Erhalt von Imperfektion in einem Data Warehouse

Heiko Schepperle, Andreas Merkel, Alexander Haag

{schepperle,merkela,haag}@ipd.uni-karlsruhe.de

Abstract: Üblicherweise werden in einem Data-Warehouse-System imperfekte Daten während des ETL-Prozesses bereinigt. Nicht jede Imperfektion kann allerdings als Qualitätsmangel gewertet werden, sondern hat für die Anwendung eine Bedeutung. Wir passen deswegen ein Kontextmodell an, welches das relationale Datenmodell für die unscharfe Klassifikationen mit Hilfe linguistischer Variablen erweitert. Wir zeigen zunächst, wie das Kontextmodell innerhalb des multidimensionalen Datenmodells verwendet werden kann, um Dimensionen mit unscharfen Klassifikationen zu modellieren. Danach schlagen wir einen erweiterten Summierbarkeitsbegriff und dafür angepasste Aggregationsoperatoren vor. Anschließend zeigen wir eine erste Erweiterung des Common-Warehouse-Metamodells, mit deren Hilfe die in einem Data-Warehouse-System beteiligten Werkzeuge Informationen über die Art der Imperfektion solcher Dimensionen automatisiert austauschen können.

1 Bewusster Umgang mit imperfekten Informationen

In der Datenbankwelt wird großen Wert darauf gelegt, dass eine gegebene Datenbasis einer einheitlichen Terminologie gehorcht, frei von Widersprüchen ist und jeder Wert genau die tatsächliche Welt repräsentiert. Dies ist auch in einem Data-Warehouse-System nicht anders. Während des ETL-Prozesses werden beispielsweise Widersprüche bereinigt, unterschiedliche Granularitäten und unterschiedliche Begriffswelten vereinheitlicht.

Ein solches Vorgehen beruht auf der Annahme, dass derartige Imperfektionen ein Problem der Datenerfassung sind und die Anwendung nur stören, wenn sie nicht beseitigt werden. Diese Annahme muss aber keineswegs immer gelten. Sind die Ursachen nicht ausreichend bekannt, ist unklar, wie eine Bereinigung aussehen könnte, oder liefert der Erhalt der Imperfektion sogar einen echten Mehrwert, dann sollte sich die Anwendung der Imperfektionen bewusst sein. Imperfektionen können auf unsicheren, unvollständigen oder ungenauen Informationen beruhen. Es werden Meinungen, Ansichten und Prognosen aufgestellt, die nicht unbedingt sicher eintreten. Natürlichsprachliche Begriffe, wie z.B. *kurz*, *günstig* oder *kurz bevor* gehorchen keiner allgemein anerkannten Definition. Möchte man solche Informationen ebenfalls in einer Datenbank verwalten, so muss man das zugrunde liegende Datenmodell erweitern.

1.1 Szenario

Im Verkehrsbereich müssen viele Messdaten aggregiert werden. Daher sind Data Warehouses Kandidaten für die Auswertung von Messdaten über grobe Zeitraster, z.B. zur Erstellung von Ganglinien. Für den Einsatz z.B. in einer Verkehrsleitzentrale sollte ein Data Warehouse allerdings auch Anfragen auf aktuellen Messdaten erlauben, um einer Anwendung eine kurzfristige Reaktion auf Entwicklungen zu ermöglichen. Die dabei zu erfassenden Messdaten können durchaus mit Imperfektion behaftet sein.

Beispielsweise formulieren Anrufer, die freiwillig als „Staumelder“ für bestimmte Rundfunksender arbeiten, eine natürlichsprachliche Verkehrsbeschreibung. Diese Beschreibung umfasst bestimmte Verkehrszustände von Strecken, z.B. die mit unscharfen Begriffen beschriebenen Zustände *frei*, *lebhaft*, *stockend* oder *Stau*. Desweiteren werden bestimmte Eigenschaften eines Staus, z.B. Staulänge und Stauposition, beschrieben. Diese mündlich überlieferten Meldungen basieren auf Schätzungen und sind deswegen ungenau. Handelt es sich bei dem Anrufer um eine bislang unbekannte Person, so wird dieser Meldung geringeres Vertrauen entgegegengebracht, als wenn es sich z.B. um eine Meldung einer Polizeidienststelle handelt, die die voraussichtliche Dauer und den voraussichtlichen Umfang einer durch einen Unfall hervorgebrachten Störung bekannt gibt. Unterschiedliches Vertrauen lässt sich als unterschiedliche Sicherheit (oder eben als unsicher) deuten. Unsere These ist nun, dass es für eine sachgerechte Nutzung unerlässlich ist, dass die Imperfektionen nach ihrer Art in einem Informationssystem gespeichert und erhalten werden.

Basierend auf einem solchen Informationssystem müsste ein Data Warehouse auch Anfragen nach imperfekten Informationen oder Aggregationsanfragen mit imperfekten Kriterien beantworten: Beispielsweise könnte eine Verkehrsleitzentrale nach der durchschnittlichen Geschwindigkeit auf Strecken fragen, die momentan als $stockend$ eingestuft werden, und diese mit dem Durchschnitt des letzten Jahres vergleichen. Solche Informationen können wichtige Kennzahlen darstellen, um bestimmte Entscheidungen zu treffen, z.B. über den veränderten Einsatz von Verkehrsleitsystemen.

1.2 Imperfektion in einem Data Warehouse

Für imperfekte Daten gibt es keinen allgemein akzeptierten Oberbegriff. Statt von Imperfektion („imperfection“) wird auch von Unvollkommenheit gesprochen, ebenso werden Unsicherheit („uncertainty“) oder Unschärfe („fuzziness“ oder „vagueness“) verwendet. Im Weiteren wird immer der Begriff „Imperfektion“ als Oberbegriff verwendet.

Man spricht von *unscharfen Daten*, wenn bei der Beschreibung ein natürlichsprachliches Konzept verwendet wird, für das keine allgemein anerkannte präzise Definition existiert. *Unsichere Daten* sind immer solche, deren Eintreten nicht völlig gewiss ist. Die Sicherheit des Eintretens lässt sich deshalb z.B. durch ein Wahrscheinlichkeitsmaß ausdrücken. *Ungenaue Daten* spiegeln ein mit Gewissheit aufgetretenes Ereignis wider, ungewiss ist aber ihr genauer Wert, etwa weil sie einer statistisch messbaren Abweichung folgen, z.B. Messdaten aus Messgeräten oder Statistiken.

Innerhalb eines Data-Warehouse-Systems (DWS) erfolgt die Bereinigung im Rahmen des ETL-Prozesses. Integriert man eine Datenquelle mit imperfekten Daten in ein DWS, so muss man steuern können, welche Imperfektionen während des ETL-Prozesses bereinigt und welche erhalten bleiben sollen. Letztere gelangen somit in das Data Warehouse und die übergeordneten Analyseapplikationen.

Dieser Fall wirft einige Fragen auf. Beispielsweise muss im Einzelnen geklärt werden, welche Arten von Imperfektion erhalten werden sollen und wie diese Imperfektion während des ETL-Prozesses überhaupt erhalten werden kann. Es stellt sich die Frage, wie imperfekte Daten gespeichert werden sollen und welche Vorkehrungen getroffen werden müssen, dass trotz der Imperfektion noch sinnvoll aggregiert werden kann. Ebenfalls offen ist, wie imperfekte Daten aus einem Data Warehouse sinnvoll gegenüber einem Analyseanwender präsentiert werden können.

1.3 Verwandte Arbeiten

In [RB92] wird die Aggregation von imperfekten Daten diskutiert. Zur Gruppierung und damit auch zur Klassifikation von unscharfen Werten werden allerdings auf α-Schnitten aufbauende Partitionen verwendet, wodurch Werte mit geringer Zugehörigkeit nicht in die Gruppierung einfließen. Ein anderer Ansatz, der sich vor allem mit der Aggregation ungenauer Daten beschäftigt, wird in [PJD99] vorgestellt. Dabei wird beschrieben, wie sich Aggregationsanfragen beantworten lassen, die nicht in derselben Granularität vorliegen. Die darin vorgeschlagene gewichtete Antwortmöglichkeit verwendet zwar die Zugehörigkeit für die Gewichtung, verzichtet aber auf die Eigenschaft, dass die Zugehörigkeiten sich zu eins summieren, wodurch keine Summierbarkeit gewährleistet ist. Eine umfassendere Beschreibung der Arbeiten im Bereich multidimensionale Datenbanken und Imperfektion ist in [DPJ03] zu finden. Darin wird unter anderem die Strategie vorgeschlagen, die Imperfektion in den aggregierten Werten zu erhalten, die wir auch in unserer Arbeit verfolgen. Für die unscharfe Klassifikation wird Summierbarkeit durch Hinzufügen zusätzlicher Kategorien und Aufteilung in sichtbare und unsichtbare Einheiten erreicht. Dieser Ansatz unterstützt allerdings nur den Fall, dass Kategorien zu mehreren Oberkategorien jeweils die Zugehörigkeit eins besitzen. Zugehörigkeiten zwischen null und eins werden nicht berücksichtigt.

Es gibt also noch keine Lösung, die auch geringe Zugehörigkeiten bei der Aggregation berücksichtigt, Aussagen über die Qualität der Aggregation erlaubt und auch alle Zugehörigkeitswerte im Intervall von null bis eins unterstützt.

1.4 Überblick

Einige der offenen Fragen aus Abschnitt 1.2 will dieser Beitrag beantworten. Wir gehen dabei folgendermaßen vor: Zunächst passen wir ein Modell zur Abbildung von imperfekten Daten an die Anforderungen in einem Data Warehouse an. Wir konzentrieren uns

dazu in dieser Arbeit auf ein Kontextmodell und zeigen, wie es im multidimensionalen Datenmodell verwendet werden kann, um Dimensionen mit unscharfer Klassifikation zu modellieren. Anschließend untersuchen wir die Qualität der Aggregierung, die mit einem solchen Modell noch erreicht werden kann. Hierzu werden ein erweiterter Summierbarkeitsbegriff vorgeschlagen und angepasste Aggregationssoperatoren beschrieben. Danach wird gezeigt, wie sich unser Ansatz mit Hilfe einer Erweiterung des Common Warehouse Metamodells so beschreiben lässt, dass am Data-Warehouse-Prozess beteiligte Werkzeuge Informationen über die Art der Imperfektion solcher Dimensionen automatisiert austauschen können.

2 Imperfekte Klassifikation durch Kontexte

Das Kontextmodell ist eine Erweiterung des relationalen Datenmodells, welche es erlaubt, die in einer Relation abgespeicherten scharfen Daten nach unscharfen Kriterien einzuteilen. Vorgestellt wurde dieser Ansatz in [Sc98], kürzere Beschreibungen finden sich auch in [MMWS03].

Als Kontext eines Attributs in einem relationalen Schema wird eine Einteilung des Wertebereichs dieses Attributs bezeichnet. Dabei kann man zwischen scharfen und unscharfen Einteilungen unterscheiden. Im scharfen Fall unterteilt ein Kontext den Wertebereich eines Attributs in Äquivalenzklassen. In der realen Welt gibt es jedoch immer wieder Objekte, die sich nicht eindeutig einer Kategorie zuordnen lassen. Dem tragen unscharfe Kontexte Rechnung, da sie Einteilungen mit kontinuierlichen Übergängen darstellen. Hierfür werden linguistische Variablen verwendet, ein Konzept zur Beschreibung unscharfer Mengen, das aus dem Bereich der Fuzzy-Logik stammt ([Zi92]). Eine linguistische Variable besitzt als Werte sprachliche Konstrukte, sogenannte Terme. Zu jedem Term T gehört eine Zugehörigkeitsfunktion $\mu_T(x)$, die für jeden Wert x aus dem Wertebereich des Attributs, zu dem die linguistische Variable gehört, angibt, wie sehr der Term auf den Wert zutrifft. Die Zugehörigkeitsfunktion kann alle Werte zwischen null (”Term trifft überhaupt nicht zu.“) und eins (”Term trifft voll zu.“) annehmen. Durch die verschiedenen Terme einer linguistischen Variable kann so der Wertebereich eines Attributs unscharf, d.h. mit kontinuierlichen Übergängen, eingeteilt werden.

Für das in Abschnitt 1.1 beschriebene Szenario könnte man beispielsweise eine linguistische Variable *Staulänge* mit den Termen *kurz*, *mittel* und *lang* definieren. Dies ist in Abbildung 1 dargestellt. Dadurch wird zum einen die Ungenauigkeit modelliert, welche in der Information über die Staulänge enthalten ist, zum anderen wird der den natürlichsprachlichen Begriffen anhaftenden Unschärfe durch kontinuierliche Übergänge Rechnung getragen.

Definiert man Kontexte für die Attribute einer Relation, so ergibt sich daraus eine Einteilung der Relation in Klassen. Im scharfen Fall ergeben sich diese Klassen als Kreuzprodukte der Äquivalenzklassen für die einzelnen Attribute, im unscharfen Fall sind die Klassen durch die Werte der linguistischen Variablen bestimmt. Diese Klassen können zur Selektion von Tupeln, die in einer bestimmten Klasse liegen, oder zur Gruppierung bei der

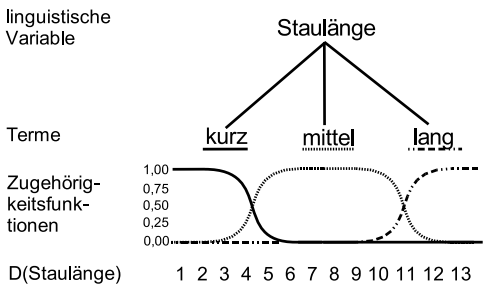

Abbildung 1: Linguistische Variable für die Staulänge

Aggregation verwendet werden.

2.1 Beziehung zum multidimensionalen Datenmodell

Zwischen dem eben beschriebenen Kontextmodell und dem multidimensionalen Datenmodell besteht eine Parallele: Die Kontexte dienen wie die Klassifikationsattribute im multidimensionalen Datenmodell (siehe [Le03]) zur Datenklassifikation.

Sind im multidimensionalen Modell für eine Dimension mehrere Klassifikationsattribute unterschiedlicher Granularität vorhanden, so ergibt sich eine Klassifikationshierarchie. Ein Beispiel hierfür wären Klassifikationsattribute für Tag, Monat und Jahr in der Zeitdimension. Eine solche Hierarchie lässt sich auch durch (scharfe) Kontexte aufbauen. Hierzu wird die Kontextbildung rekursiv angewandt. Der Kontext auf Stufe i legt fest, welche der Äquivalenzklassen des Kontexts der Stufe $i-1$ auf Stufe i als äquivalent anzusehen sind. So ergibt sich eine Klassifikationshierarchie, deren strukturierende Informationen nicht wie im multidimensionalen Modell durch die funktionalen Abhängigkeiten der Klassifikationsattribute, sondern durch die Kontexte repräsentiert werden.

Auch aus unscharfen Kontexten lässt sich eine hierarchische Einteilung aufbauen. Dies kann erreicht werden, indem man auf mehreren Stufen linguistische Variablen definiert. Die Zugehörigkeitsfunktionen einer Variable auf Stufe i geben dann an, wie sehr ihre Terme auf die Terme der Variable auf Stufe $i-1$ zutreffen. Rekursiv lässt sich so für die Terme jeder Stufe die Zugehörigkeit in Bezug auf die Elementardaten berechnen.

Neben den im Szenario für die Verkehrszustände verwendeten Termen *frei*, *lebhaft*, *stockend* und *Stau* kann man so durch eine weitere linguistische Variable *Vorankommen* mit den Termen *gut* und *schlecht*, die eine Ebene höher angesiedelt ist, eine gröbere Klassifizierung des Verkehrszustandes erreichen (siehe Abbildung 2). Durch die Terme *gut* und *schlecht* werden über die Zugehörigkeitsfunktionen die vier unscharfen Mengen, die durch die Terme auf der unteren Ebene beschrieben werden, zu zwei gröberen unscharfen Mengen zusammengefasst.

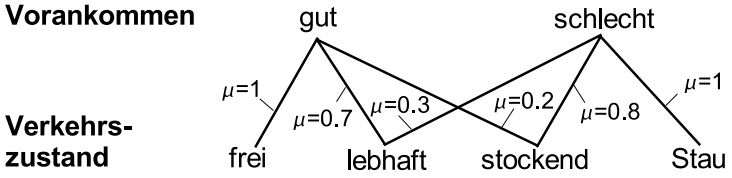

Abbildung 2: Hierarchie von Termen

2.2 Bedingungen für Aggregation über Kontexte

Eine durch Kontexte definierte Einteilung kann als Grundlage für Aggregationen verwendet werden. [LS97] führt drei notwendige Bedingungen dafür auf, dass Aggregation sinnvolle Ergebnisse liefert: *Vollständigkeit*, *Überlappungsfreiheit* und *Typverträglichkeit*. Wenn alle drei Bedingungen erfüllt sind, spricht man von *Summierbarkeit*. Die ersten beiden Bedingungen sind bei Einteilungen, die durch Anwendung scharfer Kontexte entstehen, immer erfüllt. Äquivalenzklassen sind definitionsgemäß überlappungsfrei und vollständig, d.h. jedes Element des Wertebereichs liegt in genau einer Äquivalenzklasse. Die dritte Bedingung, Typverträglichkeit, fordert, dass der Typ des zu aggregierenden Attributs mit der verwendeten Aggregatfunktion vereinbar ist und hängt nicht von der Art der Einteilung ab, die der Aggregation zugrunde liegt. Aus diesem Grund wird die Typverträglichkeit im Weiteren nicht betrachtet.

Im Falle unscharfer Einteilungen sind die Bedingungen *Vollständigkeit* und *Überlappungsfreiheit* nicht mehr erfüllt. Deswegen definieren wir einen erweiterten Vollständigkeitsbegriff: Eine Klasseneinteilung durch das Kontextmodell ist *fuzzy-vollständig*, wenn für jeden Elementarwert gilt: Die Summe der Klassenzugehörigkeiten des Werts beträgt mindestens eins. Damit bleibt gewährleistet, dass jeder Elementarwert in ausreichendem Umfang durch die Klassen erfasst wird. Ebenso lässt sich die *Überlappungsfreiheit* erweitern: Eine Klasseneinteilung durch das Kontextmodell ist *fuzzy-überlappungsfrei*, wenn für jeden Elementarwert gilt: Die Summe der Klassenzugehörigkeiten beträgt höchstens eins. Dies garantiert, dass kein Elementarwert zu stark berücksichtigt wird. Eine Klasseneinteilung ist also sowohl *fuzzy-vollständig* als auch *fuzzy-überlappungsfrei*, wenn die Summe der Zugehörigkeiten zu den Klassen genau eins beträgt. Ist gleichzeitig die *Typvertr äglichkeit* erfüllt, so liegt *Fuzzy-Summierbarkeit* vor. Diese Erweiterung ist sowohl für unscharfe als auch für scharfe Einteilungen geeignet und entspricht im Fall der scharfen Einteilung der ursprünglichen Definition von *Vollst ändigkeit* und *Überlappungsfreiheit* nach [LS97].

Die Klassifikation aus Abbildung 2 ist *fuzzy-überlappungsfrei* und *fuzzy-vollständig*, weil die Summe der Zugehörigkeiten zu den Termen der übergeordneten Dimensionsebenen immer genau eins ergibt:
$\mu_{\mathrm{gut}}(\mathrm{frei}) + \mu_{\mathrm{schlecht}}(\mathrm{frei}) = \mu_{\mathrm{gut}}(\mathrm{lebhaft}) + \mu_{\mathrm{schlecht}}(\mathrm{lebhaft}) = \mu_{\mathrm{gut}}(\mathrm{stockend}) + \mu_{\mathrm{schlecht}}(\mathrm{stockend}) = \mu_{\mathrm{gut}}(\mathrm{Stau}) + \mu_{\mathrm{schlecht}}(\mathrm{Stau}) = 1.$

Um *Fuzzy-Vollständigkeit* und *Fuzzy-Überlappungsfreiheit* zu erreichen, besteht grund-

sätzlich die Möglichkeit, die Klassenzugehörigkeit zu normieren. Hierzu dividiert man jede Klassenzugehörigkeit jedes Elements durch die Summe seiner Zugehörigkeiten zu allen Klassen ([Sc98]). Beim Entwurf einer unscharfen Klassifikation, die *Fuzzy-Summierbarkeit* erfüllt, muss natürlich zuvor die Zugehörigkeitsfunktion modelliert und parametrisiert werden.

2.3 Erweitere Aggregationsoperatoren

Um auch unscharfe Einteilungen als Grundlage für Aggregationen verwenden zu können, muss die Definition der Aggregatfunktionen, die ja nur für scharfe Einteilungen konzipiert sind, erweitert werden. Die Aggregatfunktion *Anzahl* kann für eine unscharfe Klasse berechnet werden, indem man die Zugehörigkeiten aller Tupel der Relation zu dieser Klasse summiert. Die Aggregatfunktion *Summe* kann in diesem Fall durch die mit der Zugehörigkeit gewichtete Addition der Werte des zu summierenden Attributs ermittelt werden. Die Aggregatfunktion *Durchschnitt* kann als Quotient aus *Summe* und *Anzahl* berechnet werden.

Für andere Aggregatfunktionen, wie z.B. *Minimum* und *Maximum*, gelingt eine sinnvolle Erweiterung für unscharfe Klassen jedoch nur, wenn man die Bedeutung von Minimum und Maximum anpasst. Es ist nicht mehr sinnvoll möglich, das Minimum oder Maximum aller Elemente einer unscharfen Klasse zu bestimmen. Es ist allenfalls möglich, das Minimum oder Maximum des α-Schnitts einer Klasse, also der Menge der Elemente, deren Zugehörigkeit größer als ein Wert α ist, zu bestimmen. Dies bedeutet, dass sich je nach Wert des Parameters α unterschiedliche Minima bzw. Maxima ergeben.

3 Erweiterung des CWM zur Beschreibung von Imperfektion

Die Verwendung bestimmter Konzepte zur Abbildung von imperfekten Daten, wie beispielsweise dem zuvor beschriebenen Kontextmodell, macht es erforderlich, dass alle am Data-Warehouse-Prozess beteiligten Werkzeuge die verwendeten Konzepte verstehen. Um den Einsatz verschiedener Werkzeuge zu erleichtern, schlagen wir vor, die nötigen Informationen über die zugrundeliegende Art der Imperfektion mit Metadaten zu beschreiben. Werkzeuge, die Imperfektion unterstützen, erhalten alle nötigen Informationen in Form von Metadaten. Wie sie mit diesen Informationen umgehen, bleibt ihnen selbst überlassen. Dadurch entsteht ein offenes System, das sowohl Werkzeuge mit als auch ohne Unterstützung der Imperfektion zulässt.

3.1 Common Warehouse Metamodell

Das Common Warehouse Metamodel (CWM, [PCTM03]) ist ein von der OMG propagierter Standard zur Beschreibung von Data-Warehouse-Systemen. Dieser Standard de-

finiert ein Metamodell, das Metadaten sowohl aus betriebswirtschaftlicher als auch aus technischer Sicht darstellen kann. Das CWM stellt als Metamodell sowohl die Syntax als auch die Semantik bereit, um den kompletten Data-Warehouse-Prozess beschreiben zu können. Es wird verwendet, um Metadaten zwischen heterogenen Systemen auszutauschen [PCTM03]. Es liefert zwar eine breite Basis zur Modellierung von Metadaten, ist aber in einigen Fällen nicht ausreichend, um spezielle Modelle genau zu beschreiben. Deshalb wurden im CWM zwei Typen von Erweiterungsmöglichkeiten geschaffen.

Der erste Erweiterungstyp (*Simple extensions*) bietet dem Anwender die Möglichkeit einem beliebigen Objekt eine frei wählbare Anzahl an *Stereotypes* und *TaggedValues* zuzuordnen. Einem *Stereotype* können mehrere *TaggedValues* verpflichtend zugeordnet werden. *TaggedValues* sind frei wählbare Name/Wert-Paare, die beliebigen Modellelementen zugeordnet werden können. Bei dieser Erweiterung ist zu beachten, das *TaggedValues* und *Stereotypes* und deren Semantik eigentlich keine signifikante Bedeutung für das CWM an sich haben [PCTM03]. Es wird zwar die Verbreitung, nicht aber die Semantikeinhaltung der *Stereotypes* und *TaggedValues* vom CWM gewährleistet. Werkzeuge, die solche Metadaten austauschen, müssen sich also über die Bedeutung der *Stereotypes* vorab geeinigt haben.

Beim zweiten Typ (*Modeled extensions*) kann das CWM zum Beispiel mit Vererbung erweitert werden. Dies ist ein schnell ersichtlicher, einfacher Weg eine Erweiterung zu erstellen, da das CWM selbst mit Vererbung aufgebaut ist. Dabei kann das CWM auch um ganze Packages erweitert werden. So wurde die CWM-Spezifikation bereits um das E/R-Package erweitert, mit dem E/R-Modelle abgebildet werden können.

3.2 Erweiterung des CWM

Nachdem das zuvor beschriebene Kontextmodell eine Erweiterung des relationalen Datenmodells darstellt, erweitern wir das CWM mit *Stereotypes* und *TaggedValues* so, dass man damit imperfekte Daten im relationalen Datenmodell beschreiben kann. Die Erweiterung ermöglicht, bestimmte Informationen über Imperfektion durch die ganze Architektur eines Data Warehouses hindurch bis zur obersten Analyseschicht zu erhalten. Dadurch soll eine möglichst realistische Sicht für die Analysewerkzeuge dargestellt werden.

Zunächst bietet es sich an, die Tabellen eines Schemas mit einem *Stereotype* `<<Imperfect Table>>` zu typisieren. Dieser *Stereotype* enthält einen *Constraint*, der sicherstellt, dass mindestens eine der Spalten der Tabelle Imperfektion beinhaltet. So lassen sich Tabellen, die keinerlei Imperfektion enthalten, von Tabellen mit Imperfektion unterscheiden. Diese Einteilung ermöglicht es dem Benutzer vorab, eine Zusicherung zu treffen, ob das Ergebnis seiner Anfrage möglicherweise aus imperfekten Daten erstellt wurde.

Um genau sagen zu können, welche der Spalten Imperfektion beinhalten, werden die einzelnen Spalten ebenfalls mit einem *Stereotype* versehen. Wir unterscheiden drei *Stereotypes*, die nach den verschiedenen Arten der Imperfektion benannt sind, und die entsprechend Verwendung finden: Der *Stereotype* `<<Uncertain>>` wird mit den verpflichtenden *TaggedValues* `Unsicher` und `Wahrscheinlichkeitsfunktion` versehen.

Der *Stereotype* `<<Fuzzy>>` benötigt `Unscharf` und `Zugehörigkeitsfunktion` und der *Stereotype* `<<Imprecise>>` die *TaggedValues* `Ungenau` und `Zugehörigkeitsfunktion`. Diese drei *Stereotypes* ergänzen dann die entsprechenden Spalten.

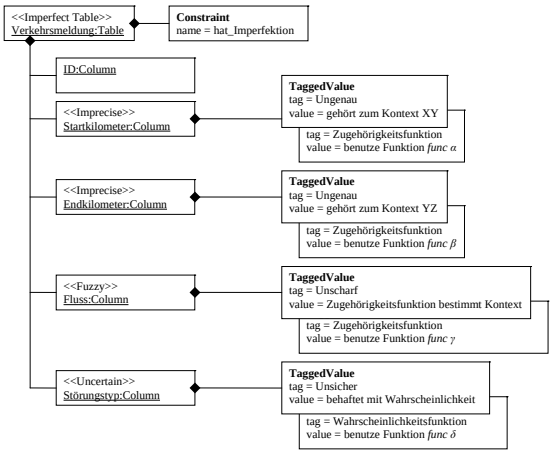

Abbildung 3: Beispiel einer Tabelle mit Imperfektion

Abbildung 3 zeigt die imperfekte Tabelle *Verkehrsmeldung* als Beispiel. Sie enthält alle drei Arten von *Stereotypes*.

Mit dem *Stereotype* `<<Imprecise>>` können auch Intervalle dargestellt werden. Da scharfe Kontexte sich auch als Intervalle beschreiben lassen, können sie mit dem *Stereotype* `<<Imprecise>>` modelliert werden. Die Zugehörigkeitsfunktion liefert dann aber nur die Werte null oder eins. Unscharfe Kontexte lassen sich mit dem *Stereotype* `<<Fuzzy>>` beschreiben. Die Zugehörigkeitsfunktion kann dann auch Werte zwischen null und eins liefern.

Mit dieser Erweiterung kann man auch die aus einer Aggregationsoperation entstehenden Tabellen und Spalten stereotypisieren, falls diese imperfekte Daten repräsentieren. Somit wird sichergestellt, dass für alle späteren Operationen auf den Daten, die Art der Imperfektion klar ersichtlich ist.

Es handelt sich in der obigen Ausführung um eine erste Erweiterung. Momentan können Zugehörigkeitsfunktionen nur textuell beschrieben werden. Für die automatische Weiterverarbeitung, die es beispielsweise erlaubt, mit den Zugehörigkeitsfunktionen zu rechnen, müssen imperfekte Datentypen und Zugehörigkeitsfunktionen explizit als Klassen modelliert werden. Dazu werden wir auf den mächtigeren Erweiterungstyp *Modeled extensions* zurückgreifen.

4 Zusammenfassung und Ausblick

In diesem Papier wurde anhand eines Szenarios aus dem Verkehrsbereich beschrieben, warum es sinnvoll sein kann, Imperfektion in einem Data Warehouse zu erhalten. Das Konzept der unscharfen Kontexte wurde zur Beschreibung von Dimensionen mit unscharfer Klassifikation verwendet. Es wurde gezeigt, dass trotzdem Aggregationsanfragen sinnvoll beantwortet werden können, sofern *Fuzzy-Summierbarkeit* vorliegt und die Aggregationsoperatoren angepasst werden. Dabei werden auch Werte mit geringer Zugehörigkeit berücksichtigt und es sind auch Zugehörigkeiten zwischen null und eins erlaubt. Im zweiten Teil wurde eine erste Erweiterung beschrieben, mit der die Art der Imperfektion mit Hilfe des Common Warehouse Metamodels beschrieben werden kann.

Um ein Data-Warehouse-System zu erhalten, das auf allen Ebenen sinnvoll mit Imperfektion umgehen kann, wollen wir auch andere Arten der Imperfektion in einem Data Warehouse verwenden, die Abbildung der Imperfektion in den übergeordneten Analyseapplikationen, wie z.B. OLAP, untersuchen und eine ausreichend mächtige Erweiterung eines Metamodells entwerfen, mit Hilfe derer am Data-Warehouse-Prozess beteiligte Werkzeuge verschiedene Arten der Imperfektion automatisiert verarbeiten können.

Literatur

[DPJ03] Dyreson, C. E., Pedersen, T. B., und Jensen, C. S.: Incomplete Information in Multidimensional Databases. In: Rafanelli, M. (Hrsg.), *Multidimensional Databases: Problems and Solutions*. S. 282–309. Idea Group. 2003.

[Le03] Lehner, W.: *Datenbanktechnologie für Data-Warehouse-Systeme*. dpunkt. Heidelberg. 2003.

[LS97] Lenz, H.-J. und Shoshani, A.: Summarizability in OLAP and Statistical Data Bases. In: *Statistical and Scientific Database Management*. S. 132–143. 1997.

[MMWS03] Meier, A., Mezger, C., Werro, N., und Schindler, G.: Zur unscharfen Klassifikation von Datenbanken mit fCQL. In: *Proceedings of the GI-Workshop LLWA Lernen, Lernen, Wissen, Adaptivität*. S. 151–158. Karlsruhe, Germany. 2003.

[PCTM03] Poole, J., Chang, D., Tolbert, D., und Mellor, D.: *Common Warehouse Metamodel — Developer's Guide*. John Wiley & Sons. Indianapolis. 2003.

[PJD99] Pedersen, T. B., Jensen, C. S., und Dyreson, C. E.: Supporting imprecision in multidimensional databases using granularities. In: *Statistical and Scientific Database Management*. S. 90–101. 1999.

[RB92] Rundensteiner, E. A. und Bic, L.: Evaluating Aggregates in Possibilistic Relational Databases. *Data & Knowledge Engineering*. 7(3):239–267. 1992.

[Sc98] Schindler, G.: *Fuzzy-Datenanalyse durch kontextbasierte Datenbankanfragen*. PhD thesis. Fakultät für Wirtschaftswissenschaften, RWTH Aachen. 1998.

[Zi92] Zimmermann, H. J.: *Fuzzy Set Theory – and Its Applications, Sec., rev. edition*. Kluwer. 1992.