the relationship between hours and days (weeks and years) is full (partial). User positions are linked to their time intervals by the "time" association.

2.4 Queries

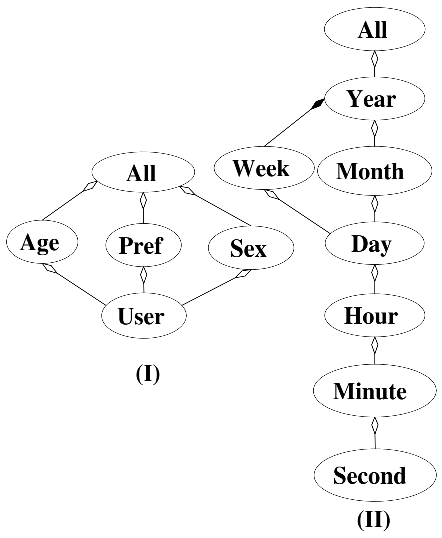
Analytical queries in LBS involve aggregations along multiple hierarchical dimensions, e.g., user content attachments will be aggregated along the "USER" , "LOCATION", and "TIME" dimensions. As mentioned above, content positions may be given with some uncertainty, and we thus need to evaluate aggregate queries over uncertain information. Consider the four sample queries below, each concerning *point* content at a *current* or *future* time (the focus of this paper) and involving some kind of uncertainty. Here we give a prose description of the query, while in Example 6.8 from Section 6.2 we give the expressions of the queries in terms of our probabilistic algebra. The queries are:

1. as the *minimum expectation*, how many users of "age less than 21", a, are *possibly* in "the eastbound lane of Main Street", l_{ms}, at the current time, t?
2. as the *average expectation*, what is an average age of the "male users", m, that will *possibly* be in "the second eastbound lane of I-90 highway between Moses Lake and Spokane", l_{90}, at the time "five minutes from now", t?
3. as the *maximum expectation*, how many users whose locations will be known with a high degree of confidence, will pass through "Stadium Way's lane towards the campus", l, during the hour between 10AM and 11AM, t?
4. (supposing some segments in "GEO_REPR" only partially contain one-meter interval segments in "POST_REPR"): what is the *maximum* age of the users that are *definitely* "between kilometer posts 45 and 46 of the eastbound lane of (Danish road) E45", l, at the current time, t?

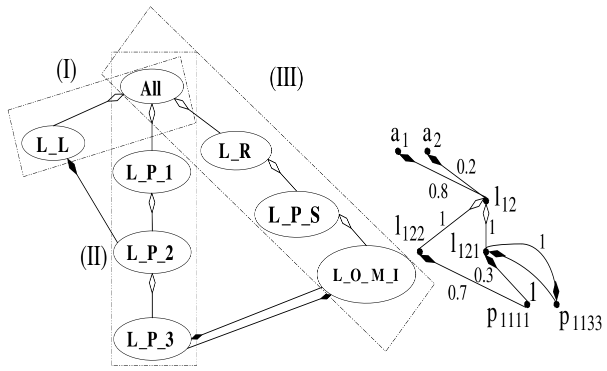
All these queries aggregate probabilistic data with varying degrees of uncertainty at either the current or a future time. They can all be formulated and evaluated in our framework, which improves the state-of-the-art by handling queries on DWs with probabilistic data and probabilistic relationships in the dimension hierarchies, including the use of pre-aggregation for efficient query processing.

3 The [OLAP$_{LBS}$] Model

We now briefly describe the data model from [27], which is the foundation for the probabilistic extension proposed in this paper. The model has constructs for defining both the *schema* (types) and the *data instances*.



(a) Dimension types $\mathcal{T}_u$ and $\mathcal{T}_t$



(b) Dimension type $\mathcal{T}_r$ and its instance

Figure 2: Dimension types

The schema of a cube is defined by a *fact schema* $\mathcal{S}$ that consists of a *fact type* $\mathcal{F}$ (cube name) and a set $\mathcal{D}$ of the *dimension types* $\mathcal{T}_i$ for each dimension.

A dimension type consists of a set $\mathcal{C}_{\mathcal{T}}$ of the *category types* $\mathcal{C}_j$ (dimension level types), a *relation* $\sqsubset_{\mathcal{T}}$ on $\mathcal{C}_{\mathcal{T}}$ specifying the hierarchical organization of the category types, and the special category types $\top_{\mathcal{T}}$ and $\perp_{\mathcal{T}}$ that denote the *top* and *bottom* category in the partial order, respectively. For example, a category type $\mathcal{C}$ may be used to model a level of *lane segments*. The transitive and irreflexive relation $\sqsubset_{\mathcal{T}}$, i.e., the partial order extended with equivalence, specifies the *partial* (including *full* as a special case) containment relationships among category types. The intuition is to specify whether members of a "child" category type have to be contained in a member of a "parent" category *fully*, e.g., segment levels from the same representation, or *partially*, e.g, segment levels from different representations. Next, a *subdimension type* of a dimension type is a set of its category types. Subdimension types of the same dimension type do not intersect except at $\top_{\mathcal{T}}$ category type. For example, a subdimension type is used to model a transportation infrastructure representation. The category types from the *same* (*different*) *subdimension type(s)* are related by full (partial) containment relationships.

Example 3.1. Figure 2(a) depicts dimension types $\mathcal{T}_u$ and $\mathcal{T}_t$. In addition, Figure 2(b) depicts a dimension type $\mathcal{T}_r$. The types capture the "USER", "TIME", and "LOCATION" clusters from Figure 1, respectively. Next, the type $\mathcal{T}_r$ has three subdimension types $\mathcal{T}_l$, $\mathcal{T}_g$, and $\mathcal{T}_p$, which capture "LN_REPR", "GEO_REPR", and "POST_REPR" , respectively. In the figure, the "boundary" of each type is a parralelogram and the types are labeled by (I), (II), and (III), respectively. In the figure, full (partial) containment category type relationships are given by empty (filled) rhombus-headed arrows. From these *direct* relationships we can deduce the *transitive* relationships between the category types. ▮

In the model *instances*, a *dimension* D consists of a set of *categories*. The *Type* function gives the corresponding type for dimensions and categories. A category C_j consists of a set of *dimension values* l_i. The transitive and irreflexive relation $\sqsubset$ on the union of all values, $\widehat{D}$, i.e., the partial order extended with equivalence, specifies the full or partial containment relationships of the values. For example, two values that model segments from the *same* (*different*) representation(s) are usually related by a full (partial) containment relationship. A special value $\top$ in each dimension *fully* contains every other value in the dimension.

Each relationship $l_1 \sqsubset l_2$ has an attached *degree of containment*, $d \in [0;1]$, written $l_1 \sqsubset_d l_2$. In a given dimension, the degrees have a unique interpretation, but different interpretations are possible. In the following definition, we present one such, *conservative*, interpretation (first introduced in [13]).

Definition 3.2. [**Safe degree of containment**] Given two dimension values l_1 and l_2 and a number $d \in [0;1]$, the notation $l_1 \sqsubset l_2 \wedge Deg_{saf}(l_1, l_2) = d$ (or $l_1 \sqsubset_d l_2$, for short) means that "l_2 contains at least $d \cdot 100\%$ of l_1". The special case of $d = 0$ means that "l_2 *may* contain l_1, and the size of contained part is unknown". We term *d safe degree of containment.* ∎

We abbreviate $l_1 \sqsubset_1 l_2 \wedge l_2 \sqsubset_1 l_1$ (equivalent values) by $l_1 \equiv l_2$. Transitive safe degrees are inferred according to the following rules. Given three dimension values, l_1, l_2, and l_3, and numbers $d_1 \in [0;1]$ and $d_2 \in [0;1)$:

1. *p-to-f transitivity:*
   $(l_1 \sqsubset_{d_1} l_2) \wedge (l_2 \sqsubset_1 l_3) \Rightarrow (l_1 \sqsubset_{d_1} l_3)$
   While l_3 may contain the part of l_1 that is not in l_2, the conditions of the property do not give us information on this. We infer only what we can guarantee: what is contained in l_2 is *also* contained in l_3.
2. *p-to-p transitivity:*
   $(l_1 \sqsubset_{d_1} l_2) \wedge (l_2 \sqsubset_{d_2} l_3) \Rightarrow (l_1 \sqsubset_0 l_3)$
   If l_1 is fully or partially contained in l_2 and l_2 is partially contained in l_3 then we can only infer that at least "nothing" of l_1 is contained in l_3. In other words, we infer that l_1 *may* be contained in l_3.

Finally, *subdimension* is an instance of a subdimension type.

Example 3.3. Suppose we are given subdimensions D_l, D_g, and D_p of the subdimension types $\mathcal{T}_l$, $\mathcal{T}_g$, and $\mathcal{T}_p$, respectively. Parts of the subdimensions are depicted in Figure 2(b). In the following, we show how to infer transitive partial containment relationships with *safe* degrees. In the subdimension D_g, we have values $l_{121} \in C_{L\_P\_3}$ and $l_{12} \in C_{L\_P\_2}$ such that $l_{121} \sqsubset_1 l_{12}$. Then, in the subdimension D_p, we have a value $p_{1111} \in C_{L\_O\_M\_I}$ such that $p_{1111} \sqsubset_{0.3} l_{121}$. Consequently, we infer that $p_{1111} \sqsubset_{0.3} l_{12}$. Again, we have $l_{121} \sqsubset_1 l_{12}$. Then, in the subdimension D_l, we have value $a_1 \in C_{L\_L}$ such that $l_{12} \sqsubset_{0.8} a_1$. Consequently, we infer that $l_{121} \sqsubset_0 a_1$, which means that l_{121} *may* be contained in a_1. Next, in the subdimension D_p, we have a value p_{1133} such that $p_{1133} \equiv l_{121}$. Then we have already inferred that $l_{121} \sqsubset_0 a_1$. Consequently, we infer that $p_{1133} \sqsubset_0 a_1$. ∎

A *multidimensional object* (cube) consists of a set of *facts* F that are mapped to each dimension, D_j, with a *fact-dimension relation*, $R_j \subseteq F \times D_j$. For a fact $f \in F$ and a dimension value $l \in D_j$, we define (1) a *covering fact-dimension relationship* $(f, l) \in R_j^c \subseteq R_j$, which is read as "*f covers l*", and (2) an *inside fact-dimension relationship* $(f, l) \in R_j^i \subseteq R_j$, which is read as "*f is inside l*". Thus, the full set of fact-dimension relationships is $R_j = R_j^c \cup R_j^i$. Next, we define three kinds of *fact characterizations*, or inferred fact-dimension relationships, written $f \rightsquigarrow^c l$, $f \rightsquigarrow^i l$, and $f \rightsquigarrow^i_m l$. The semantics of the first two characterizations coincides with that of the corresponding fact-dimension relationships. The third characterization means that *f may be inside l*.

4 Expected Degrees of Containment

In this section, we introduce a new interpretation for degrees of containment. The motivation for the new interpretation is as follows. Assume that we are given a dimension D with its set of categories and the relation on its dimension values $\sqsubset$. As mentioned in Section 3, with the *safe* degrees of containment, the notation $l_1 \sqsubset_d l_2$, where $l_1 \in \widehat{D}$, $l_2 \in \widehat{D}$, and $d \in [0;1]$ means that the value l_2 contains *at least* $d \cdot 100\%$ of the value l_1. The disadvantage of this approach is that inferred, transitive relationships between dimension values are very likely to receive a degree of containment equal to 0, because we infer only those degrees that we can guarantee, see Example 3.3. This makes the data too uncertain for practical use.

In order to make the transitive relationships more useful, we introduce the *expected* degrees of containment. Our approach is based on probability theory [3]. We consider each dimension value as an infinite set of points. We deal with the probabilistic events of the form “a value l_1 is contained in a value l_2”, which is equivalent to “any point in l_1 is contained in l_2”.

Definition 4.1. [Expected degree of containment] Given two dimension values l_1 and l_2 and a number $d \in [0;1]$, the notations $l_1 \sqsubset l_2 \wedge Deg_{exp}(l_1, l_2) = d$ (or $l_1 \sqsubset_d l_2$, for short) mean that “l_2 is expected to contain $d \cdot 100\%$ of l_1”, or, more formally, “l_1 is contained in l_2 with a probability of d”. We term d *expected degree of containment.* ▮

The formal definition is particularly useful for reasoning about transitivity of partial containment and fact characterizations. The rule of *transitivity of partial containment with expected degrees* is as follows:

$$\forall (l, l_1, \ldots, l_n, l') \in \widehat{D} \times \ldots \times \widehat{D}$$
$$(\bigwedge_{i=1}^{n} l \sqsubset_{d_i} l_i \wedge l_i \sqsubset_{d'_i} l' \Rightarrow l \sqsubset l' \wedge Deg_{exp}(l, l') = \sum_{i=1}^{n} d_i \cdot d'_i)$$

The idea behind the rule is explained next. We will use notation $P(e)$ for the probability of the event e. Let us first consider a special case of the rule, when $i = 1$, i.e., when there is only one, unique path between values l and l'. Then, the rule takes the following form:

$$\forall (l_1, l_2, l_3) \in \widehat{D} \times \widehat{D} \times \widehat{D}(l_1 \sqsubset_{d_1} l_2 \wedge l_2 \sqsubset_{d_2} l_3 \Rightarrow l_1 \sqsubset_{d_1 \cdot d_2} l_3)$$

First, $l_1 \sqsubset_{d_1} l_2$ means that $P(e_1) = d_1$, where e_1 is “l_1 is contained in l_2”. Second, $l_2 \sqsubset_{d_2} l_3$ means that $P(e_2) = d_2$, where e_2 is “l_2 is contained in l_3”. The conjunction of these two events, $e_1 \wedge e_2$, i.e., “l_1 is contained in l_3” is equivalent to $l_1 \sqsubset l_3$. Next, having assumed that the events e_1 and e_2 are independent, $P(e_1 \wedge e_2) = d_1 \cdot d_2$. This means that we have inferred the relationship $l_1 \sqsubset_{d_1 \cdot d_2} l_3$.

The general case of the rule allows n paths between l and l'. The ith path goes through a value l_i. Then, the event e, i.e., “l is contained in l'” is a disjunction of n disjoint events, $\bigvee_{i=1}^{n} e_i$, where e_i is “l is contained in l', given the ith path”. Thus, the general case of the rule is n applications of the rule’s special case. The ith application concerns an ith path and infers the probability $d_i \cdot d'_i$ of the event e_i. This means that the event e has the probability of $d = \sum_{i=1}^{n} d_i \cdot d'_i$, i.e., that there is a relationship $l \sqsubset_d l'$.

In order to produce the correct aggregates, i.e., to perform *correct aggregation*, a warehouse must consider all relevant aggregation paths between the source and destination category. Since no aggregation path is ignored during inferences of transitive partial containment relationships with expected degrees, the rule offers support for *correct aggregation*, which is missing from the analogous rule with safe degrees. A further support is offered by the rules for inferring fact characterizations (see Section 5).

Example 4.2. Continuing Example 3.3, we show how to infer transitive partial containment relationships with *expected* degrees. First, we demonstrate the support for correct aggregation. In the subdimension D_g, we have values $l_{121} \in C_{L\_P\_3}$, $l_{122} \in C_{L\_P\_3}$ and $l_{12} \in C_{L\_P\_2}$ such that $l_{121} \sqsubset_1 l_{12}$ and $l_{122} \sqsubset_1 l_{12}$. Then, in the subdimension D_p, we have a value $p_{1111} \in C_{L\_O\_M\_I}$ such that $p_{1111} \sqsubset_{0.3} l_{121}$ and $p_{1111} \sqsubset_{0.7} l_{122}$. In other words, we have two aggregation paths between values p_{1111} and l_{12}. Consequently, we ”sum up” the paths, i.e., infer that $p_{1111} \sqsubset_{0.3 \cdot 1 + 0.7 \cdot 1 = 1} l_{12}$. Second, we demonstrate the improvement in certainty of transitive relationships, compared to those obtained by the rule with *safe* degrees. Then, in the subdimension D_l, we have value $a_1 \in C_{L\_L}$ such that $l_{12} \sqsubset_{0.8} a_1$. Consequently, we infer that $l_{121} \sqsubset_{1 \cdot 0.8 = 0.8} a_1$. Note that the last relationship would have received a (much lower) *safe* degree of 0. ∎

5 Probabilistic Fact Characterizations

In this section, we introduce a new kind of fact characterizations. The motivation for this is as follows. Assume that we are given a dimension, D, with its set of categories and the relation on its dimension values, $\sqsubset$. Recall content attachments from the case study in Section 2. Such attachments record that a *user* is in a specific *location* at a given *time*. The fact characterizations described in Section 3 allow us to express positions of *static* content, which are usually certain. However, positions of *dynamic* content are usually uncertain. For example, a user location may be given by a wireless phone cell, which only approximately locates the user. Furthermore, a practical prediction algorithm would predict future user locations with some degree of uncertainty. In addition to this location uncertainty, we may also have user and time uncertainty. For example, we may be certain about a location, but uncertain about what user is at that position or we may not know the time. In order to capture these possibilities, we generalize the notion of fact characterization by defining a *probabilistic fact characterization*.

Our approach is based on probability theory [3]. We consider the probabilistic events of the form "a fact f covers (is inside) a value l". In the following, let p represent the probability; of course, $p \in [0; 1]$. We extend the definitions from Section 3 as follows.

Definition 5.1. [**Probabilistic fact-dimension relationships**] For a fact $f \in F$ and a dimension value $l \in \widehat{D}$, we define:

1. a *probabilistic covering fact-dimension relationship*, $(f, l, p_{min}, p_{max}) \in R^{c,p} \subseteq R$, which is read as "*f covers l with probability of at least* p_{min} and *of at most* p_{max}", and
2. a *probabilistic inside fact-dimension relationship*, $(f, l, p_{min}, p_{max}) \in R^{i,p} \subseteq R$, which is read as "*f is inside l with a probability of at least* p_{min} and *of at most* p_{max}".

The full set of fact-dimension relationships is $R = R^{c,p} \cup R^{i,p}$. ∎

Given an inside fact-dimension relationship, $(f, l, p_{min}, p_{max}) \in R^{i,p}$, p_{min} is a *lower* bound on the "true" probability of the relationship. Moreover, for a fact, f, and a category, C, any two events "f is inside l_1" and "f is inside l_2", where $l_1 \in C$ and $l_2 \in C$, are disjoint. For this reason, in an MO, we impose the following restriction on minimum probabilities: for any category, C, and any fact, f and given the restriction of $R^{i,p}$ on C and f, $R^{i,p}_{|C,f} = \{(f', l, p^l_{min}, p^l_{max}) | l \in C \wedge f' = f\}$, we require that $\sum_{l \in C} p^l_{min} \leq 1$. However, given an inside fact-dimension relationship, $(f, l, p_{min}, p_{max}) \in R^{i,p}$, p_{max} is a *higher* bound on the "true" probability of the relationship. For this reason, we do not impose an analogous restriction on maximum probabilities.

The exact probabilities of fact-dimension relationships may also be expressed. For example, the statement "f covers l with probability p" is expressed as $(f, l, p, p) \in R^p_c$. The non-probabilistic fact-dimension relationships are a special case of the probabilistic fact-dimension relationships, i.e., $(f, l) \in R_c$ is expressed as $(f, l, 1, 1) \in R^p_c$ and $(f, l) \in R_i$ is expressed as $(f, l, 1, 1) \in R^p_i$.

Next, we define two new kinds of fact characterizations, written $f \leadsto^c_{[p_{min}; p_{max}]} l$ and $f \leadsto^i_{[p_{min}; p_{max}]} l$. The non-probabilistic fact characterizations are a special case of the probabilistic characterizations, i.e., (1) $f \leadsto^c l$ is expressed as $f \leadsto^c_{[1;1]} l$, which is also read as "*f covers l for sure*", (2) $f \leadsto^i l$ is expressed as $f \leadsto^i_{[1;1]} l$, which is also read as "*f is inside l for sure*", and (3) $f \leadsto^i_m l$ is expressed as $f \leadsto^i_{[0;1]} l$, which is also read as f *is inside* l *with unknown probability*. In addition, $f \rightsquigarrow^{c}_{[0;1]} l$ is also read as f *covers* l *with unknown probability*.

The set R is stored in the data warehouse and the probabilistic fact characterizations are inferred when needed. For the inference, the warehouse uses the rules described in Sections 5.1 and 5.2. In essence, the rules provide a recursive definition of the notion of probabilistic fact characterization. Note that the rules are valid both with the *safe* and *expected* degrees of containment. For this reason, the notation for containment relationships used in the rules does not reflect the kind of degrees.

5.1 Basic Rules

In the following, we present the basic rules for inferring fact characterizations.

$\forall(f, l) \in F \times \widehat{D}$

1. $((f, l, p_{min}, p_{max}) \in R^{p}_{c} \Rightarrow f \rightsquigarrow^{c}_{[p_{min};p_{max}]} l)$
2. $((f, l, p_{min}, p_{max}) \in R^{p}_{i} \Rightarrow f \rightsquigarrow^{i}_{[p_{min};p_{max}]} l)$
   If a fact f is attached to and covers (is inside) a segment l with the probability of at least p_{min} and of at most p_{max} , then we can infer that f covers (is inside) l with the probability of at least p_{min} and of at most p_{max}.
3. $(f \rightsquigarrow^{c}_{[p_{min};p_{max}]} l \Rightarrow f \rightsquigarrow^{i}_{[p_{min};p_{max}]} l)$
   If a fact f *covers* a segment l with the probability of at least p_{min} and of at most p_{max}, then f is also *inside* l with the same probability. The idea behind the rule is as follows. If a piece of content *covers* a segment with some probability, i.e., between p_{min} and p_{max}, then it is possible to state that the piece is also *inside* that segment with the same or even greater probabilities. However, the data at hand, i.e., the probabilities that we can use for arguing, only allows us to record the lowest possible probabilities, i.e., those between p_{min} and p_{max}.

$\forall(f, l_1, l_2, \ldots, l_n, l) \in F \times D \times \ldots \times D$

4. $(l_1 \equiv l_2 \wedge f \rightsquigarrow^{i}_{[p_{min};p_{max}]} l_1 \Rightarrow f \rightsquigarrow^{i}_{[p_{min};p_{max}]} l_2)$
5. $(l_1 \equiv l_2 \wedge f \rightsquigarrow^{c}_{[p_{min};p_{max}]} l_1 \Rightarrow f \rightsquigarrow^{c}_{[p_{min};p_{max}]} l_2)$
   If two values are equivalent, then they characterize the same facts in the same way.

5.2 The Characterization Sum Rule

In the following, we present the most important rule for inferring fact characterizations, called the *characterization sum rule*. Among other things, the rule provides support for *correct aggregation*.

Definition 5.2. [**Characterization sum rule**] For any fact, f, and dimension values, l_1, l_2, …, l_n, and l, the following holds:

$$(\bigwedge_{i=1}^{n} (l_i \sqsubset_{d_i} l \wedge f \rightsquigarrow^{i}_{[p^{i}_{min};p^{i}_{max}]} l_i) \Rightarrow f \rightsquigarrow^{i}_{[p_{min};p_{max}]} l)),$$

where $p_{min} = \sum_{i=1}^{n} d_i \cdot p^{i}_{min}$ and $p_{max} = min(\sum_{i=1}^{n} d_i \cdot p^{i}_{max}, 1)$ ∎

In the following, we explain what the rule does. Let a fact f be inside values l_1, l_2, ..., l_n with probabilities at least p_1, p_2, ..., p_n, respectively. Then, if l_1, l_2, ..., l_n are contained in a value l, with minimum (with the "safe" approach) or expected (with the "expected" approach) sizes of the contained parts being $d_1, d_2, \ldots, d_n$, then f is also inside the containing value l with probability of at least $p^1_{min} \cdot d_1 + p^2_{min} \cdot d_2 + \ldots + p^n_{min} \cdot d_n$ and of at most $p^1_{max} \cdot d_1 + p^2_{max} \cdot d_2 + \ldots + p^n_{max} \cdot d_n$ (if the sum is lower than 1) or 1 (if the sum is equal to or greater than 1).

The basic idea behind the rule is that we obtain the probability for a fact characterization by summing up probabilities for that fact characterization obtained through n different aggregation paths. A more formal explanation is as follows. We use the notation $P(e)$ for the probability of the event e and $P(e \wedge e')$ for the conjunction of the events e and e'. First, let the event e_1 be "a piece of content is inside a segment l", i.e., is "$f \rightsquigarrow^i l$". We need to compute p_{min}, which is a lower bound on $P(e_1)$. Clearly, $P(e_1) = \sum_{i=1}^n P(e^i_2 \wedge e^i_3)$, where e^i_2 is "$f \rightsquigarrow^i l_i$" and e^i_3 is "$l_i \sqsubset l$". Since events e^i_2 and e^i_3 are independent, $P(e^i_2 \wedge e^i_3) = P(e^i_2) \cdot P(e^i_3)$. Next, $P(e^i_2) \geq p^i_{min}$ and $P(e^i_3) \geq d_i$ or $P(e^i_3) = d_i$, if d_i is an expected or safe degree, respectively. This means that $P(e_1) \geq \sum_{i=1}^n d_i \cdot p^i_{min}$. So, $p_{min} = \sum_{i=1}^n d_i \cdot p^i_{min}$. The case of p_{max}, i.e., the maximum probability that a piece of content is inside a segment l, is analogous except if the resulting probability is higher than 1, we "cut" it down to 1.

Since no aggregation path is ignored in the process of inferring fact characterizations, the characterization sum rule offers significant support for *correct aggregation*. Furthermore, if expected degrees are used for constructing an MO, the rule for inferring transitive relationships between dimension values (see Section 4) provides additional support. In particular, combined effect of these two rules is that a query engine may perform inferences on an MO in any order without losing any information, i.e., transitive relationships between values first, then fact characterizations, or in the reverse order.

Example 5.3. Given a dimension hierarchy from Figure 2(b), we exemplify the use of the characterization sum rule. Suppose our data warehouse has data on (uncertain) positions of a user in the kilometer-post representation, which are stored as $(f_1, p_{1111}, 0, 0.1) \in R^p_i$ and $(f_1, p_{1133}, 0.9, 1) \in R^p_i$. Then, the positions of the user in the link-node representation are deduced as follows. First, assuming that the degrees from Figure 2(b) are *expected degrees*, we infer the relationships $p_{1111} \sqsubset_{0.8} a_1$, $p_{1111} \sqsubset_{0.2} a_2$, $p_{1133} \sqsubset_{0.8} a_1$, and $p_{1133} \sqsubset_{0.2} a_2$. Second, by basic rule 2, we obtain the fact characterizations $f_1 \rightsquigarrow^i_{[0;0.1]} p_{1111}$ and $f_1 \rightsquigarrow^i_{[0.9;1]} p_{1133}$. Finally, by the characterization sum rule, we infer $f_1 \rightsquigarrow^i_{[p^1_{min};p^1_{max}]} a_1$ and $f_1 \rightsquigarrow^i_{[p^2_{min};p^2_{max}]} a_2$, where $p^1_{min} = 0.8 \cdot 0 + 0.8 \cdot 0.9 = 0.72$, $p^1_{max} = 0.8 \cdot 0.1 + 0.8 \cdot 1 = 0.88$, $p^2_{min} = 0.2 \cdot 0 + 0.2 \cdot 0.9 = 0.18$, and $p^2_{max} = 0.2 \cdot 0.1 + 0.2 \cdot 1 = 0.22$. ▮

6 The Algebra

In this section, we present a set of algebraic operators, i.e., algebra, that is a formal foundation for querying the data captured by our model. The operators allow us to formulate queries for probabilistic fact-dimension characterizations.

We base our algebra on the *deterministic* algebra from [20], which is proven to be at least as powerful as the relational algebra with aggregation functions [15]. The operators from [20] need to be extended to handle probabilistic aspects of our model. Intuitively, after the extension, our algebra will be at least as powerful as a probabilistic relational algebra (e.g., from [1]).

Let i range from 1 to n. For unary operators, we assume a single n-dimensional MO $M = \{\mathcal{S}, F, D_M, R_M\}$, where $D_M = \{D_i\}$ and $R_M = \{R_i\}$. For binary operators, we assume two n-dimensional MO's $M_1 = (\mathcal{S}_1, F_1, D_{M_1}, R_{M_1})$ and $M_2 = (\mathcal{S}_2, F_2, D_{M_2}, R_{M_2})$, where $D_{M_1} = \{D^1_i\}$, $D_{M_2} = \{D^2_i\}$, $R_{M_1} = \{R^1_i\}$, and $R_{M_2} = \{R^2_i\}$. In addition, we use the notation $\widehat{D}_i$ for the union of all the dimension values from D_i.

6.1 Selection Operator

The selection operator is used to select a subset of the facts in an MO based on a predicate.

Let $K = \{i, c\}$, where the symbols i and c stand for "inside" and "covering", respectively. The selection operator, σ, uses a predicate $q : \widehat{D}_1 \times \ldots \times \widehat{D}_n \times ([0;1] \times [0;1])^n \times K^n \rightarrowtail \{true, false\}$. Thus, the parameters of the predicate q are n dimension values, each from a different dimension, n intervals of probability values, and n "inside" or "covering" symbols. The resulting set of facts is:

$$\begin{aligned}
&F' = \{f \in F \mid \\
&\quad \exists(l_1, \ldots, l_n) \in \widehat{D}_1 \times \ldots \times \widehat{D}_n \\
&\quad (\exists([p^1_{min}; p^1_{max}], \ldots, [p^n_{min}; p^n_{max}]) \in ([0;1] \times [0;1])^n \\
&\quad\; (\exists(k_1, \ldots, k_n) \in K^n \\
&\quad\;\; (q(l_1, [p^1_{min}; p^1_{max}], k_1, \ldots, l_n, [p^n_{min}; p^n_{max}], k_n) \wedge \bigwedge_{j=1}^{n} f \rightsquigarrow^{k_j}_{[p^j_{min}; p^j_{max}]} l_j)))\}
\end{aligned}$$

We thus restrict the set of facts to those that are characterized by dimension values where q evaluates to $true$. This operator supports probabilistic covering/inside fact characterizations. Specifically, the operator allows us to formulate queries that select facts that are characterized (1) with given intervals of uncertainty, i.e., $[p^i_{min}; p^i_{max}]$ for a characterization by the dimension D_i, and (2) kind of characterization, i.e., inside, covering, or both by means of k_i for a characterization by the dimension D_i. In addition, we restrict the fact-dimension relations accordingly, while the dimensions and the fact schema stay the same.

Example 6.1. [**Selection operator**] Continuing Example 5.3, suppose that we would like to select *reliable* data on male users, $m \in C_{Gender}$, on a link, $a_1 \in C_{L\_L}$, at a future time, $t \in C_{Second}$. For this, the predicate q could be defined as follows:

$$\begin{aligned}
&q(l_1, [p^1_{min}; p^1_{max}], k_1, l_2, [p^2_{min}; p^2_{max}], k_2, l_3, [p^3_{min}; p^3_{max}], k_3) = true \Leftrightarrow \\
&(l_1 = m \wedge p^1_{min} = p^1_{max} = 1 \wedge k_1 = i) \wedge \\
&(l_2 = a_1 \wedge [p^2_{min}; p^2_{max}] \subseteq [0.5; 1] \wedge k_2 = i \wedge \\
&(l_3 = t \wedge p^3_{min} = p^3_{max} = 1 \wedge k_3 = i)
\end{aligned}$$

The predicate defines the *reliable* data as the fact characterizations such as: (1) in the USER and TIME dimension, the minimum and maximum probability equals to 1, (2) in the LOCATION dimension, the minimum probability is at least 0.5 and the maximum probability is any, i.e., up to 1.

Suppose we have characterizations $f_1 \rightsquigarrow^i_{[1;1]} m$ and $f_1 \rightsquigarrow^i_{[1;1]} t$ in the USER and TIME dimension, respectively. This means that the value of the predicate q depends on the characterizations in the LOCATION dimension. Since we have inferred the characterization $f_1 \rightsquigarrow^i_{[0.72;0.88]} a_1$, the fact f_1 would contribute to the result, i.e. $f_1 \in F'$. However, if we replace a_1 with a_2 in the query, then the fact f_1 would be outside the result, because of the characterization $f_1 \rightsquigarrow^i_{[0.18;0.22]} a_2$. As another example, we could select all data that is *unreliable* with respect to positioning, for instance, to remove it from a subsequent computation, as follows:

$$q(l_1, [p^1_{min}; p^1_{max}], k_1, l_2, [p^2_{min}; p^2_{max}], k_2, l_3, [p^3_{min}; p^3_{max}], k_3) = true \Leftrightarrow [p^2_{min}; p^2_{max}] \subseteq [0; 0.5] \blacksquare$$

6.2 Aggregate Formation Operator

The unary aggregate formation operator is used when applying aggregate functions to an MO. We assume a set of traditional aggregation functions, $H = \bigcup_{i=1}^{n}\{SUM_i, AVG_i, MIN_i, MAX_i\} \cup \{COUNT\}$. The $COUNT$ function works by considering fact-dimension relationships for all dimensions, while other functions "look up" the required data for the facts in the relevant fact-dimension relation. For example, SUM_1 finds its data in the fact-dimension relation R_1 and sums them.

In addition, the operator $Group : D_1 \times \ldots \times D_n \rightarrowtail 2^F$ is defined. In general, the operator groups the facts characterized by the same dimension values, i.e., $Group(l_1, \ldots, l_n) = \{f \in F \mid f \rightsquigarrow l_1 \wedge \ldots \wedge f \rightsquigarrow l_n\}$. Later in this section, we present more elaborate definitions of the grouping operator.

6.2.1 Aggregate Formation Operator Definition

Next, we restate a generic definition of the aggregate formation operator from [13], which is also suitable in our context of uncertain data. In the definition, we denote $(l_1, \ldots, l_n)$ and $Group(l_1, \ldots, l_n)$ by $\vec{l}$ and G, respetively. Also, we assume that $\vec{l} \in C_1 \times \ldots \times C_n$.

Definition 6.2. [**Aggregate formation operator**] Given a new (result) dimension D_{n+1} of a new (result) type $\mathcal{T}_{n+1}$, an aggregation function $h : 2^F \rightarrowtail D_{n+1}$ from the set H, and a set of grouping categories $\{C_i \in D_i, i = 1, \ldots n\}$, the *aggregate formation operator*, α, is defined as follows: $M' = \alpha[D_{n+1}, h, C_1, \ldots, C_n](M) = (\mathcal{S}' = (\mathcal{F}', \mathcal{D}'), F', D'_{M'}, R'_{M'})$, where

1. $\mathcal{F}' = 2^{\mathcal{F}}$
2. $\mathcal{D}' = \{\mathcal{T}'_i, i = 1, \ldots, n\} \cup \{\mathcal{T}_{n+1}\}$
3. $\mathcal{T}'_i = (\mathcal{C}'_i, \sqsubset'_{\mathcal{T}_i}, \perp'_{\mathcal{T}_i}, \top'_{\mathcal{T}_i})$
4. $\mathcal{C}'_i = \{\mathcal{C}_{ij} \in \mathcal{T}_i \mid \mathcal{C}_i \sqsubset_{\mathcal{T}_i} \mathcal{C}_{ij}\} \cup \{\mathcal{C}_i\}$
5. $\sqsubset'_{\mathcal{T}_i} = \sqsubset_{\mathcal{T}_i|_{\mathcal{C}'_i}}$ with $\perp'_{\mathcal{T}_i} = \mathcal{C}_i$ and $\top'_{\mathcal{T}_i} = \top_{\mathcal{T}_i}$
6. $F' = \{G \neq \emptyset\}$
7. $D' = \{D'_i, i = 1, \ldots, n\} \cup \{D_{n+1}\}$
8. $D'_i = (C'_{D'_i}, \sqsubset'_{D'_i})$
9. $C'_{D'_i} = \{C'_{ij} \in D_i \mid \mathcal{C}'_{ij} \in \mathcal{C}'_i\}$
10. $\sqsubset'_{D'_i} = \sqsubset_{D_i|_{D'_i}}$
11. $R'_{M'} = \{R'_i, i = 1, \ldots, n\} \cup \{R'_{n+1}\}$
12. $R'_i = \{(f', l_i) \mid \exists \vec{l}(f' = G\}, R'_{n+1} = \bigcup_{\vec{l}}\{(G \neq \emptyset, h(G)\}$ ∎

Thus, for every combination of dimension values $\vec{l} = (l_1, \ldots, l_n)$ in the given grouping categories, the aggregation function h is applied to the set of facts characterized by $\vec{l}$, i.e., to the group $G = Group(\vec{l})$, and the result is placed in the new dimension D_{n+1}.

The new facts from the set F' are of type $\mathcal{F}'$, which denotes *sets of the argument fact type*, and the resulting dimensions types from $\mathcal{D}'$ are obtained by restricting argument dimension types to the category types that are greater than or equal to the types of the grouping categories. The new dimension type $\mathcal{T}_{n+1}$ for the result is added to the set of dimension types.

The new set of facts F' consists of *sets* of the original facts, where original facts in a set share a combination of characterizing dimension values. The argument dimensions are restricted to the remaining category types, and the result dimension D_{n+1} is added. The fact-dimension relations for the argument dimensions now link sets of facts directly to their corresponding combination of dimension values, and the fact-dimension relation R'_{n+1} for the result dimension links sets of facts to the function results for these sets.

6.2.2 Grouping

In Section 5, we introduced probabilistic fact characterizations, which allows us to group facts with an arbitrary *degree of confidence*, i.e., with arbitrary requirements to the probabilities of the characterizations of the grouped facts. Next, we define different kinds of grouping, considering *inside* fact characterizations only. The cases of *covering* characterizations are analogous.

Definition 6.3. [**Grouping operators**] We define the following *grouping operators.*

1. *Degree-of-confidence grouping operator*, $Group_d$:

$$Group_d(l_1, \dots, l_n, [p^1_{min'}, p^1_{max'}] \dots, [p^n_{min'}, p^n_{max'}]) =$$
$$\{f \in F \mid \bigwedge_{k=1}^{n} f \rightsquigarrow^i_{[p^k_{min};p^k_{max}]} l_k \wedge [p^k_{min}; p^k_{max}] \subseteq [p^k_{min'}; p^k_{max'}]\}$$

2. *Conservative grouping operator*, $Group_c$:

$$Group_c(l_1, \dots, l_n) = Group_d(l_1, \dots l_n, [1; 1], \dots, [1; 1])$$

3. *Liberal grouping operator*, $Group_l$:

$$Group_l(l_1, \dots, l_n) = Group_d(l_1, \dots l_n, [0; 1], \dots, [0; 1])\blacksquare$$

In the *degree-of-confidence grouping*, a group is formed from the facts that belong to the group with a probability given by the parameters of $Group_d$ operator.

We define the following special cases of the operator. First, in the *conservative grouping*, a group is formed from the facts that *definitely* belong to the group. Since only precise data will be used in calculations and the remaining data discarded, this kind of grouping is useful for computing a "lower bound" for a query result, in the sense that the query result contains as little data as possible.

Second, in *liberal grouping*, a group is formed from the facts that *possibly* belong to the group. Liberal grouping can be used for computing an "upper bound" for a query result, in the sense that the query result contains as much data as possible, because all the data, both precise and imprecise, are taken into consideration. This means that our definition of conservative and liberal grouping corresponds to the general understanding of the terms introduced in [20].

Example 6.4. Continuing Example 5.3, suppose we also have a fact f_2 characterized as follows: $f_2 \rightsquigarrow^i_{[1;1]} m$, $f_2 \rightsquigarrow^i_{[1;1]} a_1$, and $f_2 \rightsquigarrow^i_{[1;1]} t$. Then, suppose we wish to aggregate the *certain* data to the level of C_{Gender}, C_{L_L}, and C_{Second}, and discard everything else, e.g., in order to decrease the chance of *overcounting*. Then, we will use the *conservative grouping* operator and obtain the groups $Group_c(m, a_1, t) = \{f_2\}$

and $\mathit{Group}_c(m, a_2, t) = \emptyset$. Next, if we wish to aggregate *all* data, e.g., in order to decrease the chance of *undercounting*, then we will use the *liberal grouping* operator and obtain the groups $\mathit{Group}_l(m, a_1, t) = \mathit{Group}_l(m, a_2, t) = \{f_1, f_2\}$. Finally, if we wish to aggregate the data given with a *reliable degree of confidence*, e.g., in order to balance the chances of *undercounting* and *overcounting*, then we will use a *degree-of-confidence grouping* operator, e.g., $\mathit{Group}_d(l, [0.5; 1])$. In this case, we obtain the groups $\mathit{Group}_d(m, a_1, t, [1; 1], [0.5; 1], [1; 1]) = \{f_1, f_2\}$ and $\mathit{Group}_d(m, a_2, t, [1; 1], [0.5; 1], [1; 1]) = \emptyset$. ▮

6.2.3 Aggregation Functions

In the following, we discuss aggregation functions. We assume a group

$$G = \{f_j \in F \mid \bigwedge_{k=1}^{n} f_j \rightsquigarrow^{i}_{[p^{k,j}_{min}; p^{k,j}_{max}]} l_k\}.$$

We start with the COUNT function, which counts *minimum expected*, *maximum expected*, *average expected*, *definite*, and *possible* number of facts that belong to the group G.

Definition 6.5. [**COUNT function**] Below we define different kinds of counts.

1. The *minimum expected count* is:

$$\mathit{COUNT}_{min}(G) = \sum_{j=1}^{N}(p^{1,j}_{min} \cdot \ldots \cdot p^{n,j}_{min})$$

   where N is the number of facts in the group G.

2. The *maximum expected count* is:

$$\mathit{COUNT}_{max}(G) = \sum_{j=1}^{N}(p^{1,j}_{max} \cdot \ldots \cdot p^{n,j}_{max})$$

3. The *average expected count* is:

$$\mathit{COUNT}_{avg}(G) = \frac{\mathit{COUNT}_{max}(G) + \mathit{COUNT}_{min}(G)}{2}$$

4. If G is formed according to the conservative grouping, then the *definite count* is:

$$\mathit{COUNT}_{def}(G) = N$$

5. If G is formed according to the liberal grouping, then the *possible count* is:

$$\mathit{COUNT}_{pos}(G) = N$$ ▮

Note that the procedure of computing the expected counts assigns degrees of group membership to facts, so with expected counts any grouping including the special cases of the liberal and conservative groupings may be considered weighted groupings.

Example 6.6. [**COUNT function**] Continuing Example 6.4, we consider the following three groups: $G_c = Group_c(m, a_1, t)$, $G_l = Group_l(m, a_1, t)$, and $G_d = Group_d(m, a_1, t, [1; 1], [0.5; 1], [1; 1])$.

Then, $COUNT_{min}(G_c) = 1 \cdot 1 \cdot 1 = 1$, $COUNT_{min}(G_l) = 1 \cdot 0.72 \cdot 1 + 1 \cdot 1 \cdot 1 = 1.72$, and $COUNT_{min}(G_d) = 0.72 + 1 = 1.72$. Also, $COUNT_{max}(G_c) = 1$, $COUNT_{max}(G_l) = 0.88 + 1 = 1.88$, and $COUNT_{max}(G_d) = 0.88 + 1 = 1.88$. ∎

As may be seen from Example 6.6, different $COUNT$ functions, in combination with different kinds of grouping, produce different values. For example, the difference between $COUNT_{min}(G_c)$, and $COUNT_{max}(G_l)$ is 88%. The former (latter) value is useful when the user wishes to maximally not overcount (undercount). In case the user wishes to obtain less extreme values, he/she may use an "intermediate" combination of a $COUNT$ function and grouping, such as $COUNT_{min}(G_l)$, $COUNT_{avg}(G_c)$, etc., that produce values between $COUNT_{min}(G_c)$ and $COUNT_{max}(G_l)$. Thus, the introduced means of querying flexibly adapt to concrete situation.

Due to space constraints, we only briefly discuss other aggregation functions. First, we consider the SUM function, which is, in essense, a generalization of the $COUNT$ function. Intuitively, the former sums arbitrary values of a measure, while the latter sums values of 1. Suppose, in an MO, the n^{th} dimension supplies data for the function. We assume that this dimension is *regular*, i.e. (1) there are only full containment relationships in the dimension hierarchy and (2) facts are only mapped to this dimension deterministically. Then, given the group G, we define the *minimum expected sum* by modifying the definition of the minimum expected count as follows:

$$SUM_{min}(G) = \sum_{j=1}^{N} (p_{min}^{1,j} \cdot \ldots \cdot p_{min}^{n-1,j} \cdot v(l_n, f_j))$$

where $v(l_n, f_j)$ is a numerical value assigned to a dimension value l such that $l \sqsubset l_n$ and $(f, l, 1, 1) \in R_n$. This way we sum the most precise data. Definitions of *maximum* or *average expected* and *possible* or *definite* sums can be obtained by modifying the definitions of the corresponding counts analogously.

Example 6.7. [**SUM function**] For example, suppose in our case study, we added a fourth dimension that captured weights of user cars. In addition, suppose (1) values $w_5 \in \widehat{D}_4$, $w_{2.5} \in \widehat{D}_4$, and $w_{1.75} \in \widehat{D}_4$, stand for 5, 2.5, and 1.75 tons, respectively, (2) $w_{2.5} \sqsubset w_5$ and $w_{1.75} \sqsubset w_5$, and (3) $(f_1, w_{1.75}, 1, 1) \in R_4$ and $(f_2, w_{2.5}, 1, 1) \in R_4$. Thus, $v(w_5, f_1) = 1.75$ and $v(w_{10}, f_2) = 2.5$. Then, we could find the *minimum expected sum* of weights of cars of users from the group $G'_l = Group(m, a_1, t, w_5)$ (see Example 6.6), as follows: $SUM_{min}(G'_l) = 1 \cdot 0.72 \cdot 1 \cdot 1 \cdot 1.75 + 1 \cdot 1 \cdot 1 \cdot 1 \cdot 2.5 = 1.26 + 2.5 = 3.76$. ∎

Second, we consider the AVG function. Given the group G, we define (different kinds of) the function as follows:

$$AVG_{mod}(G) = \frac{SUM_{mod}(G)}{COUNT_{mod}(G)}$$

where mod is one of the following: min, max, avg, def, and pos.

Finally, we consider the *MIN* function. Given the group G, we define the *possible* and *definite minimum* as follows:

$$MIN_{mod}(G) = min(\{v(l_n, f_j), j = 1, \ldots, N\})$$

where mod is either pos or def and min is a function that returns the minimum number from a set of numbers. Analogously with the $COUNT$ function, MIN_{pos} (MIN_{def}) is defined, if G is a *liberal* (*conservative*) group.

Example 6.8. [**Queries**] In this example, we express queries from Section 2.4 with the operators of our probabilistic algebra. The queries and their corresponding expressions are:

1. as the *minimum expectation*, how many users of "age less than 21", a, are *possibly* in "the eastbound lane of Main Street", l_{ms}, at the current time, t?
$$COUNT_{min}(GROUP_l(a, l_{ms}, t))$$
2. as the *average expectation*, what is an average age of the "male users", m, that will *possibly* be in "the second eastbound lane of I-90 highway between Moses Lake and Spokane", l_{90}, at the time "five minutes from now", t?
$$AVG_{avg}(GROUP_l(m, l_{90}, t))$$
3. as the *maximum expectation*, how many users whose locations will be known with a high degree of confidence, will pass through "Stadium Way's lane towards the campus", l, during the hour between 10AM and 11AM, t?
$$COUNT_{max}(GROUP_d(\top, l, t, [1;1], [0.75;1], [1;1]))$$
4. (supposing some segments in "GEO_REPR" only partially contain one-meter interval segments in "POST_REPR"): what is the *maximum* age of the users that are *definitely* "between kilometer posts 45 and 46 of the eastbound lane of (Danish road) E45", l, at the current time, t?
$$MAX_{def}(GROUP_c(\top, l, t))$$

6.3 Union Operator

The union operator is used to take the union of two MOs. Prior to defining the operator itself, we define two helper union operators, *union on dimensions* and *union on fact-dimension relations*.

The union operator is used to take the union of two MOs. Prior to defining the operator itself, we define two helper union operators, *union on dimensions* and *union on fact-dimension relations*.

In the next definition, we assume two dimensions of the same type $\mathcal{T}$: (1) D_1 with its set of categories C_{D_1} and relation $\models^{D_1}$ and (2) D_2 with its set of categories C_{D_2} and $\models^{D_2}$. Let us assume that $C_{D_1} = \{C_j^1, j = 1, \ldots, m\}$ and $C_{D_2} = \{C_j^2, j = 1, \ldots, m\}$.

Definition 6.9. The *union* operator on *dimensions*, $\bigcup^D$, is defined as follows:
$$D' = D_1 \overset{D}{\bigcup} D_2 = (C_{D'}, \sqsubset^{D'})$$
where $C_{D'} = \{C_j^1 \bigcup C_j^2, j = 1, \ldots, m\}$.

The relation $\sqsubset^{D'}$ is defined as follows:
$$\forall (l_1, l_2) \in (D_1 \cup D_2) \times (D_1 \cup D_2)(l_1 \in Desc(l_2) \wedge l_1 \sqsubset_d^{D'} l_2 \Leftrightarrow l_1 \sqsubset_{d_1}^{D_1} l_2 \vee l_1 \sqsubset_{d_2}^{D_2} l_2)$$
where $Desc(l_2)$ is a set of immediate predecessors of l_2 and d depends on d_1 and d_2. ▮

Stated less formally, given two dimensions of the same type, the union operator on dimensions performs set union on corresponding categories and builds a new relation on dimension values: there exists a *direct* relationship between two dimension values if there exists a *direct* relationship between the values in the first dimension, in the second dimension, or in both. The degree of containment for a resulting relationship may be determined in different ways. We discuss this issue later in this section. Note that only the degrees of containment for the *direct* relationships are found using these rules. The *indirect* relationships between values in the resulting dimension are inferred using our transitivity rules from Section 5.

In the next definition, we assume two fact-dimension relations: (1) R_1 that relates facts from a set F_1 with dimension values from a dimension D_1 and (2) R_2 that relates facts from a set F_2 with dimension values from a dimension D_2. The sets of facts are of the same fact type and the dimensions are of the same dimension type.

Definition 6.10. The *union* operator on *fact-dimension relations*, $\bigcup^R$, is defined as follows:

$$R' = R_1 \overset{R}{\bigcup} R_2 = \{(f, l, p'_{min}, p'_{max}) : (f, l, p^1_{min}, p^1_{max}) \in R_1 \vee (f, l, p^2_{min}, p^2_{max}) \in R_2\}$$

where p'_{min} and p'_{max} depend on p^1_{min}, p^1_{max}, p^2_{min}, and p^2_{max}.

Stated less formally, given two fact-dimension relations, relating facts and dimensions of the same type, the union operator on the relations builds new fact-dimension relation: the new relation relates a fact and a dimension value if the first relation, the second relation, or both the relations relate(s) the fact and the value from the first dimension, from the second dimension, or from both. The probabilities for a resulting relationship may be determined in different ways. We discuss this issue later in this section. Note that only *fact-dimension relations* are found using these rules. The *fact characterizations* are inferred using the rules from Section 5.

Definition 6.11. Consider two n-dimensional MO's with the same fact schema, i.e., $\mathcal{S}_1 = \mathcal{S}_2$. The *union* operator on *MOs*, $\bigcup$, is defined as:

$$M' = M_1 \bigcup M_2 = (\mathcal{S}', F', D'_{M'}, R'_{M'})$$

where

1. $\mathcal{S}' = \mathcal{S}_1$,
2. $F' = F_1 \bigcup F_2$,
3. $D'_{M'} = \{D^1_i \bigcup^D D^2_i, i = 1, \ldots, n\}$,
4. $R'_{M'} = \{R^1_i \bigcup^R R^2_i, i = 1, \ldots, n\}$.

Stated less formally, given two MO's with common fact schemas, the union operator combines dimensions and fact-dimension relations with the help of the $\bigcup^D$ and $\bigcup^R$ operator, respectively.

Next, we refer to Definition 6.9 and consider the dependency of the resulting degrees of containment d on the given ones d_1 and d_2. We present several ideas on the issue. First, if a user believes that dimension hierarchies from the first MO are *less precise* than those from the second MO, then d should be equal to d_2. Second, if a user wants to decrease the chance of *overcounting (undercounting)* facts in the resulting MO, then d should be equal to the *minimum (maximum)* of d_1 and d_2. Finally, d may be computed by applying weights to d_1 and d_2 and summing the weighted values. Note that in any case normalization of the resulting degrees (probabilities) may be necessary.

Next, we refer to Definition 6.10 and consider the dependency of the resulting probabilities p'_{min} and p'_{max} on the given ones p^1_{min}, p^1_{max}, p^2_{min}, and p^2_{max}. We present several ideas on the issue. First, if the facts-dimension relationships from the first MO are *less precise* than those from the second MO, then p'_{min} and p'_{max} should be equal to p^2_{min} and p^2_{max}, respectively. Second, if a user wants to decrease the chance of *overcounting (undercounting)* facts in the resulting MO, then p'_{min} and p'_{max} should be equal to the *minimum (maximum)* of p^1_{min}, p^2_{min} and p^1_{max}, p^2_{max}, respectively. Finally, p'_{min} and p'_{max} may be computed by applying weights to p^1_{min}, p^1_{max}, p^2_{min}, and p^2_{max}, and combining the weighted values. Note that in any case normalization of the resulting probabilities may be necessary.

6.4 Other Operators

Given argument MOs, *difference*, *projection*, and *identity-based join* operators are not meant to transform the probabilities of the fact characterizations or the degrees of containment in the dimensions. The only requirement is to preserve the probabilities. Therefore, we define the probabilistic version of these operators as their deterministic counterparts in [20] except the probabilistic operators take probabilistic MOs as arguments and produce probabilistic MOs in the result.

7 Query Processing with Pre-Aggregation

7.1 Minimum Expected Count

Pre-aggregation means pre-computing aggregates for one category and using them to compute aggregates for higher categories. In this section, we present a method for computing the minimum expected count, $COUNT_{min}$. We assume that all the fact characterizations in our data warehouse are inside characterizations. Therefore, the computations are based on the characterization sum rule from Section 5.2. In this section, we consider one-dimensional MOs only, but our method may be easily extended to multidimensional MOs.

Suppose C_1 is the pre-aggregated category and we need to compute $COUNT_{min}(G)$, where $G = Group_d(l, p_{min}, p_{max})$ such that $l \in C_2$ and $C_1 \sqsubset_{\mathcal{T}} C_2$. Next, suppose that $K = \{l_i : l_i \in C_1 \wedge l_i \sqsubset_{d_i} l\}$, i.e., K is the set of all the descendants of l at the pre-aggregated category C_1. We assume that the pre-aggregated values are $COUNT_{min}(G_i)$, where $G_i = Group_d(l_i, [0; 1])$ for each $l_i \in K$.

We distinguish between two cases: (1) $[p_{min}; p_{max}] = [0; 1]$ and (2) $[p_{min}; p_{max}] \subset [0; 1]$.

Case (1) In this case, the group G is formed according to the liberal grouping, i.e, all the facts regardless of their minimum probabilities are included in the group. Therefore, we *precisely* compute the minimum expected count as follows:

$$COUNT_{min}(G) = \sum_{i=1}^{|K|} d_i \cdot COUNT_{min}(G_i)$$

This computation corresponds directly to the characterization sum rule from Section 5.2.

Case (2) In this case the group G is a subset of the liberal group $Group(l, [0; 1])$, which we denote as G'. So it is impossible to come up with the precise value of the minimum expected count. However, given only the pre-aggregate values, we have no precise information on which facts belong to the group G. For this reason, we come up with a method for estimating the difference between the minimum expected counts for the groups G' and G, which we present next.

The method is based on the idea that at a given aggregation level the data has a certain uncertainty level. For a given category, this uncertainty level is described by the probability distribution of the minimum and maximum probabilities of fact characterizations by the values from this category. For example, we may say for values from the category C_2, how many facts (in percentage) are characterized with the probability of at least 0.5 and of at most 1. Such distributions may be computed from the historical data. A more formal definition of the distribution is presented next.

Definition 7.1. **[Uncertainty distribution]** For a category being aggregated, C_2, we maintain an *uncertainty distribution*, $P : [0;1] \times [0;1] \rightarrowtail [0;1]$. For a given pair (p_1, p_2), $P(p_1, p_2)$ is a probability that a fact is characterized by a value from C_2 with the probability of at least p_1 and of at most p_2. ▮

In order to make our method more practical, we assume that an uncertainty distribution is a finite discrete distribution, i.e., it is defined on a finite set of pairs (p_1, p_2), e.g., on the following set:

$$S = \{(0,0), (0,0.1), (0,0.2), \ldots, (0,1), (0.1,0.1), \ldots, (1,1)\}$$

Having defined the uncertainty distribution, we also assume that the facts in our warehouse may only be characterized with a pair of probabilities that belong to the set S. Obviously, this is a practical assumption.

Getting back to the actual computation of the minimum expected count in case (2), given an uncertainty distribution $P : S \rightarrowtail [0;1]$:

1. We define a set $S' = \{(p_1, p_2) \in S | p_1 \geq p_{min} \wedge p_2 \leq p_{max}\}$. The set contains only those pairs of minimum and maximum probabilities that the facts from the group G will be characterized with.
2. Thus, the probability for a fact to belong to the group G is $P(G) = \sum_{(p_1,p_2) \in S'} P(p_1, p_2)$.
3. Next, if a number of facts in the group G' is N, then we may assume that for each fact its expected contribution to the group G' is $\frac{COUNT_{min}(G')}{N}$.
4. This means that for each fact from the group G' its expected contribution to the group G is $P(G) \cdot \frac{COUNT_{min}(G')}{N}$.
5. By summing up the expected contributions of all the facts, we approximately compute the minimum expected count as follows:

$$COUNT_{min}(G) \approx P(G) \cdot COUNT_{min}(G'),$$

   where (just like in case (1)) $COUNT_{min}(G') = \sum_{i=1}^{|K|} d_i \cdot COUNT_{min}(G_i)$.

Next, we discuss the error of the method. First, when no facts belong to the group G, i.e., when $COUNT_{min}(G) = 0$, maximal overcounting occurs, and the error is equal to $P(G) \cdot COUNT_{min}(G')$. Second, when all the facts belong to the group G, i.e., when $COUNT_{min}(G) = COUNT_{min}(G')$, maximal undercounting occurs, and the error is equal to $(1 - P(G)) \cdot COUNT_{min}(G')$. It is possible to inform the user about the amount of the undercounting *prior* to query processing: the undercounting will amount up to $(1 - P(G))$ % of the "true" value.

7.2 Maximum and Average Expected Counts

As in Section 7.1, suppose C_1 is a pre-aggregated category and we need to compute $COUNT_{max}(G)$ and $COUNT_{avg}(G)$, where $G = Group_d(l, p_{min}, p_{max})$ such that $l \in C_2$ and $C_1 \vDash_{\mathcal{T}} C_2$. Next, suppose that $K = \{l_i : l_i \in C_1 \wedge l_i \vDash_{d_i} l\}$, i.e., K is a set of all the descendants of l at the pre-aggregated category C_1.

If we inspect the characterization sum rule from Section 5.2, we notice that unlike the minimum probability the maximum probability of a fact f being characterized by the value l is obtained not only by summing up the maximum probabilities of the same fact being characterized by the values from the set K. In addition to the summation, the resulting maximum probability must be "cut" down to 1. This means that compared to the case of the minimum expected count, with the maximum expected count, we have an additional problem of estimating the values to be "cut". We leave this estimation for the future work. Consequently, we leave the complete methods for computing the maximum and average expected counts for the future work.

8 Conclusions and Future Work

Motivated by the increasing use of location-based data warehouses (LBDWs) in industry, and the need to handle complex, dynamic, uncertain multidimensional data in such LBDWs, we have proposed a powerful, probabilistic data model that is able to capture the complexity of such data. The model provides a foundation for handling complex, hierarchical, and uncertain data, e.g., LBS data such as transportation infrastructures and the attached static and dynamic content, for example speed limits and vehicle positions. The paper also formally defines a set of algebraic query operators that support querying of the afore-mentioned data. Finally, the paper outlines a real-world case study, based on our collaboration with a leading Danish vendor of location-based services.

To our knowledge, this paper is the first to address the management of complex, hierarchical, and uncertain location-based data. More specifically, this paper is the first to describe a formal multidimensional data model, a query algebra, and query processing techniques for such data.

In future work, it is interesting to develop the theoretical framework by generalizing the expected degree of containment approach to intervals. On the implementation side, the most interesting directions concern pre-aggregation issues such as methods for using pre-aggregation to compute a wider range of aggregate functions, developing probabilistic pre-aggregation techniques, and finally developing pre-aggregation techniques for dynamic content such as user positions, including an embedded probabilistic position prediction method.

Acknowledgements

This work was supported in part by the Danish Centre for IT Research, grant no. 216.

References

[1] D. Barbara, H. Garcia-Molina, and D. Porter. The Management of Probabilistic Data. *Transactions on Knowledge and Data Engineering* 4(5):487–502, 1992.

[2] R. Cavallo and M. Pittarelli. The Theory of Probabilistic Databases. In *Proceedings of the Thirteenth International Conference on Very Large Databases*, pp. 71–81, 1987.

[3] M. H. DeGroot and M. J. Schervish. Probability and Statistics. *Addison-Wesley*, 816 pp., 2002.

[4] R. Cheng, D. V. Kalashnikov, and S. Prabhakar. Querying Imprecise Data in Moving Object Environments. *Transactions on Knowledge and Data Engineering* 16(9):1112–1127, 2004.

[5] R. Cheng, D. V. Kalashnikov, and S. Prabhakar. Evaluating Probabilistic Queries over Imprecise Data. In *Proceedings of the 2003 ACM SIGMOD International Conference on Management of Data*, pp. 551-562, 2003.

[6] N. Dalvi and D. Suciu. Efficient Query Evaluation on Probabilistic Databases. In *Proceedings of the Thirtieth International Conference on Very Large Data Bases*, pp. 864–875, 2004.

[7] C. Dyreson. Information Retrieval from an Incomplete Data Cube. In *VLDB'96, Proceedings of 22th International Conference on Very Large Data Bases*, pp. 532–543, 1996.

[8] Euman. *www.euman.com* (in Danish). Current as of 03/17/2005.

[9] E. Gelenbe and G. Hebrail. A Probability Model of Uncertainty in Data Bases. In *Proceedings of the Second International Conference on Data Engineering*, pp. 328-333, 1986.

[10] R. H. Güting, M. Böhlen, M. Erwig, C. S. Jensen, N. Lorentzos, M. Schneider, and M. Vazirgiannis. A Foundation for Representing and Querying Moving Objects. *ACM Transactions on Database Systems* 25(1):1–42, 2000.

[11] C. Hage, C. S . Jensen, T. B. Pedersen, L. Speičys, and I. Timko. Integrated Data Management for Mobile Services in the Real World. In *VLDB 2003, Proceedings of 29th International Conference on Very Large Data Bases*, pp. 1019-1031, 2003.

[12] R. Jacobson. Microsoft SQL Server 2000 Analysis Services Step By Step. *Microsoft Press*, 400 pages, 2000.

[13] C. S. Jensen, A. Kligys, T. B. Pedersen, and I. Timko. Multidimensional Data Modeling for Location-Based Services. *The Very Large Databases Journal* 13(1):1–21, 2004.

[14] R. Kimball et al. The Data Warehouse Lifecycle Toolkit. *Wiley Computer Publishing*, 800 pages, 1998.

[15] A. Klug. Equivalence of Relational Algebra and Relational Calculus Query Languages Having Aggregate Functions. *Journal of ACM* 29(3):699-717, 1982.

[16] B. R. Moole. A Probabilistic Multidimensional Data Model and Algebra for OLAP in Decision Support Systems. In *Proceedings of IEEE SoutheastCon*, pp. 18–30, 2003.

[17] C. Murray. Oracle spatial user guide and reference, Release 9.2. *Oracle Corporation*, 2002.

[18] NCHRP. A Generic Data Model for Linear Referencing Systems. *Transportation Research Board, Washington, DC*, 28 pp., 1997.

[19] Oracle Corporation. Oracle9i Business Intelligence. In: *otn.oracle.com/products/bi/content.html*. Current as of 03/17/2005.

[20] T. B. Pedersen, C. Dyreson, and C. Jensen. A Foundation for Capturing and Querying Complex Multidimensional Data. *Information Systems* 26(5):383–423, 2001.

[21] T. B. Pedersen and N. Tryfona. Pre-aggregation in Spatial Data Warehouses. In *Proceedings of Advances in Spatial and Temporal Databases, 7th International Symposium, SSTD 2001*, pp. 460–478, 2001.

[22] D. Papadias, J. Zhang, N. Mamoulis, and Y. Tao. Query Processing in Spatial Network Databases. In *VLDB 2003, Proceedings of 29th International Conference on Very Large Data Bases*, pp. 802–813, 2003.

[23] S. Šaltenis, C. S. Jensen, S. T. Leutenegger, and M. A. Lopez. Indexing the Positions of Continuously Moving Objects. In *Proceedings of the 2000 ACM SIGMOD International Conference on Management of Data*, pp. 331–342, 2000.

[24] L. Speičys, C. S. Jensen, and A. Kligys. Computational Data Modeling for Network-Constrained Moving Objects. In *ACM-GIS 2003, Proceedings of the Eleventh ACM International Symposium on Advances in Geographic Information Systems*, pp. 118–125, 2003.

[25] J. Sun, D. Papadias, and Y. Tao. Querying about the Past, the Present and the Future in Spatio-Temporal Databases. In *Proceedings of the 20th International Conference on Data Engineering, ICDE 2004*, pp. 202–213, 2004.

[26] Y. Tao, G. Kollios, J. Considine, F. Li, and D. Papadias. Spatio-Temporal Aggregation Using Sketches. In *Proceedings of the 20th International Conference on Data Engineering, ICDE 2004*, pp. 214–226, 2004.

[27] I. Timko and T. B. Pedersen. Capturing Complex Multidimensional Data in Location-Based Data Warehouses. In *12th ACM International Workshop on Geographic Information Systems, ACM-GIS 2004*, pp. 147-156, 2004.

[28] G. Trajcevski. Probabilistic Range Queries in Moving Object Databases with Uncertainty. In *Proceedings of Third International ACM Workshop on Data Engineering for Wireless and Mobile Access, MobiDE'2003*, pp. 39-45, 2003.

[29] G. Trajcevski, O. Wolfson, F. Zhang, and S. Chamberlain. The Geometry of Uncertainty in Moving Objects Databases. In *Proceedings of EDBT*, pp. 233–250, 2002.

[30] D. Zhang, V. J. Tsotras, and D. Gunopulos. Efficient Aggregation over Objects with Extent. In *Proceedings of the Twenty-First ACM SIGMOD-SIGACT-SIGART Symposium on Principles of Database Systems, PODS'2002*, pp. 121–132, 2002.

[31] D. Zhang, D. Gunopulos, V. J. Tsotras, and B. Seeger. Temporal and Spatio-Temporal Aggregations over Data Streams Using Multiple Time Granularities. *Information Systems* 28(1–2): 61–84, 2003.