February 2007

Kollaboratives Data Warehousing - Konzeption und prototypische Realisierung flexibler Schema- und Datenintegration

Thomas Matheis
Institut für Wirtschaftsinformatik (IWi) im Deutschen Forschungszentrum für Künstliche Intelligenz (DFKI), thomas.matheis@iwi.dfki.de

Dirk Werth
Institut für Wirtschaftsinformatik (IWi) im Deutschen Forschungszentrum für Künstliche Intelligenz (DFKI), Dirk.Werth@iwi.dfki.de

Peter Loos
Institut für Wirtschaftsinformatik (IWi) im Deutschen Forschungszentrum für Künstliche Intelligenz (DFKI), loos@iwi.uni-sb.de

In: Oberweis, Andreas, u.a. (Hg.) 2007. *eOrganisation: Service-, Prozess-, Market-Engineering*; 8. Internationale Tagung Wirtschaftsinformatik 2007. Karlsruhe: Universitätsverlag Karlsruhe

ISBN: 978-3-86644-094-4 (Band 1)
ISBN: 978-3-86644-095-1 (Band 2)
ISBN: 978-3-86644-093-7 (set)

Kollaboratives Data Warehousing

Konzeption und prototypische Realisierung flexibler Schema- und Datenintegration

Thomas Matheis, Dirk Werth, Peter Loos

Institut für Wirtschaftsinformatik (IWi)
im Deutschen Forschungszentrum für Künstliche Intelligenz (DFKI)
66123 Saarbrücken
{thomas.matheis, dirk.werth, peter.loos}@iwi.dfki.de

Abstract

Die Bedeutung von Geschäftsmodellen des collaborative Business ist gewachsen. Kollaboratives Data Warehousing ermöglicht es den partizipierenden Unternehmen, sich in einem steigenden Wettbewerb strategische Wettbewerbsvorteile zu sichern, und sich damit in den stark umkämpften Kundenmärkten zu behaupten. Vor diesem Hintergrund werden Ansätze zum kollaborativen Data Warehousing vermehrt an Bedeutung gewinnen. Der vorliegende Beitrag stellt auf der Grundlage von föderierten Datenbanktechnologien einen Integrationsansatz zum kollaborativen Data Warehousing vor. Der Ansatz bezieht sowohl die Schemaebene, als auch die Datenebene in den Integrationsprozess ein, und erlaubt es insbesondere, dass ein beliebiges Unternehmen wieder aus der Kollaboration de-integriert werden kann. Über die Konzeption hinausgehend präsentiert der Beitrag auch einen Software-Prototypen, der für den Aufbau, den Betrieb und die Auflösung kollaborativer Data Warehouses eingesetzt werden kann.

1 Einleitung

Die fortschreitende Auflösung von Grenzen innerhalb und zwischen Unternehmen, sowie das Fortschreiten des technologischen Wandels, insbesondere der Internettechnologien, hat in den letzten Jahren zunehmend Einfluss auf die Gestaltung wertschöpfender Geschäftsprozesse ausgeübt [Sche+03]. Die dadurch entstandenen neuen Geschäftsmodelle werden unter dem

Schlagwort collaborative Business (c-Business) zusammengefasst und beinhalten Konzepte für die Zusammenarbeit über Unternehmensgrenzen hinweg [RöSc01, S. 289-292]. Im Mittelpunkt steht dabei die effiziente und effektive Gestaltung wertschöpfender Geschäftsprozesse, die nicht mehr nur unternehmensintern, sondern vor allem unternehmensübergreifend betrachtet werden. Die zunehmende Integration von einzelnen Unternehmen in global strategische Netzwerke ist eine globale Tendenz und eröffnet den beteiligten Unternehmen enorme Chancen. Nicht mehr einzelne Unternehmen werden künftig miteinander konkurrieren, sondern Netzwerke einzelner Unternehmen. Die Fähigkeit von Unternehmen, sich in kollaborative Netzwerke zu integrieren, stellt daher einen entscheidenden Schlüssel zu ihrem Geschäftserfolg dar [Öste+00]. Data-Warehouse-Systeme sind als Kern entscheidungs-unterstützender Informationssysteme für viele Unternehmen von strategischer Bedeutung [Lint01, S. 47f]. Da Informationen für Unternehmen immer wichtiger werden, um in Zeiten der Globalisierung auf dem Markt schnell reagieren zu können, ist das Vorhandensein aussagekräftiger Fakten für die strategische Entscheidungsfindung von großer Relevanz [AlÖs01]. Traditionelle Data-Warehouse-Lösungen sind in der Regel nur auf die Entscheidungsfindung einzelner Unternehmen ausgerichtet. Eine kollaborative Data-Warehouse-Lösung vereint die Daten mehrerer Unternehmen, die ihre Unternehmensdaten der Kollaboration über einzelne Data-Warehouse-Lösungen zur Verfügung stellen. Den an der Kollaboration beteiligten Unternehmen bietet eine kollaborative Data-Warehouse-Lösung bedeutende Möglichkeiten, um unternehmensübergreifende Entscheidungen zu treffen und strategische Wettbewerbsvorteile gegenüber ihren Konkurrenten zu erzielen [MaWe05].

Dieser Beitrag stellt auf der Grundlage von föderierten Datenbanktechnologien einen Ansatz für die flexible Integration von Data-Warehouse-Lösungen zu einer Data-Warehouse-Kollaboration vor. Kapitel 2 beschreibt die Methodik zur Integration multidimensionaler Datenmodelle, die die Grundlage des kollaborativen Data Warehousings bildet. Anschließend wird in Kapitel 3 die prototypische Realisierung des Integrationsansatzes vorgestellt. Der Beitrag schließt in Abschnitt 4 mit einer Zusammenfassung und einem Ausblick auf weitere Forschungsaufgaben.

2 Methodik zur Integration multidimensionaler Datenmodelle

Die Thematik der Schemaintegration ist bereits gut durchdrungen [Böhn01], so dass bei der Entwicklung einer Methode zur Integration multidimensionaler Datenmodelle ein kompletter

Neuentwurf nicht notwendig ist. Vielmehr kann ein bereits existierender Integrationsansatz um multidimensionale Konstrukte erweitert werden. Beispielhaft seien zur Schemaintegration Ansätze wie Upward Inheritance [Conr97], Correspondence Assertation [Spac+92] oder Generic Integration Model [Schm98] genannt. Für eine ausführliche Darstellung über Grundideen und Ansätze weiterer Integrationstechniken sei an dieser Stelle auf [RaBe01; Bati+86] verwiesen. Die zusicherungsbasierte Integration (Correspondence Assertation) ist unter den bestehenden Integrationsansätzen am ehesten geeignet, um als Grundlage für die Integration von multidimensionalen Datenmodellen zu dienen [MaWe05]. Daher wird in diesem Kapitel eine Methode für die Integration multidimensionaler Datenschemata vorgestellt, die auf dem Prinzip der zusicherungsbasierten Integration basiert. Bei diesem Ansatz werden auf der Grundlage eines generischen Datenmodells zwischen den zu integrierenden Schemata Inter-Schema-Korrespondenzen in Form von Zusicherungen definiert. Durch Anwendung von Integrationsregeln kann unter Einbezug der Zusicherungen das integrierte Schema konstruiert werden.

2.1 Vorüberlegungen und Voraussetzungen

Ein kollaboratives Data-Warehouse-System erfordert die Integration von mehreren Data Warehouses und somit die Integration von mehreren multidimensionalen Datenschemata. Ein wesentlicher Unterschied zwischen multidimensionalen Datenschemata und „einfachen“ Datenschemata besteht im Konzept der Summierbarkeit [LeSh97, S. 132-143]. Zentrales Ziel der Summierbarkeit ist es, die Korrektheit von Ergebnissen von Aggregatanfragen über multidimensionalen Daten zu garantieren. Beispielsweise kann die von den meisten multidimensionalen Datenmodellen verwendete Roll-Up-Beziehung, die eine Aggregation der Daten auf eine höhere Granularität beschreibt, nur dann sinnvoll eingesetzt werden, wenn eine korrekte Summierbarkeit vorausgesetzt werden kann [Lehn03]. Auch wenn vorausgesetzt wird, dass die lokalen Data-Warehouse-Datenquellen eine korrekte Summierbarkeit gewährleisten, das heißt die Bedingungen der Disjunktheit, Vollständigkeit und Typverträglichkeit erfüllen, so wird durch die Integration von mehreren multidimensionalen Schemata die Bedingung der Summierbarkeit in der Regel verletzt. Die dadurch entstehende Problematik im Rahmen der Anfragebearbeitung wird im Folgenden anhand des Beispiels der Abbildung 1 näher erläutert.

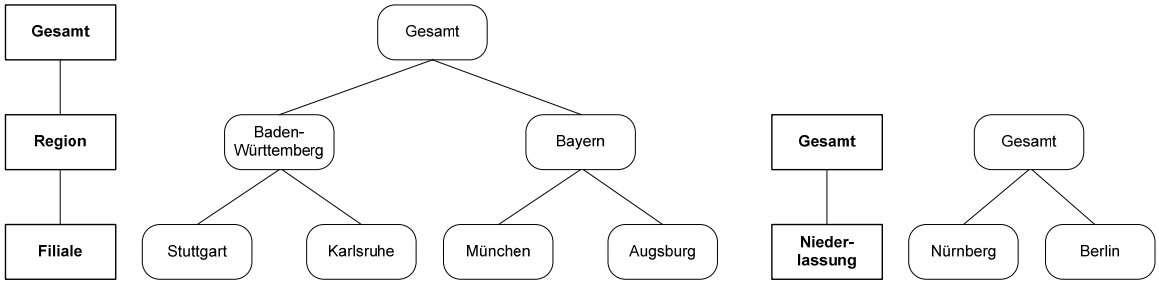

Abbildung 1: Zu integrierende multidimensionale Schemata

Die beiden Schemata der Abbildung 1 weisen die semantisch äquivalenten Schemaelemente ’Filiale’ und ’Niederlassung’ auf. Das Schemaelement ’Filiale’ bzw. ’Niederlassung’ wird somit in das integrierte Schema aufgenommen. Auf Instanzebene weist das integrierte Schema alle Instanzen der beiden Schemata auf der Ebene der Filialen bzw. Niederlassungen auf. Aufgrund der funktionalen Abhängigkeit [Lehn03] bestimmt beispielsweise die Instanz ’Stuttgart’ die übergeordnete Instanz ’Baden Württemberg’ funktional. Es ist allerdings unklar, welche Instanz des Attributes ’Region’ von den Instanzen ’Nürnberg’ und ’Berlin’ funktional bestimmt werden, da das Schemaelement ’Region’ im rechten Schema nicht enthalten ist. Diese Information muss bei der Integration erfasst werden, um das Konzept der Summierbarkeit, insbesondere der Vollständigkeit, zu gewährleisten. Die Vollständigkeit kann wieder hergestellt werden, indem die entsprechenden Instanzen den Instanzen der nächsthöheren Hierarchieebene direkt zugeordnet werden beziehungsweise entsprechende Instanzen der nächsthöheren Hierarchieebene neu definiert werden. In diesem Fall ergibt sich folgendes integriertes Schema mit den entsprechenden Instanzen, das in Abbildung 2 dargestellt ist.

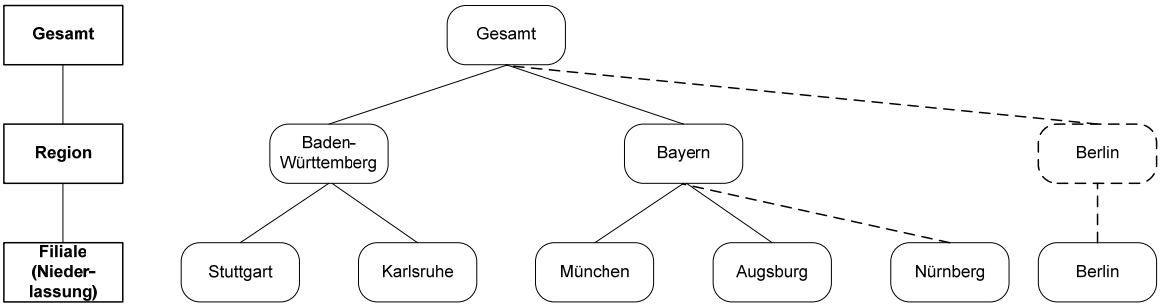

Abbildung 2: Integriertes Schema

Die Zuordnung einer Instanz zu einer Instanz der nächsthöheren Hierarchieebene muss meist manuell erfolgen, da hierfür semantisches Wissen erforderlich ist, das in der Regel nicht automatisch abgeleitet werden kann. Das wesentliche Ziel eines Data-Warehouse-Systems beziehungsweise eines kollaborativen Data-Warehouse-Systems, nämlich die Bereitstellung von aussagekräftigen Informationen zur Entscheidungsunterstützung, kann allerdings nur durch

diese Zuordnung der dimensionalen Instanzen erfüllt werden, da sonst keine aussagekräftigen Informationen bereitgestellt werden können.

Die oben dargestellte Problematik kommt besonders in einem kollaborativen Data-Warehouse-System zu tragen. Da an der Kollaboration eine Vielzahl von Unternehmen teilnehmen, die sich in die Kollaboration integrieren und auch wieder de-integrieren können, kann nicht davon ausgegangen werden, dass diese dieselben Schemaelemente in ihren multidimensionalen Schemata abgebildet haben. Um das Konzept der Summierbarkeit besser berücksichtigen zu können, bezieht der hier vorgeschlagene Integrationsansatz daher nicht nur die Schemaebene, sondern auch die Instanzebene der Dimensionshierarchien in den Integrationsprozess ein. Dabei spielt insbesondere das Konzept der funktionalen Abhängigkeit eine wesentliche Rolle. Durch Einbezug der Instanzebene in den Integrationsprozess erhöht sich zwar der Integrationsaufwand. Allerdings kann nur dadurch gewährleistet werden, dass im Rahmen einer Anfragebearbeitung aussagekräftige Anfrageergebnisse auf der Ebene der Kollaboration erzielt werden können. Durch Einschränkungen der zu integrierenden Daten kann der Integrationsaufwand wesentlich verringert und die praktische Anwendbarkeit des Ansatzes erhöht werden [MaWe05].

Die hier vorgestellte Methode beschränkt sich auf die Integration von multidimensionalen Schemata, die durch Fakten (Basiskennzahlen) und Dimensionen mit einfacher Hierarchie modelliert sind. Auf die Darstellung von abgeleiteten Kennzahlen, nicht dimensionalen Attributen und Dimensionen mit parallelen Hierarchiepfaden wurde bewusst verzichtet. Diese Konstrukte werden vom vorgestellten Integrationsansatz jedoch nicht ausgeschlossen, sondern können mit Hilfe weiterer Integrationsregeln ergänzt werden. Die Entscheidung für die Einschränkung liegt darin, dass sich mit Fakten und Dimensionen einfacher Hierarchie bereits die wesentlichen Konzepte zur Integration multidimensionaler Datenmodelle zeigen lassen.

2.2 Datenmodell

Um unabhängig von einem konkreten Datenmodell zu sein, werden die Zusicherungen bei der zusicherungsbasierten Integration auf der Grundlage des generischen Datenmodells formuliert. Das generische Datenmodell besitzt für multidimensionale Datenmodelle jedoch eine zu geringe Ausdrucksstärke. Daher wird im Folgenden in Anlehnung an die Arbeit von [Lehn03] ein multidimensionales Datenmodell beschrieben, das als Grundlage für die Formulierung der Zusicherungen und der Integrationsregeln dient.

An dieser Stelle muss zunächst das Prinzip der funktionalen Abhängigkeit erläutert werden, welches in den nachfolgenden Beschreibungen verwendet wird.

Definition Funktionale Abhängigkeit: Zwischen zwei Attributen A und B existiert eine funktionale Abhängigkeit ($A \rightarrow B$) genau dann, wenn für jede Instanz $a \in A$ genau eine Instanz $b \in B$ existiert. Das Attribut B wird damit von dem Attribut A funktional bestimmt.

Definition Schwache funktionale Abhängigkeit: Zwischen zwei Attributen A und B existiert eine schwache funktionale Abhängigkeit ($A \Rightarrow B$) genau dann, wenn für jede Instanz $a \in A$ höchstens eine Instanz $b \in B$ existiert.

Definition Datenwürfel: Das Schema eines Datenwürfels C besteht aus einer Menge von dimensionalen Schemata D und einer Menge von Fakten F. Dabei gilt: $C = (D,F) = (\{D_1,\ldots,D_n\},\{F_1,\ldots,F_n\})$.

Definition Dimension: Das Schema einer Dimension D besteht aus einer geordneten Menge von dimensionalen Attributen $(\{A_1,\ldots,A_n, TopA\};\rightarrow)$, wobei $\rightarrow$ die funktionale Abhängigkeit bezeichnet und TopA ein generisches maximales Element in Bezug auf $\rightarrow$ darstellt. Für zwei dimensionale Attribute A_i und A_{i+1} gilt, dass A_{i+1} von A_i funktional bestimmt wird. TopA wird von allen dimensionalen Attributen funktional bestimmt. Das dimensionale Attribut A_1 bestimmt alle anderen dimensionalen Attribute und stellt somit die feinste Granularität einer Dimension dar. Dabei gilt:

- $\forall i\ (1 \leq i < n)$: $A_i \rightarrow A_{i+1}$
- $\forall i\ (1 \leq i \leq n)$: $A_i \rightarrow Top_A$
- $\exists i\ (1 \leq i \leq n)\ \forall j\ (1 \leq j \leq n), i \neq j$: $A_i \rightarrow A_j$

Die Definition einer Dimension bezieht neben der Schemaebene die Instanzebene nur indirekt über die funktionale Abhängigkeit ein. Um auch die Instanzebene besser berücksichtigen zu können, wird die Definition einer Dimension wie folgt erweitert. Jedes dimensionale Attribut A_i besteht aus einer Menge von Instanzen: $A_i = \{a_1,\ldots,a_n\}$.

Zu beachten ist an dieser Stelle, dass die Definition eines Datenwürfels damit nicht nur die Schemaebene, sondern auch die Instanzebene der dimensionalen Attribute einbezieht.

2.3 Korrespondenz-Zusicherungen

Korrespondenz-Zusicherungen setzen die Bestandteile von Datenwürfeln zueinander in Beziehung. Es werden zwei Arten von Korrespondenz-Zusicherungen unterschieden.

Zusicherungen, die über Attribute Dimensionen in Beziehung zueinander setzen, sowie Zusicherungen, die Fakten in Beziehung zueinander setzen. Im Folgenden werden die verschiedenen Korrespondenz-Zusicherungen vorgestellt.

Dimension-Korrespondenz-Zusicherung: Seien A und B zwei Dimensionen, wobei A aus dem Datenwürfel C1 und B aus dem Datenwürfel C2 stammt. Es gilt: $A=(\{A_1,\ldots,A_n, TopA\};\rightarrow)$ und $B=(\{B_1,\ldots,B_n, TopB\};\rightarrow)$. Dann gibt es zwei Möglichkeiten, um Korrespondenzen zwischen A und B durch Zusicherungen zu formulieren.

- $A \leftrightarrow B$: Die Dimensionen A und B repräsentieren eine semantisch äquivalente Dimensionshierarchie.
- $A \updownarrow$: Die Dimension A repräsentiert eine Dimensionshierarchie, die zu keiner Dimension des Datenwürfels C2 semantisch äquivalent ist. Diese Zusicherung ermöglicht es, neue Dimensionen in den integrierten Datenwürfel aufzunehmen beziehungsweise einzelne Attribute einer Dimension von A in eine neue Dimension aufzunehmen.

Attribut-Korrespondenz-Zusicherung: Seien A_i und B_j zwei beliebige Attribute, wobei A_i aus der Dimension A und B_j aus der Dimension B stammt. Dann gibt es die folgenden Möglichkeiten, um Korrespondenzen zwischen A_i und B_j durch Zusicherungen zu formulieren.

- $A_i \leftrightarrow B_j$: Die Attribute A_i und B_j sind semantisch äquivalent, das heißt sie repräsentieren die gleiche Menge von Instanzen. Die Menge D bezeichnet dabei alle Instanzen der Attribute A_i und B_j, die zueinander semantisch äquivalent sind. Es gilt: $D = \{(a,b) \mid a \in A_i, b \in B_j, a \text{ semantisch äquivalent zu } b\}$.
- $A_i \rightarrow B_j$: Das Attribut B_j wird von dem Attribut A_i direkt funktional bestimmt. Das Attribut A_i repräsentiert also eine Menge von Instanzen, die die Menge von Instanzen von B_j direkt funktional bestimmen.
- $A_i \rightarrow TopB$: TopB wird von dem Attribut A_i direkt funktional bestimmt. Das Attribut A_i repräsentiert also eine Menge von Instanzen, die von allen Mengen von Instanzen der Attribute aus B funktional bestimmt werden.
- $A_i \updownarrow$: Das Attribut A_i lässt sich nicht durch eine der vorangegangenen Zusicherungen einem Attribut der Dimension B zuordnen.

Die Attribut-Zusicherungen können den Dimension-Zusicherungen wie folgt angefügt werden:

- $A \leftrightarrow B$ mit $A_i \leftrightarrow B_j$, $A_i \rightarrow B_j$, $A_i \rightarrow$ TopB oder $A_i \updownarrow$
- $A \updownarrow$ mit $A_i \leftrightarrow A_i$ oder $A_i \updownarrow$

Fakten-Korrespondenz-Zusicherung: Seien F1 und F2 zwei Fakten, wobei F1 aus dem Datenwürfel C1 und F2 aus dem Datenwürfel C2 stammt. Dann gibt es zwei Möglichkeiten, um Korrespondenzen zwischen F1 und F2 durch Zusicherungen zu formulieren.

- $F1 \leftrightarrow F2$: Die Fakten F1 und F2 repräsentieren eine semantisch äquivalente Basiskennzahl.
- $F1 \updownarrow$: Das Faktum F1 repräsentiert eine Basiskennzahl, die zu keinem Faktum des Datenwürfels C2 semantisch äquivalent ist.

2.4 Integration

Mit Hilfe der folgenden Integrationsregeln kann der integrierte Datenwürfel basierend auf den vorgestellten Zusicherungen schrittweise konstruiert werden. Dabei werden sowohl Veränderungen auf Schema- als auch auf Instanzebene angegeben. Die Dimensionen der zu integrierenden Datenwürfel weisen eine funktionale Abhängigkeit auf, während die Dimensionen des integrierten Datenwürfels durch das Anwenden der Integrationsregeln in der Regel nur eine schwache funktionale Abhängigkeit aufweisen. Deshalb wird auch insbesondere aufgezeigt, welche Integrationsregeln die funktionale Abhängigkeit verletzen können.
Seien die Datenwürfel C1 und C2 gegeben. Der Datenwürfel C1 wird in den Datenwürfel C2, der zum Beispiel den bestehenden Datenwürfel einer Kollaboration beschreibt, integriert. Die nachfolgend angegebenen Integrationsregeln beschreiben, wie sich der Datenwürfel C2 durch die Integration von C1 verändert. Die Dimension A repräsentiert dabei eine Dimension des Datenwürfels C1, die Dimension B eine Dimension des Datenwürfels C2.
Integrationsregel 1: Alle Fakten, Dimensionen und Attribute des Datenwürfels C2, zu denen keine Korrespondenz zu dem Datenwürfel C1 besteht, bleiben durch die Integration des Datenwürfels C1 in den Datenwürfel C2 unverändert. Auch die Instanzen der Attribute, die nicht mit dem Datenwürfel C1 korrespondieren, bleiben durch die Integration unverändert erhalten.
Integrationsregel 2: Sei die Zusicherung $A \leftrightarrow B$ mit $A_i \leftrightarrow B_j$ und der Menge D gegeben. Die Menge $DIFF_A$ bezeichnet die Menge von Instanzen von A_i, die nicht semantisch äquivalent zu Instanzen von B_j sind. Es gilt: $DIFF_A = \{a |\ a \in A_i, (a_d, b) \in D, a \neq a_d\}$. Analog bezeichnet die

Menge $DIFF_B$ die Menge von Instanzen von B_j, die nicht semantisch äquivalent zu Instanzen von A_i sind. Es gilt: $DIFF_B=\{b|\ b \in B_j, (a,b_d) \in D, b \neq b_d\}$. Das Schema der Dimension B bleibt durch das Anwenden der Integrationsregel unverändert. Auf Instanzebene wird das Attribut B_j um alle Instanzen der Menge $DIFF_A$ erweitert. Die funktionale Abhängigkeit kann bei dieser Integrationsregel in folgenden Fällen verletzt werden.

- Existiert eine Zusicherung der Form $A_{i+1} \rightarrow B_{j+1}$, so existieren für alle Instanzen aus B_j, die zu der Menge $DIFF_B$ gehören, keine funktional abhängigen Instanzen aus dem übergeordneten Attribut A_{i+1}. Dies trifft ebenfalls zu, falls B_{j+1} TopB entspricht.

- Existiert ein Attribut B_{j+1} und keine Zusicherung der Form $A_{i+1} \leftrightarrow B_{j+1}$ oder $A_{i+1} \rightarrow B_{j+1}$, so existieren zu allen Instanzen aus B_j, die zu der Menge $DIFF_A$ gehören, keine funktional abhängigen Instanzen aus B_{j+1}.

Integrationsregel 3: Sei die Zusicherung $A \leftrightarrow B$ mit $A_i \rightarrow B_j$ gegeben. Die Dimension B wird um das Attribut A_i erweitert, so dass B_j von A_i direkt funktional bestimmt wird. Alle Instanzen des Attributes A_i werden übernommen. Die funktionale Abhängigkeit kann bei dieser Integrationsregel in folgenden Fällen verletzt werden.

- Existiert keine weitere Zusicherung der Form $A_{i+1} \leftrightarrow B_j$ oder $A_{i+1} \rightarrow B_j$, so existieren für alle Instanzen aus A_i keine funktional abhängigen Instanzen aus B_j.

- Existiert ein Attribut B_{j-1} und keine Zusicherung Form $A_{h,h<i} \leftrightarrow B_{j-1}$, so existieren für alle Instanzen aus B_{j-1} keine funktional abhängigen Instanzen aus A_i. Ausnahme: Existiert zusätzlich eine Zusicherung der Form $A_{h,h<i} \leftrightarrow B_{k,k<j-1}$ mit $DIFF_A$ und $DIFF_B$, so existieren nur für die Instanzen aus B_{j-1}, keine funktional abhängigen Instanzen aus A_i, die von den Instanzen der Menge $DIFF_B$ funktional bestimmt werden.

Integrationsregel 4: Sei die Zusicherung $A \leftrightarrow B$ mit $A_i \rightarrow TopB$ gegeben. Die Dimension B wird um das Attribut A_i erweitert, so dass TopB von A_i direkt funktional bestimmt wird. Alle Instanzen des Attributes A_i werden übernommen. Die funktionale Abhängigkeit kann bei dieser Integrationsregel in folgendem Fall verletzt werden.

- Existiert keine weitere Zusicherung der Form $A_{h,h<i} \rightarrow TopB$ und keine Zusicherung der Form $A_{h,h<i} \leftrightarrow B_n$ für das Attribut B_n, das TopB direkt funktional

bestimmt, so besitzen alle Instanzen des Attributes B_n keine funktional abhängigen Instanzen aus A_i. Ausnahme: Existiert zusätzlich eine Zusicherung der Form $A_{h,h<i} \leftrightarrow B_{k,k<n-1}$ mit $DIFF_A$ und $DIFF_B$, so existieren nur für die Instanzen aus B_n keine funktional abhängigen Instanzen aus A_i, die von den Instanzen der Menge $DIFF_B$ funktional bestimmt werden.

Integrationsregel 5: Sei die Zusicherung $A \leftrightarrow B$ mit $A_i\updownarrow$ gegeben. Die Dimension B bleibt auf Schema- und Instanzebene unverändert. Die Dimension B wird nicht um das Attribut A_i erweitert, da nur Dimensionen mit einfachen Hierarchiepfaden und somit keine parallelen oder alternativen Hierarchiepfade entstehen sollen. Das Konzept der funktionalen Abhängigkeit bleibt in diesem Fall erhalten.

Integrationsregel 6: Sei die Zusicherung $A\updownarrow$ mit $A_i \leftrightarrow A_i$ beziehungsweise $A_i\updownarrow$ gegeben. Der Datenwürfel C2 wird um die Dimension A erweitert. Das Attribut A_i wird im Fall von $A_i \leftrightarrow A_i$ mit in die Dimension übernommen, im Fall von $A_i\updownarrow$ nicht in die Dimension mit aufgenommen. Das Konzept der funktionalen Abhängigkeit bleibt in diesem Fall erhalten.

Integrationsregel 7: Sei die Zusicherung $F1 \leftrightarrow F2$ gegeben. Der Datenwürfel C2 bleibt unverändert.

Integrationsregel 8: Sei die Zusicherung $F\updownarrow$ gegeben. Der Datenwürfel C2 wird um das Faktum F2 erweitert.

Im Rahmen der Integration müssen neben den Zusicherungen ferner noch Abbildungsinformationen zwischen dem integrierten Datenwürfel und den lokalen Datenwürfeln festgehalten werden, die notwendig sind, um Anfragen an den integrierten Datenwürfel korrekt auf die einzelnen lokalen Datenwürfel zu transformieren. Die De-Integration eines Datenwürfels aus einem integrierten Datenwürfel, der nach dem vorgestellten Integrationsansatz konstruiert wurde, kann mit Hilfe von De-Integrationsregeln durchgeführt werden. Auf die Darstellung der zu erfassenden Abbildungsinformationen und der Methodik zur De-Integration wird an dieser Stelle nicht näher eingegangen.

2.5 Beispiel

Die eingeführten Zusicherungen und Integrationsregeln werden anhand des folgenden Beispiels verdeutlicht. Die Abbildung 3 zeigt die Schemata zweier Datenwürfel, die im multidimensionalen E/R-Modell (MERM) [Dete02] modelliert sind, sowie beispielhaft einige dimensionale Instanzen. Der Datenwürfel C1 soll in den Datenwürfel C2 integriert werden.

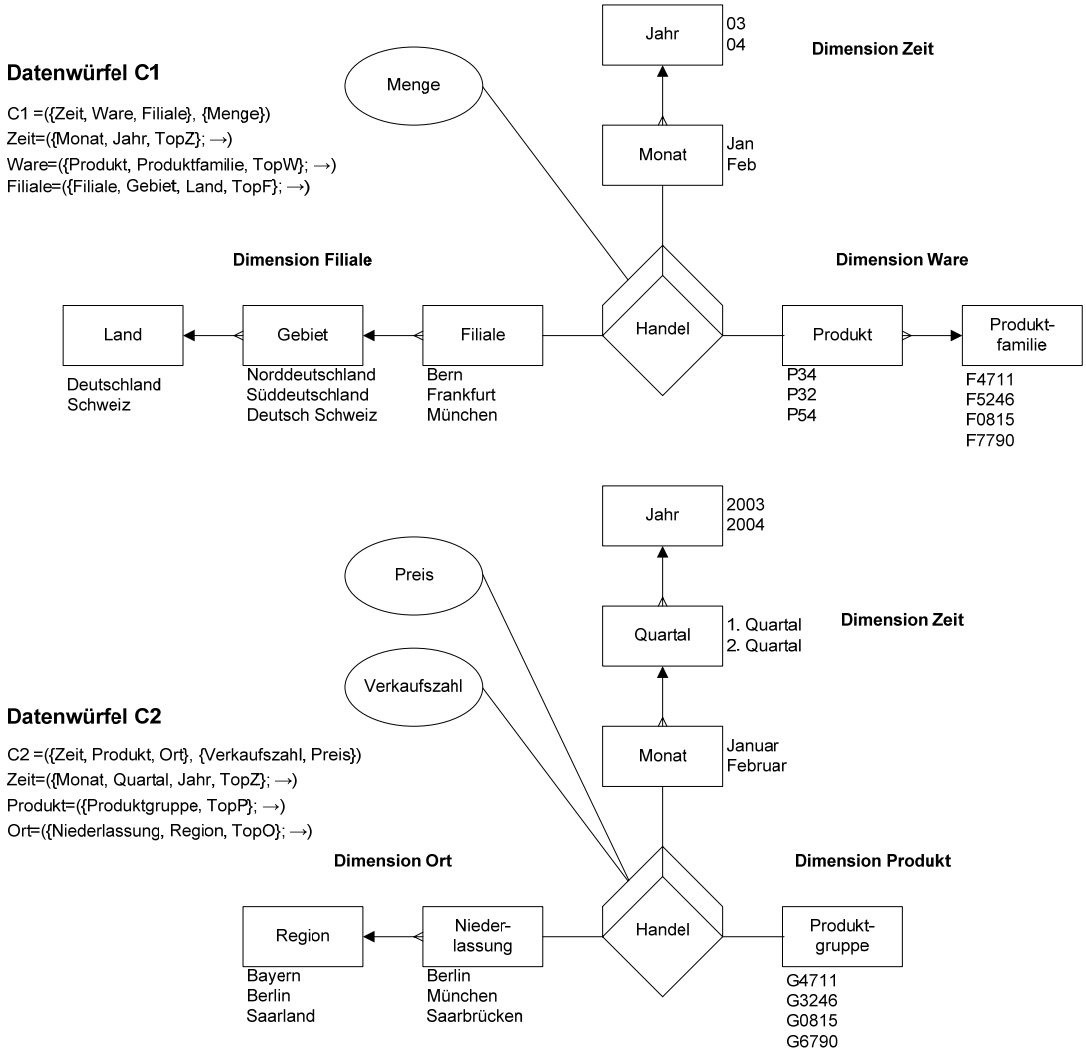

Abbildung 3: Zu integrierende Datenwürfel

Bevor die Integration durchgeführt werden kann, müssen zwischen den beiden Datenwürfeln die Korrespondenzen in Form von Zusicherungen bestimmt werden. Zwischen den beiden Datenwürfeln bestehen die folgenden Korrespondenz-Zusicherungen (siehe Abbildung 4).

1) C1.Zeit ↔ C2.Zeit mit C1.Monat ↔ C2.Monat und D={(Jan,Januar),(Feb,Februar)}
2) C1.Zeit ↔ C2.Zeit mit C1.Jahr ↔ C2.Jahr und D={(03,2003),(04,2004)}
3) C1.Ware ↔ C2.Produkt mit C1.Produkt → C2.Produktgruppe
4) C1.Ware ↔ C2.Produkt mit C1.Produktfamilie ↔ C2.Produktgruppe und D={(F4711,G4711),(F0815,G0815)}
5) C1.Filiale ↔ C2.Ort mit C1.Filiale ↔ C2.Niederlassung und D={(München,München)}
6) C1.Filiale ↔ C2.Ort mit C1.Gebiet ↕
7) C1.Filiale ↔ C2.Ort mit C1.Land → C2.TopO
8) C1.Menge ↔ C2.Verkaufszahl

Abbildung 4: Korrespondenz-Zusicherungen

Mit Hilfe der Zusicherungen und unter Anwendung der Integrationsregeln kann nun der Datenwürfel C1 in den Datenwürfel C2 integriert werden. Die Integration wird hierbei am

Beispiel der Dimensionen Ort und Filiale verdeutlicht. Die Integrationsregel 2 wird auf die Zusicherung 5 angewandt. Das Schema der Dimension ’Ort’ bleibt erhalten. Auf Instanzebene werden dem Attribut ’Niederlassung’ die Instanzen ’Bern’ und ’Frankfurt’ zugefügt. Da das Attribut ’Region’ existiert, aber keine Zusicherung der Form $\leftrightarrow$ oder $\rightarrow$, die sich auf das Attribut ’Region’ bezieht, existieren zu den Instanzen ’Frankfurt’ und ’Bern’ keine funktional abhängigen Instanzen des Attributes ’Region’. Durch Anwenden der Integrationsregel 5 auf die Zusicherung 6 bleibt der integrierte Datenwürfel auf Schema- und Instanzebene unverändert, da das Attribut ’Gebiet’ nicht aufgenommen wird. Die Integrationsregel 4 verursacht bei Anwendung auf die Zusicherung 7 folgende Änderungen. Die Dimension ’Ort’ wird um das Attribut ’Land’ erweitert. ’TopO’ wird nun von dem Attribut ’Land’ direkt funktional bestimmt. Alle Instanzen des Attributes ’Land’ werden übernommen. Da keine weitere Zusicherung für ’TopO’ und keine Zusicherung der Form $\leftrightarrow$ für das Attribut ’Region’ existiert, besitzen alle Instanzen des Attributes ’Region’ keine funktional abhängigen Instanzen aus dem Attribut ’Land’. Da aber zusätzlich die Zusicherung 5 besteht, tritt die Ausnahme der Integrationsregel 4 ein, so dass nicht alle Instanzen des Attributes ’Region’, sondern nur die Instanzen ’Berlin’ und ’Saarbrücken’ keine funktional abhängigen Instanzen aus dem Attribut ’Land’ besitzen’. Das Ergebnis der Integration ist in der Abbildung 5 dargestellt. Die in der Abbildung 5 kursiv dargestellten Bestandteile stellen synonyme Bezeichnungen dar, die im Rahmen der Integration durch die Zusicherungen aufgenommen werden.

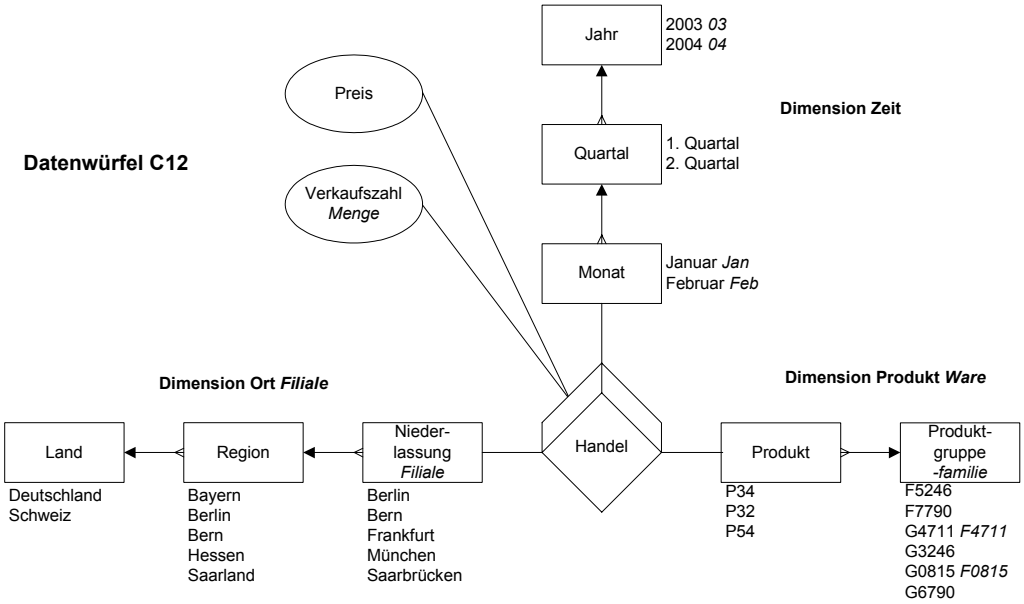

Abbildung 5: Integrierter Datenwürfel

3 Prototypische Implementierung

3.1 Architektur

Zur Evaluation der Integrationsmethode wurde das Konzept in einem Software-Prototyp umgesetzt. Dabei wurde von einer organisatorischen Umgebung ausgegangen, in der mehrere Unternehmen unabhängige und autonome Data Warehouses bzw. Data Marts betreiben. Systemseitig impliziert dies eine vollständige Kapselung der Datenquellen. Diese Kapselung sowie die Software-Funktionalitäten, die zum Aufbau, zum Betrieb und zur Auflösung kollaborativer Data Warehouses benötigt werden, sind die Hauptaufgaben des Prototyps. Einen Überblick über die Systemumgebung ist in der Abbildung 6 dargestellt.

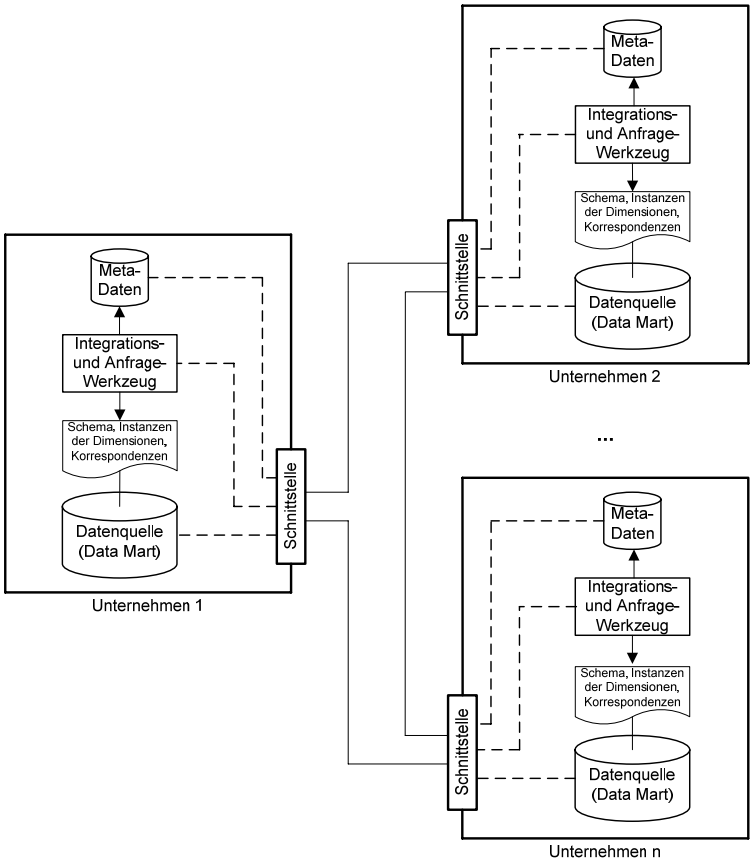

Abbildung 6: Systemumgebung

Jedes an der Kollaboration beteiligte Unternehmen besitzt als zentrale Komponente ein Werkzeug zur Unterstützung des Integrationsprozesses und der Anfragebearbeitung. Tritt ein Unternehmen der Kollaboration bei, so müssen zunächst der zu integrierende Datenwürfel und die zugehörigen Korrespondenzen in Form von XML-Dokumenten erstellt werden. Das Werkzeug liest aus den XML-Dokumenten den lokalen Datenwürfel, die Korrespondenzen sowie aus Metadaten den bereits bestehenden kollaborativen Datenwürfel ein. Unter

Anwendung von Integrationsregeln führt das Werkzeug anschließend die Integration durch. Der aus der Integration resultierende Datenwürfel wird in den Metadaten gespeichert. Entsprechend der Definition eines Datenwürfels werden nicht nur das Schema des kollaborativen Datenwürfels, sondern auch alle Instanzen der dimensionalen Attribute mit ihren funktionalen Abhängigkeiten in den Metadaten abgelegt. Jedes Unternehmen muss ferner eine Schnittstelle besitzen, über die es seine Daten der Kollaboration verfügbar macht. Auf der Grundlage des in den Metadaten gespeicherten kollaborativen Datenwürfels können im Werkzeug Anfragen an die Kollaboration gestellt werden. Das Werkzeug transformiert die globale Anfrage in Teilanfragen, liest über die entsprechenden Schnittstellen die lokalen Daten ein und transformiert diese zu einem globalen Ergebnis. Im Rahmen der prototypischen Realisierung wird ein Werkzeug zur Unterstützung des Integrationsprozesses und der Anfragebearbeitung realisiert.

Die Abbildung 7 gibt einen Überblick über die System-Architektur des entwickelten Werkzeuges zur Unterstützung des Integrationsprozesses und der Anfragebearbeitung. Die wesentlichen Komponenten der Architektur sind die Integrationskomponente und die Anfragekomponente. Die Integrationskomponente setzt sich aus dem Integrator und der Importkomponente zusammen, die Anfragekomponente aus dem Parser, der Anfragezerlegungskomponente und der Ausführungs- und Auswertungskomponente. Der integrierte Datenwürfel sowie alle benötigten Abbildungsinformationen werden in einer Datenbank abgelegt, so dass sie für die Anfragebearbeitung genutzt werden können. Über das Integrations-Interface kann der Administrator den Integrationsprozess steuern. Der Anwender kann über das Query-Interface SQL-Anfragen in Form von Star-Queries an die Kollaboration stellen. Bei der Entwicklung des Werkzeuges stehen die Anwendung der Integrationsmethode und die Behandlung von Konflikten auf Schema- und Instanzebene im Rahmen der Anfragebearbeitung im Mittelpunkt der Betrachtung. Auf die Entwicklung von Adaptern für unterschiedliche Datenmodelle und Datenbanksysteme wird daher verzichtet.

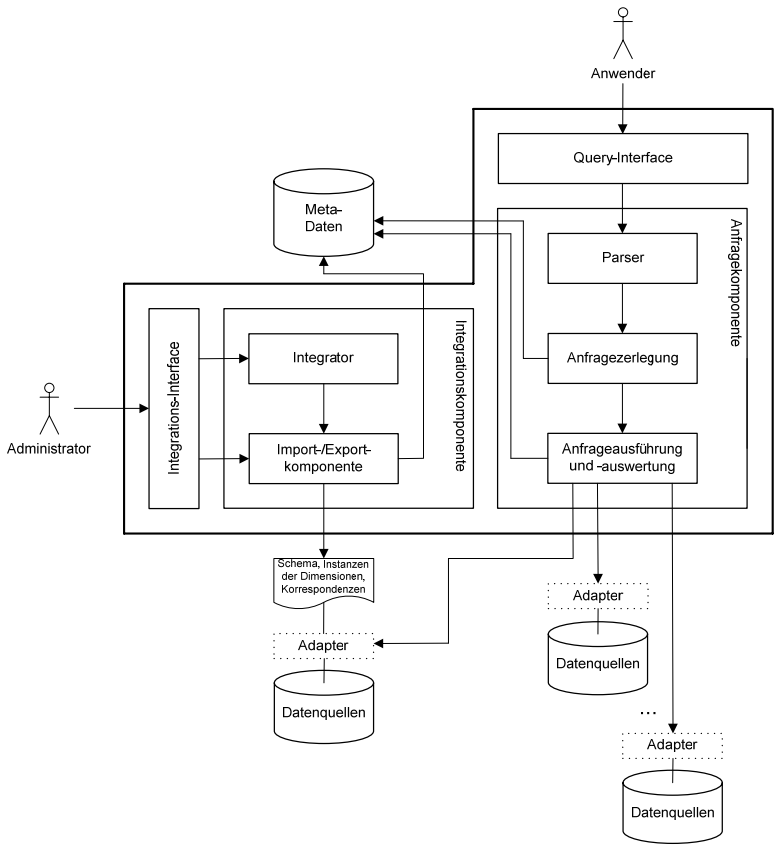

Abbildung 7: System-Architektur

3.2 Umsetzung

Die aktuelle Implementierung beinhaltet dieses Werkzeug, das den Anwender bzw. den Administrator bei der Durchführung des Integrations- und De-integrationsprozesses unterstützt sowie Abfragen auf dem Datenbestand des kollaborativen Data Warehouses ausführen kann. Die Korrespondenzen sowie die benötigten Informationen über das lokale Data Warehouse müssen vom Administrator über eine XML-Spezifikation in das Tool geladen werden. Das Tool führt anschließend durch Anwendung der Integrationsregeln die Integration durch. Dabei werden dem Administrator insbesondere auch die Stellen angezeigt, an welchen die funktionale Abhängigkeit innerhalb des Datenwürfels verletzt wird. Zur manuellen Behebung wird der Administrator durch Assistenten unterstützt, die ihn interviewartig durch den Prozess leiten. Die De-Integration kann im Tool mit Hilfe von De-Integrationsregeln durchgeführt werden.

Des Weiteren wurde im Tool eine Anfragebearbeitung realisiert. Das Tool bietet dem Anwender dabei die Möglichkeit, Anfragen an die Data-Warehouse-Kollaboration zu stellen. Auf der Basis der Abbildung zwischen globalem Datenwürfel und lokalen Datenwürfeln, die im Integrationsprozess definiert wurde, wird die globale Anfrage des Anwenders in lokale

Teilanfragen zerlegt [Satt+00]. Die lokalen Anfragen werden auf den lokalen Data-Warehouse-Systemen ausgeführt. Die lokalen Ergebnisse werden anschließend anhand der Abbildungsinformationen zu dem globalen Ergebnis zusammengesetzt. Die Abbildung 8 zeigt einen Ausschnitt des Tools.

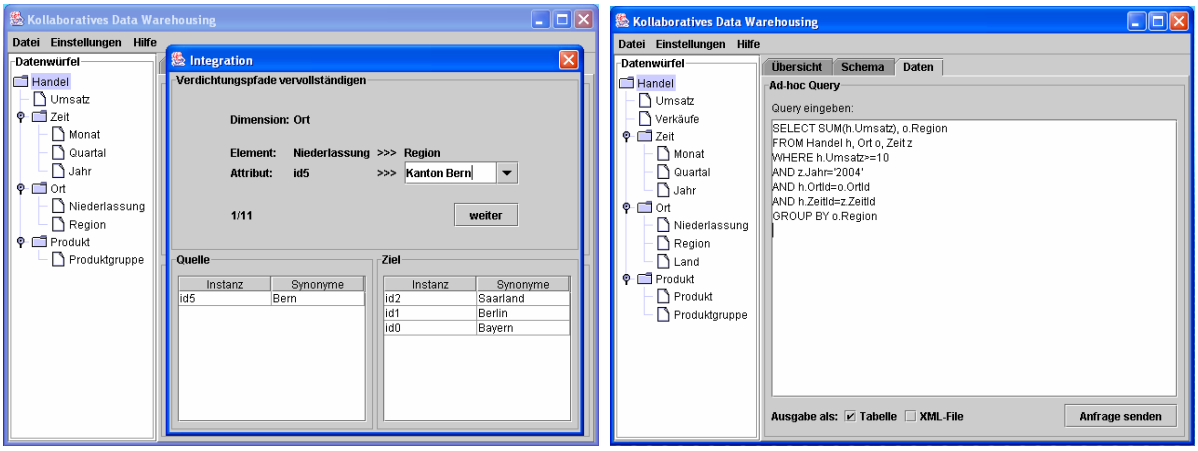

Abbildung 8: Screenshot

4 Zusammenfassung und Ausblick

In dem vorliegenden Beitrag wird auf der Grundlage von föderierten Datenbanktechnologien ein Ansatz für den Aufbau, den Betrieb und die Auflösung einer Data-Warehouse-Kollaboration vorgestellt. Im Mittelpunkt steht dabei der Integrationsansatz. Der Ansatz bietet Unternehmen die Möglichkeit sich flexibel in die Data-Warehouse-Kollaboration zu integrieren. Bestehende unternehmenseigene Data-Warehouse-Lösungen bleiben unverändert. Unternehmensnetzwerke können aus der Kollaboration strategische Informationen gewinnen und dadurch ihre Wettbewerbsposition verbessern.

Im Rahmen zukünftiger Arbeiten sollten noch weitere Aspekte Beachtung finden. Der vorgestellte Integrationsansatz erlaubt nur die Integration von Datenwürfeln, die durch Fakten und Dimensionen mit einfacher Hierarchie modelliert sind. Die Aufnahme weiterer Modellierungsmöglichkeiten bezüglich der Datenwürfel, wie zum Beispiel abgeleitete Kennzahlen oder Dimensionen mit parallelen oder alternativen Hierarchiepfaden, sollten daher noch weiter untersucht werden. Weiterhin sollte die vorgestellte Methodik in ein umfassendes Vorgehensmodell eingebettet werden, das weitere organisatorische und informationstechnische Aspekte des kollaborativen Data Warehousing aufgreift.

Literaturverzeichnis

[AlÖs01] Alt, R.; Österle, H.: Real-Time Business: Lösungen, Bausteine und Potenziale des Business Networking. Springer, Berlin et al., 2004.

[Bati+86] Batini, C.; Lenzerini, M., Navathe, S.: A Comparative Analysis of Methodologies for Database Schema Integration. In: ACM Computing Surveys, 18(4), 1986, S. 323-364.

[Böhn01] Böhnlein, M: Konstruktion semantischer Data-Warehouse-Schemata. Forschungsbeiträge zur Wirtschaftsinformatik, 1.Auflage, Deutscher Universitäts-Verlag, Wiesbaden, 2001.

[Conr97] Conrad, S.: Föderierte Datenbanksysteme. Konzepte der Datenintegration. Springer, Berlin et al., 1997.

[Dete02] Determann, L.: Modellierung analytischer Informationssysteme – Ein Konzept zur multidimensionalen Datenstrukturierung. Deutscher Universitätsverlag, Wiesbaden, 2002.

[Lehn03] Lehner, W.: Datenbanktechnologien für Data-Warehouse-Systeme. Konzepte und Methoden. dpunkt, Heidelberg, 2003.

[LeSh97] Lenz, H.-J.; Shoshani, A.: Summarizability in OLAP and Statistical Data Bases. In: Proceedings of the 9th International Conferenz on Statistical and Scientific Database Management, 1997, S. 132-143.

[Lint01] Linthicum, D.: B2B Application Integration: e-Business-Enable Your Enterprise. Addison-Wesley, Boston et al., 2001, S. 48-51.

[MaWe05] Matheis, T.; Werth, D.: Konzeption und Potenzial eines kollaborativen Data Warehouse-Systems. In: Loos, P. (Hrsg.): Veröffentlichungen des Instituts für Wirtschaftsinformatik, Nr. 185, Universität des Saarlandes, Saarbrücken, 2005.

[Öste+00] Österle, H.; Fleisch, E.; Alt, R.: Business Networking - Shaping Collaboration Between Enterprises. Springer, Berlin et al., 2000.

[RaBe01] Rahm, E.; Bernstein, P.: A Survey of Approaches to Automatic Schema Matching. In: The VLDB Journal, 10(4), 2001, S. 334-350.

[RöSc01] Röhricht, J.; Schlögel C.: cBusiness. Erfolgreiche Internetstrategien durch Collaborative Business am Beispiel der mySAP.com. Addison-Wesley, München et al., 2001.

[Sche+03] Scheer A.-W.; Adam, O.; Hofer, A.; Zangl, F.: Nach Cost Cutting – Aufbruch durch Innovation. In: IM - Fachzeitschrift für Information Management & Consulting, (18), 2003, gleichzeitig Proceedings zur 24. Saarbrücker Arbeitstagung 2003.

[Schm98] Schmitt, I.: Schemaintegration für den Entwurf föderierter Datenbanken. Dissertationen zu Datenbanken und Informationssystemen DISDBIS, Band 43, infix, Sankt Augustin, 1998.

[Satt+00] Sattler, K.-U.; Conrad, S.; Saake, G.: Adding Conflict Resolution Features to a Query Language for Database Federations. In: Proceedings of the 3nd International Workshop on Engineering Federated Information Systems, 2000, S. 41-52.

[Spac+92] Spaccapietra, S.; Parent, C.; Dupont, Y.: Model Independent Assertions for Integration of Heterogeneous Schemas. In: VLDB Journal, 1(1), 1992, S. 81-126.