Qualitätsanalyse im Data Warehousing*

Manfred A. Jeusfeld
Universität Tilburg, Infolab, Postbus 90153, NL-5000 LE Tilburg
Tel: +31-13-466-3119, Email: jeusfeld@kub.nl

Matthias Jarke, Christoph Quix
RWTH Aachen, Informatik V, Ahornstr. 55, D-52056 Aachen
Tel: +49-241-80-21500, Email: {jarke|quix}@informatik.rwth-aachen.de

Zusammenfassung

Der Idee des Data Warehousing liegt zugrunde, daß transaktionsorientierte System (OLTP) und Analysesysteme (OLAP) zwar auf den gleichen Unternehmensdaten operieren, sie jedoch fundamental verschiedene Anforderungen an die Zugreifbarkeit dieser Daten haben. Transaktionen im OLTP-Bereich ändern nur wenig Daten und müssen mit hohem Durchsatz ausgeführt werden. OLAP-Analysen hingegen greifen auf hochaggregierte Daten aus einer Vielzahl von Bereichen zu, um die Unternehmung als Ganzes zu verstehen und zu steuern. Bisher hat sich die Softwareindustrie und auch die Forschung auf die technische Lösung mittels materialisierter Sichten, den Data Cubes konzentriert. In diesem Artikel berichten wir über erste Ergebnisse aus einem Forschungsprojekt, das sich mit dem Qualitätsmanagement in Data Warehouses beschäftigt. Ergebnis ist eine Variante des Goal-Question-Metric-Ansatzes, die über die Metadatenbank des Data Warehouses operationalisiert wird. Fernziel ist eine Entwurfsmethode, in der ein Data Warehouse so entworfen wird, daß die Qualitätsanforderungen der Beteiligten berücksichtigt werden. Dazu muß zunächst ein Rahmen geschaffen werden, in der Qualitätsanforderungen formulierbar und ihr momentaner Erfüllungsgrad analysiert werden kann.

1. Die Gegenstände der Qualitätsanalyse

Mit Hilfe eines Data Warehouse werden wichtige Entscheidungen der Unternehmung getroffen. Mangelhafte Qualität im Data Warehouse verursacht durch Fehlentscheidungen hohe Kosten für die Unternehmung. Die große Zahl von Unternehmensberatern, die sich auf Data Warehousing spezialisiert haben, zeigt das mangelhafte formale Verständnis über das Qualitätsmanagement in diesem Bereich. Selbst wenn Qualitätseffekte im Prinzip verstanden sind, so macht die schiere Anzahl an Komponenten im Data Warehouse eine manuelle Qualitätskontrolle praktisch unmöglich. Hinzu kommt, daß unterschiedliche Beteiligte (Stakeholders) unterschiedliche Qualitätspreferenzen haben. Der Administrator einer Datenquelle ist an einer hohen Verfügbarkeit seines Systems interessiert, während ein Entscheidungsträger möglichst aktuelle Informationen verlangen mag. Das Projekt DWQ (Foundations of Data Warehouse Quality) [3,6] will hier Abhilfe schaffen. Ein Baustein der Lösung ist ein Qualitätsmodell, in dem Qualitätsziele formuliert werden, ihre Kontrolle über

---

* Das Original dieses Artikels erscheint im Heft 1/99 (Schwerpunktthema “Data Warehousing”) der Fachzeitschrift INFORMATIK/INFORMATIQUE der Schweizer Gesellschaft für Informatik. Die vorliegende Fassung enthält einige zusätzliche Abbildungen, die im Orginalartikel aus Platzgründen fehlen.

Messungen geplant wird und mathematische Modelle über Abhängigkeiten zwischen Faktoren verwaltet werden. In diesem Artikel konzentrieren wir uns auf die Qualitätsanalyse mit Hilfe dieses Metamodells.

Welche Entitäten eines Data Warehouse sollen überhaupt dem Qualitätsmanagement unterliegen? Die übliche Sicht auf ein Data Warehouse ist vorwiegend physisch: ein Data Warehouse besteht aus Analysewerkzeugen, einer Anzahl von Datenspeichern (Datenbanken), Transportagenten, Kontrollagenten, und einer Metadatenbank, die zur Administration genutzt wird. Die Terminologie verrät eine rein technische Sicht, die weit entfernt von der Sprache ist, in der Entscheidungsträger ihre Anforderungen formulieren. Sie interessieren sich eher für Aussagen über die Gegenstände der Unternehmung als für technische Komponenten eines Systems, das letztlich nur Daten zwischen Datenspeichern transportiert. Die physische Sicht vernachlässigt Struktur (logische Sicht) und Bedeutung (konzeptuelle Sicht) der Komponenten. Ziel muß also sein, diese drei Perspektiven eines Data Warehouses in Beziehung zueinander zu setzen und darauf ein Qualitätsmodell aufzubauen.

Die *konzeptuelle Perspektive* beschreibt die Entitäten, die im Data Warehouse behandelt werden unabhängig von der logischen Repräsentation und ihrem physischen Speicherort. Wir unterscheiden Konzepte auf der operationellen Ebene, der Unternehmensebene, und der Anwenderebene. Auf der operationellen Ebene finden wir die konzeptuellen Modelle der Datenquellen. Diese können etwa als ER-Diagramme dargestellt sein. Auf der Unternehmensebene sprechen wir von Unternehmensmodellen. Die Anwenderebene schließlich beschreibt jene Konzepte, die für den Anwender relevant sind. Die Konzepte der drei Ebenen stehen zueinander in Beziehung. Die typische (und einfachste) Beziehung ist die Teilmengenbeziehung. Man nehme etwa an, daß im Unternehmensmodell ein Konzept ‘Kunde’ repräsentiert ist. Ferner nehme man an, daß es zwei konzeptuelle Modelle von Datenquellen gibt, die die Konzepte ‘US-Kunde’ und ‘Vorzugskunde’ enthalten. Beide Konzepte beschreiben Teilmengen des Konzeptes ‘Kunde’ (siehe [2] für eine ausführliche formale Diskussion möglicher Beziehungen zwischen Konzepten verschiedener Ebenen im Data Warehousing).

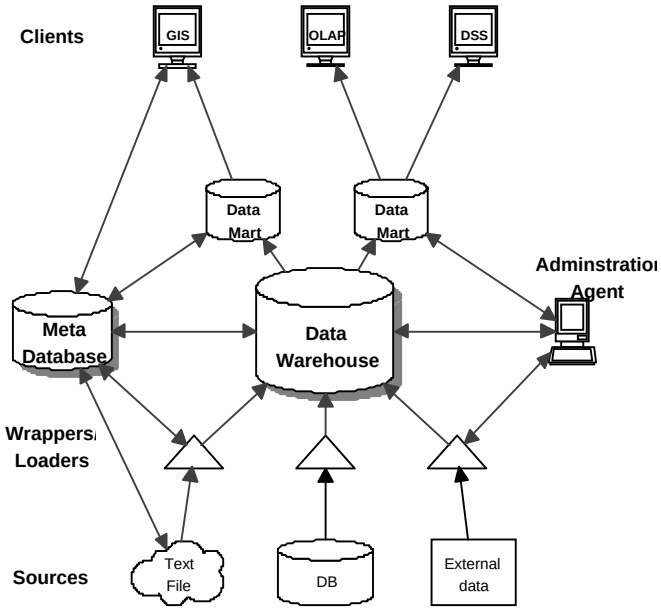

Abbildung 1: Eine übliche Darstellung von Data Warehouses

Die *logische Perspektive* enthält Datenstrukturen für die Repräsentation von Instanzen der Konzepte. Dies sind die Datenstrukturen der Datenquellen, des Data Warehouses und der Applikationsprogramme. Diese Datenstrukturen sind per Forward-Engineering aus dem entsprechendem konzeptuellen Modell entstanden oder das konzeptuelle Modell ist durch Reverse-Engineering aus dem logischen Schema gewonnen worden. Während konzeptuelle Modelle Mengen von Entitäten und ihre Eigenschaften beschreiben, gibt ein logisches Schema an, welche Datenfelder vorgesehen sind, um Objekte und ihre Attribute auf dem Rechner darzustellen. Die *physische Perspektive* schließlich repräsentiert die Orte der Datenspeicher (Bezeichner einer Datenbank, Dateiname in einem Dateisystems eines Rechners) und der Prozesse (Prozeßidentifikatoren von Wrappern, Anwenderprogrammen etc.). Die Prozesse und Datenspeicher genügen den Spezifikationen der logischen Perspektive. Ein Transportprozeß liest Daten von einer Quelle in deren Datenstruktur und speichert sie in der Datenstruktur des Zielspeichers in dessen Datenformat.

Ein umfassendes Qualitätsmanagement muß in der Lage sein, auf Objekte aller drei Perspektiven und Ebenen Bezug zu nehmen. Als Beispiel nehme man an, daß ein Entscheidungsträger an dem Konzept ‘Person’ interessiert ist. Zwar mag das Unternehmensmodell ein solches Konzept bereitstellen, jedoch zeigt ein Vergleich mit der logischen Perspektive, daß es keine Datenstruktur für dieses Konzept gibt. In der Dimension ‘Vollständigkeit’ hat das logische Schema des Data Warehouses also eine nicht ausreichende Qualität. Ein Beurteilung der Vollständigkeit erfordert einen Vergleichsmaßstab. Eben dies leistet der Bezug zwischen den Perspektiven. Im folgenden nehmen wir an, daß alle qualitätsrelevanten Objekte eines Data Warehouse-Systems abstrakt in der Metadatenbank repräsentiert sind. Wir verwenden die Kategorie ‘MeasurableObject’ zum Bezug auf solche Objekte.

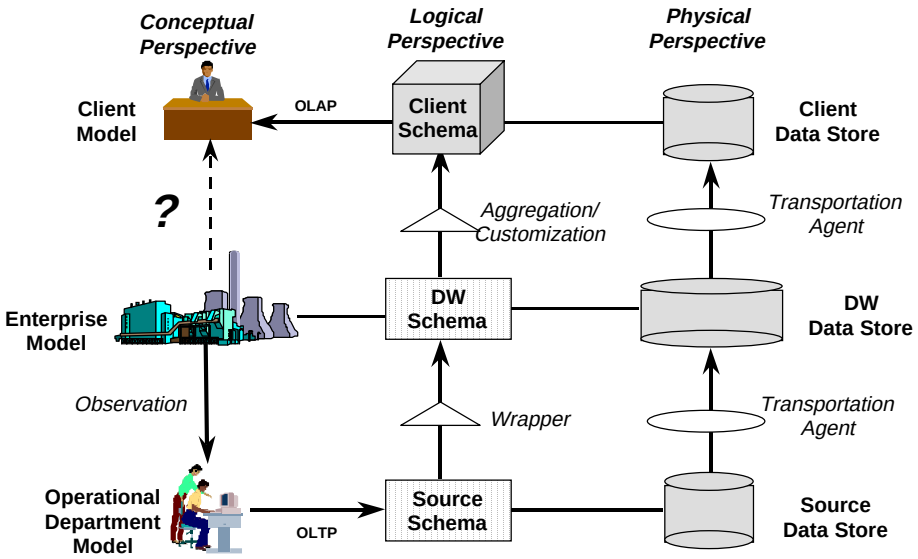

Abbildung 2: Drei Perspektiven in Data Warehouses

Abbildung 2 unterteilt die dem Qualitätsmanagement unterliegenden Objekte in neun Kategorien. Dies wird zur Grundlage des Schemas der Metadatenbank gemacht: die Metadatenbank soll all diese Objekte sowie ihre Querbezüge abstrakt repräsentieren. Abbildung 3 zeigt dieses Schema als Hierarchie. Man beachte, daß mit den Konzepten wie ‘EntityType’ und ‘Relation’ komplette Notationen (Datenmodelle) definiert werden.

Tatsächlich sind konzeptuelle, logische und physische Modelle in der Metadatenbank abzuspeichern. Das Objekt ‘MeasurementType’ schafft die Beziehung zum Qualitätsmodell. Abbildung 3 sagt nicht nur, daß Objekte aus allen neun Kategorien im Prinzip der Qualitätsmessung unterliegen, sondern daß dies in der Metadatenbank des Data Warehouse explizit gemacht wird.

2. Ein Qualitätsmodell für Data Warehouses

Wenn man über Qualität eines Produktes (oder Prozesses) spricht, so bezieht man sich auf Attribute des Produktes. Um das Qualitätsmanagement zu objektivieren, werden solchen Attribute Qualitätswerte zugeordnet. Diese Zuordnung heißt Qualitätsmessung [4]. Da Qualität an Anforderungen von beteiligten Personen gebunden ist, sollte ein Qualitätsmodell verschiedenen Beteiligten erlauben, verschiedene Qualitätsziele zu formulieren und den Grad ihrer Erfüllung zu evaluieren. Ein bewährte Methode aus dem Software-Engineering ist der GQM-Ansatz [1]: Qualitätsziele (Goals) sind Aussagen der Art “Verbessere die Vollständigkeit des Data Warehouses”. Basili gibt hierzu einige allgemeine Satzmuster an. Im nächsten Schritt werden die Qualitätsziele auf Qualitätsfragen (Questions) abgebildet. Eine mögliche Qualitätsfrage ist: “Wie hoch ist der Anteil der Konzpete im Unternehmensmodell, die keine vollständige Entsprechung im Data Warehouse-Schema haben?” Ein zweite mögliche Qualitätsfrage für das obige Qualitätsziel ist: “Wie vollständig sind die materialisierten Datentupel im Data Warehouse”? Zur Beantwortung der Qualitätsfragen zieht man Messungen (Metrics) heran. Qualitätsmessungen benutzen eine vorgeschriebene Methode, um einem Objekt einen Meßwert (Qualitätswert) zuzuordnen. Für die erste Qualitätsfrage kann man etwa die Zahl der Konzepte des Unternehmensmodells bestimmen, für die es keine Datenstruktur gibt, und diese Zahl dann durch die Gesamtzahl aller Konzepte des Unternehmensmodells dividieren. Für die zweite Qualitätsfrage ist eine mögliche Metrik die Anzahl der Tupel mit Null-Werten im Data Warehouse.

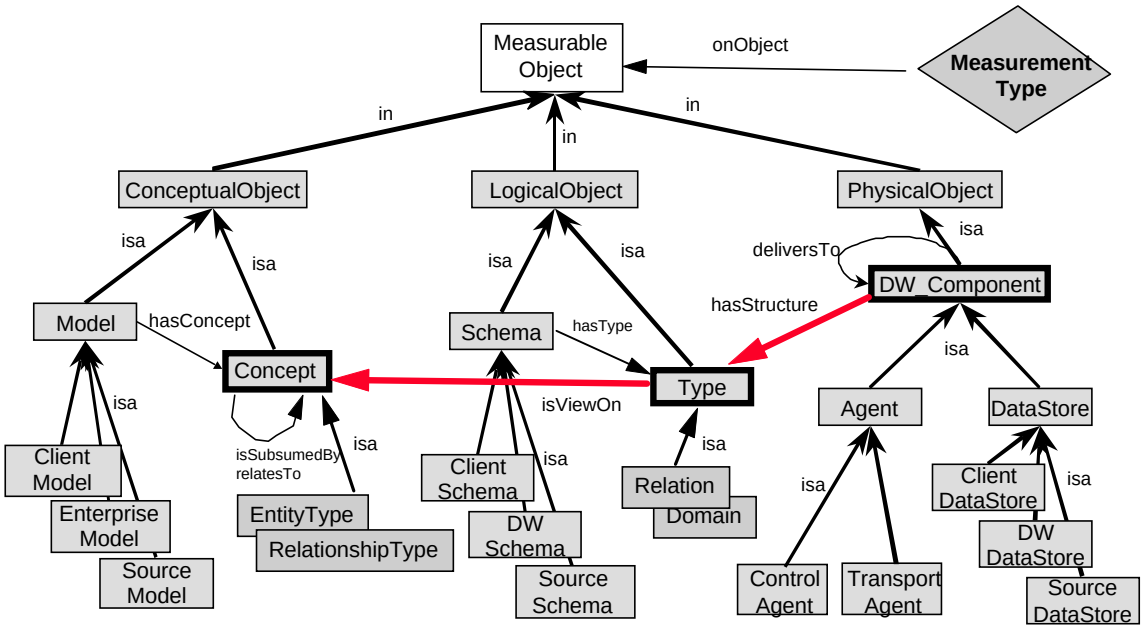

Abbildung 3: Metamodell der Objekte eines Data Warehouse-Systems

Ursprünglich ist der GQM-Ansatz als Hilfestellung für die Auswahl von Metriken entwickelt worden. Wir schlagen eine Variante vor, die auf einer formalen Repräsentation der Aussagen beruht und dadurch das automatisierte Qualitätsdatenmanagement und die Qualitätsanalyse durch Anfragen erlaubt. Ein konventionelles Data Warehouse verfügt bereits über eine Metadatenbank, mit Hilfe derer die Komponenten des Systems verwaltet werden. Genau diese Objekte sollen aber Gegenstand des Qualitätsmanagements sein! Es liegt also nahe, die Metadatenbankkomponente um die Konzepte des GQM-Ansatzes anzureichern. Abbildung 4 zeigt dieses Qualitätsmodell. Der obere Teil beschreibt das Muster zur Formulierung der Qualitätsziele. In der Mitte stehen die Qualitätsanfragen, die nun als Anfragen an die Metadatenbank interpretiert werden. Der untere Teil beschreibt, wie Qualitätsmessungen repräsentiert werden. Man beachte, daß sowohl Qualitätsziele als auch Messungen sich auf ‘meßbare Objekte’ beziehen.

Die Objekte des Metamodells sind Metaklassen, d.h. die stellen eine Notation bereit, in der das Qualitätsmanagement für ein Data Warehouse beschrieben wird. Formal ist das Qualiätsmodell Teil des Schemas der Metadatenbank des Data Warehouses. Wie bereits gesehen, bezieht sich das Qualitätsmodell auf Objekte aus unterschiedlichen Perspektiven. Die Objekte des Metamodells sind Metaklassen, d.h. die stellen eine Notation bereit, in der das Qualitätsmanagement für ein Data Warehouse beschrieben wird. Formal ist das Qualiätsmodell Teil des Schemas der Metadatenbank des Data Warehouses. Wie bereits gesehen, bezieht sich das Qualitätsmodell auf Objekte aus unterschiedlichen Perspektiven. Bisher haben wir nicht berücksichtigt, daß Objekte auch unterschiedliches Abstraktionsniveau haben. Es stellt sich heraus, daß das Qualitätsmodell in zwei Schritten instanziiert werden kann. Im ersten Schritt werden Muster für Qualitätsziele und -messungen festgelegt. Im zweiten Schritt werden diese Muster zur Formulierung konkreter Qualitätsziele und zur Darstellung konkreter Messungen herangezogen.

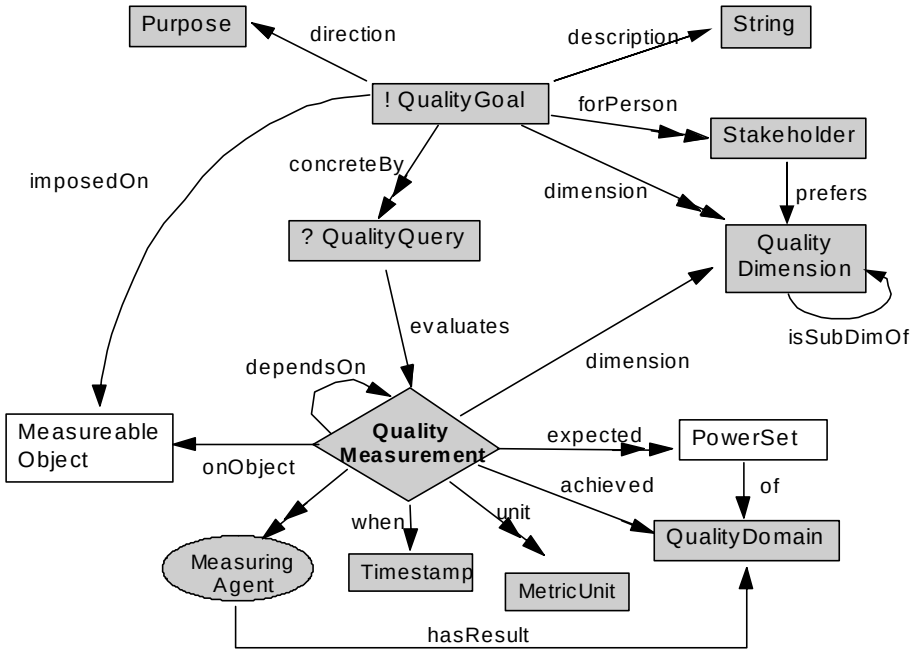

Abbildung 4: Ein Qualitätsmodell für Data Warehouses

Für Datentyp ‘Relation’ kann man zum Beispiel eine Messung definieren, die die Zahl der Nullwerte bestimmt. Dies ist der *Typ* einer Qualitätsmessung. Für ein Beispiel einer Relation, also etwa ‘Kunde’, wird über eine *Instanz* dieses Meßtyps zu einem konkreten Meßzeitpunkt eine konkrete Zahl ermittelt. Das Qualitätsmodell wird also zweifach instanziiiert. Dieses Verfahren hat den Vorteil, daß das Qualitätsmodell direkt zum Qualitätsmanagement genutzt wird. Dies sei anhand eines Beispiels erläutert.

Abbildung 5 zeigt in im oberen des Ausschnitt des Qualitätsmodells zur Formulierung von Qualitätszielen. Im mittleren Drittel wird formuliert, daß Data Warehouse-Administratoren an der Erreichung eines Grades an Verfügbarkeit von (Quell-)Relationen interessiert sein können. Diese Aussage ist der *Typ eines Qualitätsziels* für Data Warehouse-Administratoren. In unserem Ansatz werden solche Muster in der Metadatenbank des Data Warehouse abgespeichert. Zu jedem Zeitpunkt können neue solche Muster definiert werden. Die Menge aller solcher Muster stellt das Wissen dar, wie man im Prinzip Qualitätsziele im Data Warehousing beschreiben kann. Die Neuigkeit an diesem Ansatz ist, daß dieses Wissen Teil der Metadatenbank des Data Warehouses ist. Eine Unternehmung, die ein Data Warehouse betreibt, kann auf diese Weise ihre Politik und Erfahrung beim Qualitätsmanagement explizit verfügbar machen. Das untere Drittel in Abbildung 5 zeigt die konkrete Instanziierung eines Qualitätszieles. Der Administrator Mokrane sagt als sein erstes Ziel aus, daß er an der Verfügbarkeit der Relation ‘SalesEurope’ interessiert ist. Die Relation ‘SalesEurope’ ist als Klasse gekennzeichnet, da sie Instanzen haben kann. Die Objekte ‘Mokrane’ und ‘Goal 1’ sind hingegen faktischer Natur. Dieser Wechsel der Abstraktionsebenen ist typisch für Qualitätsmessungen.

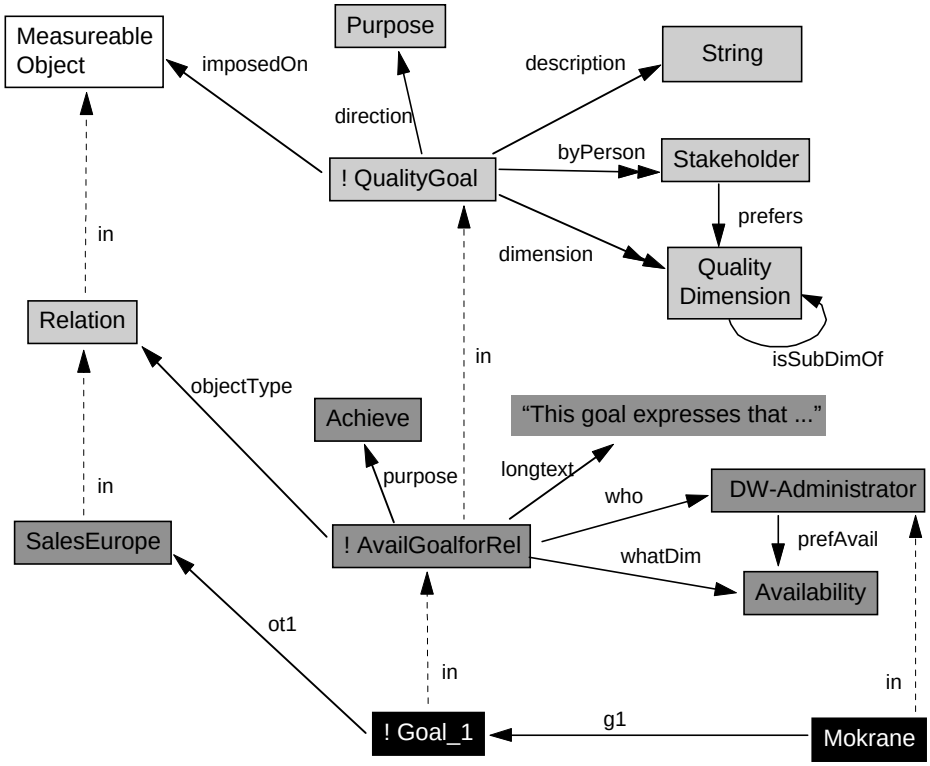

Abbildung 5: Formulierung von Qualitätszielen

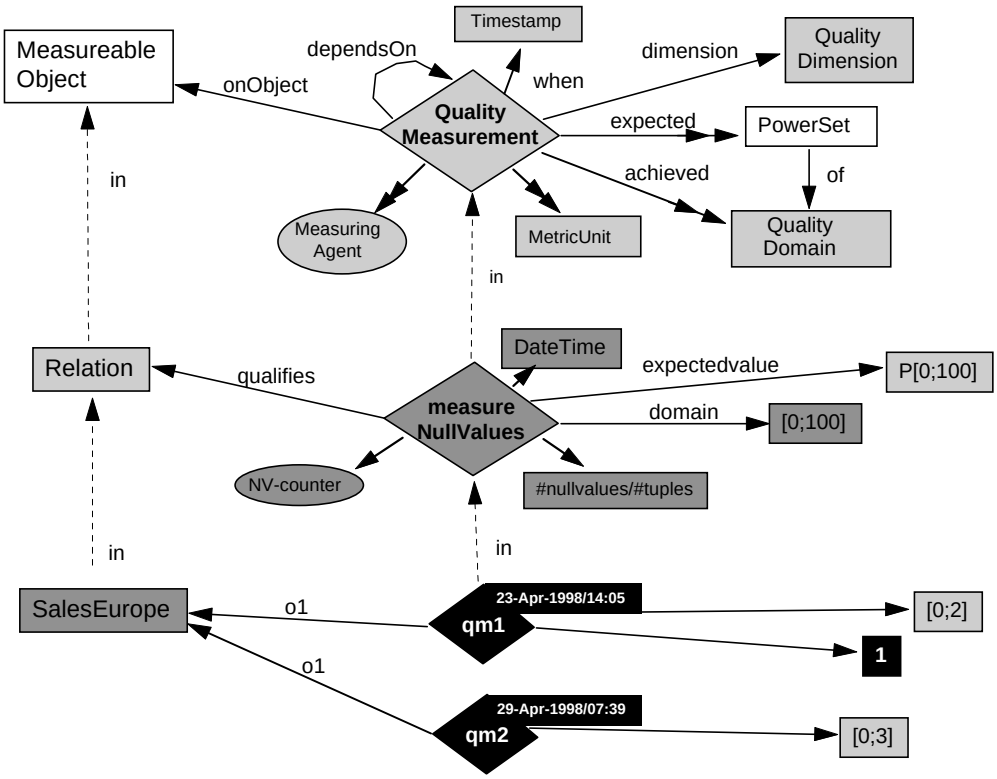

Abbildung 6: Darstellung von Qualitätsmessungen

Auch bei Qualiätsmessungen (vgl. Abbildung 6) finden wir die zweistufige Instanziierung. In der mittleren Ebene ist ein Typ einer Qualitätsmessung repräsentiert, hier die Messung von Nullwerten. Die Messung führt in das Interval [0;100]. Als Meßagent ist eine externe Methode ‘nv-counter’ angegeben. Zudem wird spezifiert, daß ein Teilmenge des Intervals als erwarteter Meßwert angegeben werden kann. Genauso wie es im Data Warehouse Transport- und Kontrollagenten gibt, die den Datentransfer zwischen Quellen und Zielen leisten, gibt es nun Meßagenten, die Meßwerte liefern. Die untere Ebene zeigt das Resultat einer solchen Ausführung: die Messung ‘qm1 ergab des Qualitätswert 1 und liegt somit im erwarteten Interval für diese Messung. Das Objekt ‘qm2’ ist eine partielle Instanziierung des Meßtyps. Diese wird als Plan interpretiert, diese Messung zu dem vorgegebenen Zeitpunkt auszuführen und somit einen Messwert zu bestimmen. Da alle Objekte in der Metadatenbank gehalten werden, kann diese den Meßagenten gemäß dem Plan aufrufen. Der Meßagent wiederum trägt den Meßwert in die Metadatenbank ein (vgl. Abbildung 7).

Bisher wurde beschrieben, wie das Qualitätsmetamodell sowohl zur Formulierung von Qualitätszielen und -messungen genutzt werden kann. Es fehlt noch die Brücke zwischen den eher vagen Qualitätszielen und den sehr konkreten Qualitätsmessungen. Hierzu werden Anfragen genutzt. Während in GQM nur von *Qualitätsfragen* gesprochen wird, deren Beantwortbarkeit und Antwort von der Interpretation der Frage durch den Menschen abhängt, nutzen wir die Tatsache, daß alle qualitätsrelevanten Daten in der Metadatenbank gespeichert sind. Wir verwenden hier die Anfragesprache von ConceptBase [5,7], um *Qualitätsanfragen* zu repräsentieren. Im einfachsten Fall wird durch eine Qualitätsanfrage bestimmt, ob eine Qualitätsmessung im erwarteten Bereich liegt. Die Anfrage ‘TooManyNullValues’ leistet dies für Quell-Relationen: eine solche Relation hat zu viele Nullwerte, wenn es eine Messung m der Nullwerte gibt, die nicht im erwarteten Interval liegt.

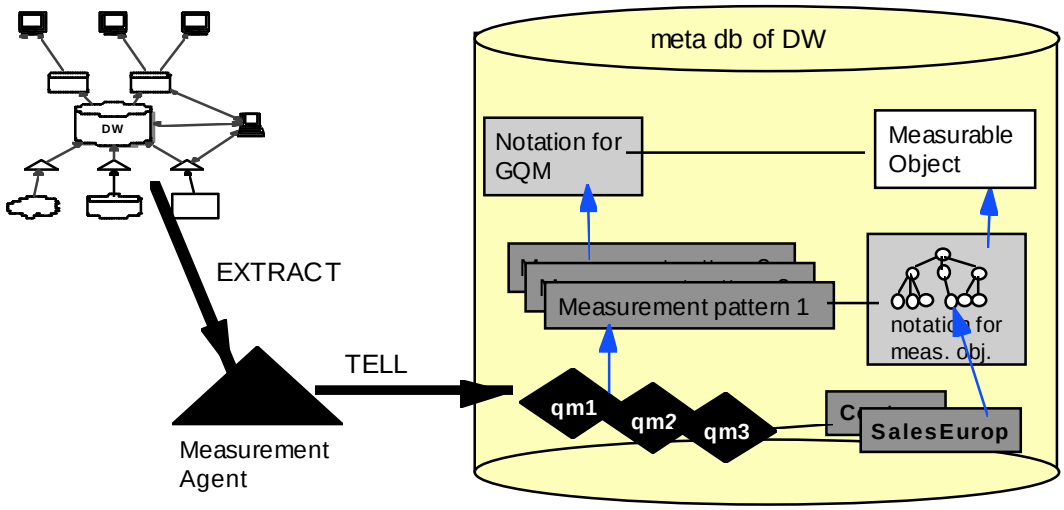

Abbildung 7: Interaktion von Meßagenten mit der Metadatenbank

```
QualityQuery TooManyNullValues isA Source,Relation with
  constraint
     c: $ exists m/MeasureNullValues (this hasMeasure m) and
           not (m in MeasureNullValues^exprange) $
end
```

Als zweites Beispiel betrachte man die Anfrage ‘BetterOnNullValues’. Sie gibt als Antwort solche Relationen, deren Meßwert für die Anzahl der Nullwerte abnimmt. Das Prädikat ‘after’ bezieht sich dabei auf den Zeitstempel der Messungen.

```
QualityQuery BetterOnNullValues isA Source,Relation
with constraint
     c: $ exists m1,m2/MeasureNullValues t1,t2/DateTime
            (this hasMeasure m1)  and (m1 when t1) and
            (this hasMeasure m2) and (m2 when t2) and (t2 after t1) and
            exists v1,v2/[0;100] (m1 qualityvalue v1) and
            (m2 qualityvalue v2) ==> (v1 > v2) $
end
```

Da Qualitätsfragen nun Anfragen an die Metadatenbank sind, können Qualitätsberichte automatisch erstellt werden: sie enthalten die Antworten auf die Qualitätsanfragen. Regelmäßige Berichte über den Inhalt eines Data Warehouses sind in der Praxis üblich. Man denke etwa an monatliche Berichte über die Verkaufszahlen von Niederlassungen. Durch den vorgestellten Ansatz werden die Qualitätsdaten in dieses Berichtswesen einbezogen. In der Tat sind Qualitätsdaten hochaggregierte Sichten auf das Data Warehouse.

Abbildung 8 zeigt einen Ausschnitt des graphischen Browsers von ConceptBase. Zu sehen sind einige Objekte der physischen Perspektive eines Data Warehouses. Die langegezogenen Ovale repräsentieren Qualitätsmessungen, hier bezogen auf die Aktualität von Datenspeichern. Die Messung für den Datenspeicher auf der Anwenderseite ist außerhalb des erwarteten Intervals und wird vom graphischen Browser schwarz angezeigt. Die Antwort auf die Qualitätsfrage “Welche Datenspeicher haben nicht die geforderte Aktualität”? wird also durch die Darstellung der Knoten visualisiert. Ein solches Werkzeug ist für die Überwachung der Qualität sinnvoll. Ein Administrator (oder auch Benutzer) kann die interessierenden meßbaren Objekte auf dem Bildschirm auswählen. ConceptBase bestimmt dann die

Qualitätswerte und aktualisiert die Darstellung der Meßwerte auf dem Bildschirm, wann immer sich eine Änderung ergibt.

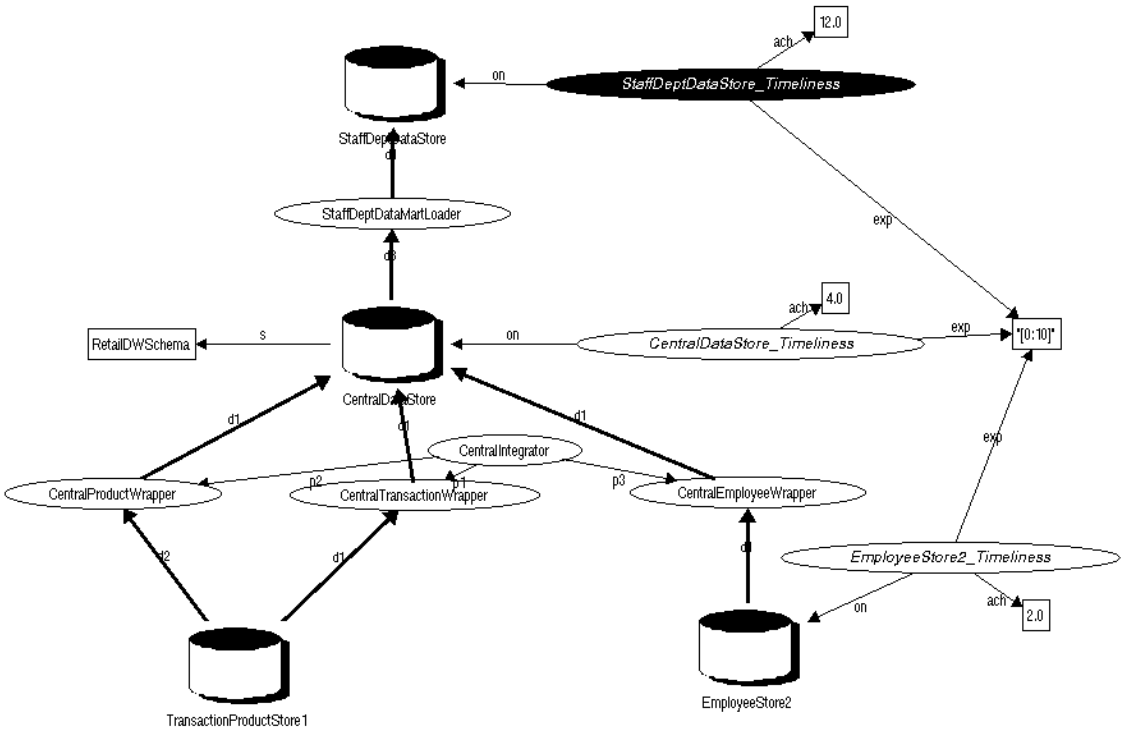

Abbildung 8: Visualisierung der Qualitätsanalyse in ConceptBase

3. Zusammenfassung und Ausblick

Der vorgestellte Ansatz wendet eine bekannte Methode zur Qualitätsplanung auf das Data Warehousing an. Durch Erweiterung der Metadatenbank wird die ursprünglich manuelle Methode zu dem Schlüssel zum teilautomatisierten Qualitätsmanagement. Die Realisierung des Ansatzes profitiert von der Metamodellierfähigkeit von ConceptBase, um Objekte aller Perspektiven und Ebenen eines Data Warehouse zu repräsentieren. Wir sehen folgende Vorteile:

- Die Instanziierung des Qualitätsmodells in zwei Schritten erlaubt die Unterscheidung von Typen von Qualitätszielen bzw. -messungen und tatsächlichen Zielen bzw. Messungen. Dadurch wird pragmatisches Wissen über Qualitätsmanagement explizit.
- Anfragen an die Metadatenbank können zur Analyse der Qualität genutzt werden. Wesentlich ist hier, daß alle qualitätsrelevanten Informationen in der Metadatenbank bereitstehen.

Es bleiben wichtige Aspekte des Qualiätsmanagements offen. Eine Frage etwa ist, wie ein Data Warehouse so entworfen werden kann, daß es vorgebene Qualitätsziele erfüllt. Eine weitere Frage ist die mathematische Modellbildung. Qualitätswerte stehen in Abhängigkeit zueinander. So ist die Aktualität eines Datenelements auf der Anwenderseite eine Konsequenz von (meßbaren) Eigenschaften der Datenquellen und der Transportagenten. Es wäre sinnvoll,

die mathematischen Modelle zur Vorhersage der abhängigen Qualitätswerte zur Laufzeit des Data Warehouses verfügbar zu haben.

Danksagung: Diese Arbeit wurde teilweise durch die Kommission der Europäischen Union als ESPRIT Long Term Research Project 22469 DWQ (Foundations of Data Warehouse Quality, http://www.dbnet.ece.ntua.gr/~dwq/) gefördert. Die Autoren danken dem Projektteam DWQ für die intensive Diskussion der hier präsentierten Ergebnisse. Besonderer Dank gilt Panos Vassiliadis, Maurizio Lenzerini, Mokrane Bouzeghoub und Enrico Franconi.

4. Literatur

[1] V.R. Basili, D.M. Weiss. A method for collecting valid software engineering data. *IEEE Trans. Software Engineering*, **10**(6):728-738 (1984).

[2] D. Calvanese, G. De Giacomo, M. Lenzerini, D. Nardi, R. Rosati. Information integration: conceptual modeling and reasoning support. In *Proc. 3rd Intl. Conf. on Cooperative Information Systems (CoopIS'98)*, New York, August 20-22, 1998, pp. 280-289.

[3] DWQ Consortium. *Deliverable D1.1, Data Warehouse Quality Requirements and Framework*. Technical Report DWQ–NTUA-1001, NTUA Athens, Greece (1997).

[4] Fenton, S. L. Pfleeger. *Software Metrics - A Rigorous & Practical Approach*. Second Edition, PWS Publ., Boston, MA. (1998).

[5] M. Jarke, R. Gallersdörfer, M.A. Jeusfeld, M. Staudt, S. Eherer. ConceptBase - a deductive objectbase for meta data management. *Journal of Intelligent Information Systems*, **4**(2):167-192 (1995).

[6] M. Jarke, Y. Vassiliou. Foundations of data warehouse quality -- a review of the DWQ project. In *Proc. 2nd Intl. Conf. Information Quality (IQ-97)*, Cambridge, Mass. (1997).

[7] M.A. Jeusfeld, M. Jarke, H.W. Nissen, M. Staudt. ConceptBase - Managing conceptual models about information systems. In P. Bernus, K. Mertins, and G. Schmidt: *Handbook on Architectures of Information Systems*, pp. 265-285, Springer-Verlag (1998).