# Datenintegration zur Anfragezeit[1]

Julian Eberius[2]

**Abstract:** In der Big-Data-Ära werden neue Daten oft in einer Geschwindigkeit gesammelt, die klassische Integration mit statischen ETL-Prozessen und globalen Schemata nicht mehr erlaubt. Diese Arbeit stellt das Prinzip der *Datenintegration zur Anfragezeit* vor, das darauf abzielt, zur Laufzeit einer Datenbankanfrage zusätzliche externe Datenquelle zu integrieren, und diese direkt im Anfrageergebnis darzustellen. Um dieses Ziel zu erreichen, wurde eine Reihe neuer Methoden, Algorithmen und Systeme entwickelt. An erster Stelle steht ein *Top-k-Entity-Augmentation-System*, das es ermöglicht, einen Datensatz ad hoc um neue Attribute zu erweitern. Darauf aufbauend wurde ein Datenbanksystem weiterentwickelt, das sogenannte Open-World-SQL-Anfragen verarbeitet, also Anfragen die über das definierte Schema hinausgehen. Die letzte Komponente ist ein *Datenkurationssystem*, das darauf zielt, die individuelle Nachnutzbarkeit heterogener Datenbestände für die Ad-hoc-Integration zu erhöhen, ohne jedoch ein zentrales Schema vorauszusetzen.

## 1 Einführung

Der Begriff *Big Data* wird meistens mit den Chancen und Risiken der immer schneller wachsenden Datenvolumen, die heute gesammelt werden, in Verbindung gebracht. Tatsächlich beschreibt er neben dem wachsenden Volumen und der zunehmenden Dynamik aber auch noch den Aspekt der wachsenden *Vielfältigkeit* der gesammelten Daten [La01]. Dabei treten sowohl immer verschiedenere Datenarten, -formate und Schemata, als auch immer heterogenere Datenquellen auf. Das Spektrum neuer Datenqellen reicht von großangelegten Sensornetzwerken über Messdaten von mobilen Endgeräten oder Industriemaschinen, bis hin zu Log- und Clickströmen der immer komplexer werdenden Softwarearchitekturen und -applikationen. Hinzu kommen öffentlich zugängliche Daten, wie etwa soziale Netzwerkdaten sowie Web- und Open Data. Auch wenn die Wertschöpfung aus diesen neuen Datenformen nicht trivial ist, ist ihr Potenzial allgemein anerkannt [LJ12]. Eine besondere Herausforderung die sich aus dem *Vielfältigkeitsaspekt* von Big Data ergibt, ist die Heterogenität und Vielfalt der verwendeten Datenquellen und damit das Problem der *Datenintegration*. Diese Arbeit stellt diesen Aspekt unter besonderer Berücksichtigung von sich ändernden Datenmanagementprozessen und -praktiken in den Fokus.

Datenintegration ist ein häufiges Problem bei der Datenverwaltung, das sich mit der Kombination von Daten unterschiedlicher Herkunft und ihrer Überführung in eine einheitlich nutzbare Form beschäftigt. Im Allgemeinen ist es ein mühsamer und meist manueller Prozess, der *vorzeitig* durchgeführt werden muss, das heißt, bevor

[1] Englischer Titel der Dissertation: “Query-Time Data Integration” [Eb15]

[2] Lehrstuhl für Datenbanken, TU Dresden, julian.eberius@tu-dresden.de

Anfragen auf den kombinierten Daten ausgeführt werden können. Aufgrund seiner Komplexität wird er üblicherweise von Fachleuten durchgeführt, beispielsweise ETL- und Integrationsspezialisten. Zugleich werden datenbasierte Ansätze in mehr und mehr Kontexten verwendet und beziehen immer mehr Anwender außerhalb der traditionellen IT-community ein. Neue, *agile* Datenmanagement-Ansätze, wie etwa MAD [Co09], ergänzen oder ersetzen zunehmend die statischen Prozesse der Data Warehouse Infrastrukturen. Mehr und mehr setzt sich die Überzeugung durch, dass alle Arten von Organisationen davon profitieren können, ihren Domänenexperten die Möglichkeit zu eigenem Datenmanagement und zur eigenen Datenanalyse zu geben, ohne dass diese in hohem Maße IT-Personal miteinbeziehen müssen [MB12].

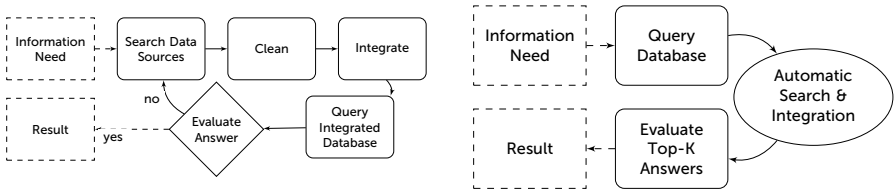
Demgegenüber stehen die konventionellen Dateninfrastrukturen, die kontrollierte ETL-Prozesse mit wohldefinierten Quell- und Zielschemata voraussetzen, und die noch immer die Systemlandschaften in den meisten Organisationen dominieren. Ihre Schemata definieren unmittelbar was für einen Analysten anfragbar ist. Beim Auftreten eines situativen Informationsbedürfnis, das mit dem vorhandenen Schema nicht befriedigt werden, müssen komplexe Prozesse durchlaufen werden um etwa externe Informationsquellen zu integrieren. Weil das Warehouse oft ein geschäftsentscheidendes System ist, wird es meist hochgradig reglementiert und kontrolliert und ist daher für die Ad-hoc-Integration völlig ungeeignet. Tatsächlich wäre eine statische Integration in vielen Fällen nicht einmal wünschenswert, da die zukünftige Nachnutzung solcher dynamisch ergänzten Daten nicht gesichert ist.

Zusammengefasst lassen sich zwei Trends feststellen: zum einen gibt es eine *wachsende Verfügbarkeit wertvoller aber heterogener Datenquellen*, zum anderen einen *zunehmenden Bedarf an self-service und Ad-hoc-Integration*, welcher von neuen Daten-Nutzergruppen und -kontexten bestimmt wird. Während der erste Trend mehr und mehr zu einer Situation führt, in der Daten durch Integration externer Quellen angereichert werden können, sorgt er auch für eine höhere Komplexität. Der zweite Trend dagegen macht es notwendig, dass die Werkzeuge und Prozesse, die bei der Integration zum Einsatz kommen, möglichst einfach sein sollten, um den Ansprüchen eines breiteren Nutzerspektrums gerecht zu werden.

## 2 Datenintegration zur Anfragezeit

Diese Arbeit hat sich zum Ziel gesetzt, die beiden oben beschriebenen Trends zu adressieren und damit die verfügbare Datenvielfalt und -menge für die Analysen der Anwender zu erschließen.

Abbildung 1a zeigt eine Übersicht über einen manuellen Prozess zur Beantwortung einer Ad-hoc-Datenbankanfrage, die auf Daten von noch unbekannten externen Quellen angewiesen ist. Bevor der Nutzer die eigentliche Anfrage stellen kann, muss er zunächst relevante Datenquellen identifizieren, zum Beispiel durch eine normale Suchmaschine. Im Folgenden müssen die Daten extrahiert und bereinigt werden, also in eine Form gebracht werden, die in einem normalen Datenbanksystem verwendbar




(a) Regular Ahead-of-time integration

(b) Top-k Query-time Data Integration

Abb. 1: Alternative Vorgehensweisen für Ad-hoc-Anfrage über externe Daten

ist. In einem nächsten Schritt müssen die Daten mit der vorhandenen Datenbasis integriert werden, was die Abbildung von einander entsprechenden Konzepten, aber auch Instanztransformationen beinhaltet. Erst wenn dieser Prozess abgeschlossen ist, kann die ursprüngliche Anfrage gestellt werden. Falls das Ergebnis nicht den Anforderungen des Nutzers entspricht oder der Nutzer versuchsweise eine andere Datenquelle nutzen möchte, muss der Prozess wiederholt werden.

In dieser Arbeit wird das Konzept der *Datenintegration zur Anfragezeit* als alternatives Konzept eingeführt. Dabei geht es darum, dem Nutzer zu ermöglichen, Anfragen an eine Datenbank zu stellen, und dabei beliebige, noch nicht definierte Attribute zu referenzieren, für die dann automatisch zur Anfragezeit Quellen gesucht und integriert werden. Dabei soll es nicht erforderlich sein, bestimmte Quellen oder Abbildungen explizit anzugeben. In dieser Arbeit werden solche Anfragen als *Open World Anfragen* bezeichnet, da sie über Daten definiert sind, die (noch) nicht in der Datenbank enthalten sind. Das Ziel dabei ist es, dem Nutzer die Spezifikation von Informationsbedürfnissen auf deklarativem Wege zu ermöglichen, und zwar so als ob dabei nur lokale Daten zum Einsatz kommen würden. Das Datenbanksystem automatisiert dann den Prozess der Datensuche und -integration.

Allerdings ist ein vollständig automatischer Such- und Integrationsprozess nicht vorstellbar. Die zugrundeliegenden Methoden aus den Bereichen Information Retrieval und Automatischem Matching arbeiten selbst mit Top-k Ergebnissen oder liefern unsichere Antworten mit Konfidenzwerten. Um mit diesen Faktoren umzugehen, überträgt diese Arbeit das Konzept der Top-k Suchergebnisse, wie es aus Suchmaschinen bekannt ist, auf strukturierte Datenbankanfrageergebnisse. Unter diesem Paradigma antwortet das System auf eine Open World Anfrage nicht mit einem einzelnen, eindeutigen Ergebnis, wie es bei einer normalen Datenbankanfage stets der Fall wäre. Stattdessen produziert es eine geordnete Liste von möglichen Antworten, von denen jede auf anderen Datenquellen oder anderen Anfrageinterpretationen basiert. Der Nutzer kann dann die Antwort wählen, die dem vorhandenen Informationsbedürfnis am ehesten entspricht. Dieser alternative Prozess ist in Abbildung 1b dargestellt.

Die Architektur, die das Prinzip der *Datenintegration zur Anfragezeit* realisiert, nähert sich dem Problem der Ad-hoc-Integration auf drei verschiedenen Wegen, die in Abbildung 2 dargestellt sind. Dabei wird eine große Sammlung externer Daten-

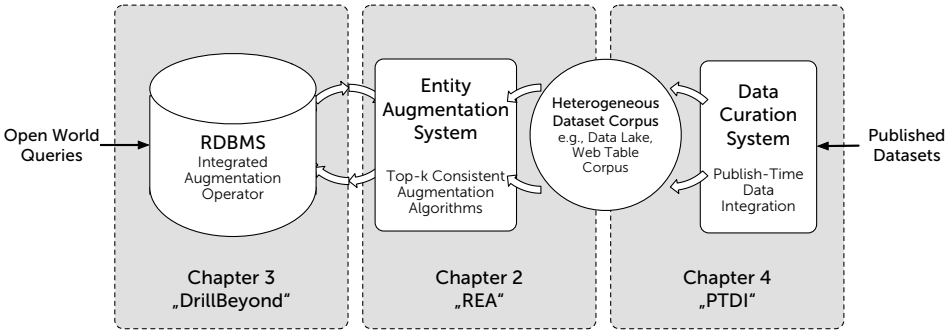



Abb. 2: Query-time data integration architecture and thesis structure

quellen angenommen, also eine Menge von potentiell nützlichen, aber unabhängigen und heterogenen Datensätzen, die zur Erfüllung von ad-hoc Informationsbedürnissen eingesetzt werden könnten. Dabei könnte es sich konkret beispielsweise um einen Korpus von Webtabellen, also datentragenden Tabellen aus dem offen Web handeln, oder aber um die Datensätze, die auf einer Open Data Plattform oder einem kommerziellen Data Lake [MBC15]verfügbar sind. Um aus einer solchen lose gekoppelten Menge von Datensets einen Nutzen ziehen zu können, werden in dieser Arbeit drei komplementäre System vorgeschlagen: Das Erste, genannt *REA*, ermöglicht *Top-k Entity Augmentation*, eine Grundoperation der Ad-hoc-Integration, bei der mehrere alternative Vorschläge zur Erweiterung ("Augmentierung") eines bestehenden Datensatzes mit neuen Attributen erzeugt werden. Das zweite System, genannt *DrillBeyond*, ist ein erweitertes RDBMS, das Open World SQL-Anfragen verarbeiten kann, also Anfragen die Attribute referenzieren, die über das definierte Schema hinausgehen. Die dritte Komponente ist ein *Datenkurationssystem*, das darauf zielt, die individuelle Wiederverwendbarkeit der Daten im Korpus zu erhöhen, ohne ein zentrales Schema zu fordern.

Die neuartigen Methoden und Algorithmen, die in diesen drei Systemen vorgestellt werden, bilden den Kern dieser Arbeit und werden in drei Hauptkapiteln diskutiert, die in den folgenden drei Abschnitten angerissen werden sollen.

## 3 Top-k Entity Augmentierung

Im ersten Hauptkapitel werden neue Algorithmen vorgestellt, die multiple, konsistente, aber komplementäre Integrationsvorschläge für eine Vielzahl potentieller Datenquellen erzeugen. Dabei wird das *Entity Augmentation* Problem diskutiert, welches auf die Erweiterung einer gegebenen Menge von Entititäten durch ein zusätzliches, nutzerbestimmtes Attribut abzielt, welches für diese noch nicht definiert ist. Dieses Attribut wird dann materialisiert, indem relevante Datenquellen gesucht und integriert werden. In Abbildung 3 ist ein Beispiel für ein solches Entity Augmentation Problem dargestellt. Dabei symbolisiert die obere Tabelle die Augmentations-Anfrage, die potentiellen Datenquellen sind darunter dargestellt. Die Anfragetabelle enthält eine Menge von fünf Konzernen, und das geforderte

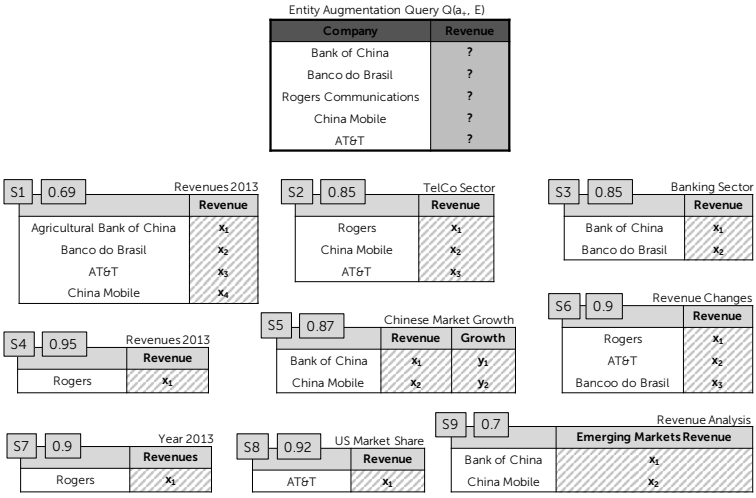



Abb. 3: Beispielhaftes Augmentierungs-Szenario: Anfrage- und Kandidatentabellen

Augmentationsattribute "*revenue*", also *Umsatz*. Die Kandidatequellen darunter variieren in ihrer Abdeckung der Anfragedomäne, dem genauen Attribut, das sie zur Verfügung stellen, und ihrem Kontext. Die Aufgabe ist also, das geforderte Attribut mit einer Menge von Werten aus den heterogenen und sich überlappenden Kandidaten zu ergänzen.

In den letzten Jahren ist eine Vielzahl verwandter Arbeiten publiziert worden, die solche Anfragen auf Basis großer Webtabellenkorpora beantworten, so etwa *InfoGather* [Ya12] oder das *WWT* System [PS12]. Allerdings beantworten alle vorgeschlagenen Methoden solche Anfragen mit einer einzelnen Antwort, bei der die Werte auf Basis der einzelnen Entititäten bestimmt werden, wobei wenig auf Konsistenz und Anzahl der dazu verwendeten Datenquellen geachtet wird. In dieser Arbeit werden Schwächen dieser Ansätze im Bezug auf Probleme wie *Attributvariationen*, *unklare Nutzerabsichten*, *Vertrauenswürdigkeit*, *Explorative Suche* und *Fehlertoleranz* aufgezeigt.

Um diese Schwächen zu vermeiden, zielt der in dieser Arbeit vorgeschlagenen Ansatz darauf ab, *konsistente* wie auch *relevante* Ergebnisse aus einer *minimalen* Menge von Quellen zu erzeugen und dabei statt einer Antwort *diversifizierte Top-k* Antworten zu geben. Dazu stellt diese Arbeit eine erweiterte Form des bekannten Set Cover Problems vor, das *Top-k konsistente Set Cover*-Problem, auf das die obigen Anforderungen abgebildet werden. Es wird ein Kern-Framework zur Lösung des Top-K konsistenten Set Cover-Problems vorgestellt, das es erlaubt, die vier erwähnten Problemdimensionen gemeinsam zu optimieren. Als Basisverfahren wird ein einfacher Greedy-Algorithmus, der nach den entsprechenden Ansätzen für das klassische Set Cover-Problem modelliert ist, vorgestellt. Dieser wählt konsistente, d.h. möglichst ähnliche Quellen aus, um die einzelnen Lösungen zu erzeugen, während er

die Diversität zwischen den in mehreren Iterationen erzeugten Lösungen maximiert, um komplimentäre Alternativlösungen zu erzeugen. Im folgenden wird der einfache Greedy-Algorithmus noch mehrmals erweitert: zunächst um einen Ansatz, der den durchsuchten Teil des Suchraums vergrößert und dafür eine Selektionsphase einführt, um die enstandenen Mehr-Lösungen zu filtern. Schließlich wird das Problem noch auf evolutionäre Algorithmen abgebildet, da diese durch das Konzept der Population eine besonders effektive Abbildung des Problems der Erzeugung von diversifizierten Top-k Lösungen ermöglichen.

Um die Vorteile des *Top-k konsistenten Set Cover*-Ansatzes zu illustrieren, kann man das erwähnte Problem der *Attributvariationen* am Beispiel in Abbildung 3 heranziehen. Denn selbst bei der scheinbar einfache Anfrage nach *"revenue/Umsatz"*, handelt es sich real um ein komplexes Konzept, mit vielen Varianten wie "US Umsatz" oder "Umsatz in Schwellenländern", mit verschiedenen Gültigkeitszeiträumen, unterschiedlichen Währungen und auch mit abgeleiteten Attributen wie "Umsatzwachstum". Während die meisten verwandten Arbeiten annehmen, dass man eine eindeutige Wahrheit in einem einzelnen Anfrageergebnis abbilden kann, ergeben sich aus dem in dieser Arbeit vorschlagenen Ansatz stets mehrere, alternative Lösungen. Letzlich löst das Top-k Paradigma viele Schwächen, die aus der Unsicherheit und Mehrdeutigkeit von Webdaten und Nutzeranfragen resultieren, indem der Nutzer bei der Auswahl der richtigen Lösung in den Prozess einbezogen wird, analog zu dem Vorgehen einer klassischen Suchmaschine.

Die neuen Algorithmen sind in einem öffentlich verfügbaren neu-entwickelten Webtabellen Such- und Integrationssystem namens REA implementiert. Das System wurde auf Basis des DWTC evaluiert, eines im Zuge der vorliegenden Arbeit entstandenen Korpus' von 125 Millionen aus dem Web extrahierten Datentabellen, welcher der wissenschaftlichen Community als Datensatz zugänglich gemacht wurde. In dieser Evaluation wurde der Effekt der vorgeschlagenen Algorithmen auf die Basismaße Präzision, Abdeckung und Laufzeit, aber auch auf die neu-identifizierten Problemdimensionen Konsistenz, Minimaliät und Diversität der Ergebnisliste hin untersucht. Die Experimente haben gezeigt, dass besonders der genetische Set Covering Ansatz die Konsistenz und Minimalität der Ergebnisse signifikant verbessert, ohne dabei Kompromisse bei Präzision oder Abdeckung machen zu müssen.

## 4 Open-world SQL Queries

Bisher wurde der Ansatz der Entity Augmentation lediglich isoliert, bezogen auf eine Tabelle betrachtet und in einem abgeschlossenen Entity-Augmentation-System implementiert. Jedoch kann angenommen werden, dass die Ad-hoc-Integration von Daten, vor allem komplexer Analyseszenarien, gewinnbringend eingesetzt werden kann, in denen die Ausführung der Augmentation-Anfrage nur ein Baustein in einer ganzen Reihe analytischer Operationen darstellt. Der Nutzer eines solchen Systems ist dann nicht an dem reinen Ergebnis der Augmentation-Anfrage interessiert, sondern stattdessen an einem höherwertigen Anfrageergebnis, das nur zu

```
select
    n_name,
    avg(o_totalprice)
from nation, customer, orders
where
    n_nationkey=c_nationkey
    and c_custkey=o_custkey
group by n_name
```

(a) Initiale Anfrage

```
select
    nation. creditRating ,
    avg(o_totalprice)
from nation, customer, orders
where
    n_nationkey=c_nationkey
    and c_custkey=o_custkey
    and nation. gdp > 10.0
group by nation. creditRating
```

(b) Open World Anfrage

Abb. 4: Beispielhafte Open World SQL-Anfrage

einem gewissen Teil aus externen Attibuten besteht. Der größere Teil der Daten wird weiterhin aus klassischen Datenbanksystemen stammen. Dies soll im folgenden an einem Beispiel illustriert werden: hier sei eine typische Data-Warehouseanfrage angenommen (siehe Abbildung 4a), welche die Verkäufe, gruppiert nach einzelnen Ländern, analysiert. In einem nächsten Schritt möchte der Nutzer den aggregierten Verkaufsdaten externe Kontextinformationen, wie etwa das Bruttoinlandsprodukt (GDP) oder die Bonität der Länder (credit rating), hinzufügen. Da diese nicht im lokalen Datenbankschema repräsentiert sind, muss dafür auf externe Datenquellen zurückgegriffen und die darin enthaltenen Daten müssen über viele Verarbeitungsschritte in das Datenbanksystem überführt werden. Idealerweise wäre der Nutzer in der Lage, dieses situative Informationsbedürfnis deklarativ, durch eine einfache Erweiterung der Ursprungsanfrage, auszudrücken. Dies ist in der Anfrage in Abbildung 4b dargestellt, in der die offenen Attribute emphgdp und *credit* einfach in die SQL-Anweisung aufgenommen wurden und in verschiedenen Klauseln direkt adressiert werden.

In dieser Dissertation wurde ein hybrides Datenbank/IR-System namens *Drill-Beyond* vorgestellt, das eben solche Anfragen, die zum einem über den regulären Inhalt der Datenbank zum anderen auch über offene Attribute definiert sind, beantworten kann. Dazu wurde das bereits vorgestellte Entity-Augmentation-System in einem neuen Datenbankoperator $\omega$ eingekapselt und so eine enge Verzahnung mit einem bestehenden Datenbanksystem gewährleistet. Des Weiteren wurde die optimale Platzierung des neuen Operators in einem Anfrageplan unter dem Aspekt der Anfragelaufzeit als auch der Ergebnisqualität diskutiert. Dafür wurde ein Kostenmodell sowie eine ganze Reihe von Optimierungsregeln eingeführt. Diese werden entweder zur Planungszeit einer Anfrage oder zur Ausführungszeit angewandt. Hier galt es insbesondere zu berücksichtigen, wie hinsichtlich der möglichen Ergebnisalternativen die invarianten Teile eines Anfrageplans maximiert werden können. Dadurch können Pläne erzeugt werden, die bei einer einmaligen Ausführung langsamer sind aber bei einer durch den Top-k-Ansatz bedingten Mehrfachausführung stärker von Zwischenmaterialisierung profitieren. Weiterhin wurde untersucht, wie der ursprüngliche Prozess der Entity-Augmentation durch Kontextinformation, die in den SQL-Anfragen kodiert sind, verbessert werden kann. Hier seien beispielhaft die über Typinferenz ermittelten Datentypen offener Attribute oder Prädikate genannt.

Schließlich wurde der Operator und die entwickelten Optimierungen praktisch in PostgresQL implementiert, was die aussagekräftige Evaluierung der Konzepte auf standardisierten Testdatenbanken und vollwertigen SQL-Anfragen erlaubt. In dieser Evaluation konnte die Effektivität der vorgestellten Optimierungen zur Reduktion des Laufzeit-Overheads von Open World Anfragen nachgewiesen werden. Desweiteren wurde gezeigt, dass der Einsatz von Kontextwissen aus den SQL-Anfragen im zugrundeliegenden Augmentation-System die Qualität und Performanz der Augmentierung im Vergleich zum isolierten Betrieb verbessert werden kann.

Zusammenfassend lässt sich sagen, dass das DrillBeyond System mit seiner engen Integration von Augmentierung und relationaler Anfrageverarbeitung die Verarbeitung von Ad-hoc-Datensuche und -integrationsanfragen auch in praktischen Kontexten ermöglicht.

## 5 Publish-time Data Integration

Die in den vorherigen Abschnitten beschriebenen Erweiterungstechniken basieren alle auf großen Dokumentsammlungen bestehend aus vielen, heterogenen Datensätzen. Im letzen Kapitel dieser Dissertation werden die Herausforderungen bei der Verwaltung solcher Dokumentenkorpora beschrieben und darauf aufbauend ein *Datenkurationssystem* (engl. *data curation systems* [St13]) als Komplement zu klassischen Datenbank- und Data-Warehouse-Systemen eingeführt. Im ersten Schritt werden Beispielausprägungen solcher Systeme, wie etwa Open-Data-Plattformen, wissenschaftliche Datenrepositorien und Data Lakes diskutiert. All diesen Systemen ist gemein, dass sie die Daten in ihrem Urspungsformat speichern, so wie sie entweder vom Quellsystem oder von einem Menschen erzeugt wurden und keinem zentralen Schema folgen. Die zugrundeliegende Motivation dafür besteht darin, zunächst alle Daten zu speichern, auch wenn noch nicht bekannt ist, ob und wie diese Daten in Zukunft genutzt werden.

Um die Probleme zu definieren, die bei dieser neuen Form der Datenverwaltung entstehen können, wurde eine große Zahl von Open-Data-Plattformen untersucht. Ein zentrales Ergebnis dieser Studie war die Erkenntnis, dass obwohl diese Plattformen viele nützlichen Daten bereitstellen, sie die Nachnutzbarkeit der Daten nur sehr begrenzt unterstützten. Insbesondere das automatisierte Finden und die Ad-hoc-Integration offener Daten wird durch das Fehlen einheitlicher Metadaten sowie Standards erschwert. Da eine vollständige Integration, wie sie üblicherweise angestrebt ist, nicht von Open-Data-Plattformen geleistet werden kann, fällt der gesamte Aufwand der Datensuche, -bereinigung und -integration den potentiellen zukünftigen Nutzer der Daten zu.

Damit sind *Datenkurationssystem* in ihrem Wesen eng mit dem Konzept des sogenannten Dataspace [FHM05] verwandt, die im Sinne eines “Pay-as-you-go”-Ansatzes [Ma07] die Datenintegration erst leisten, wenn klar ist, wie die Daten genutzt werden sollen. Nichtsdestotrotz, sollten einige wesentliche Integrationsaufgaben bereits a

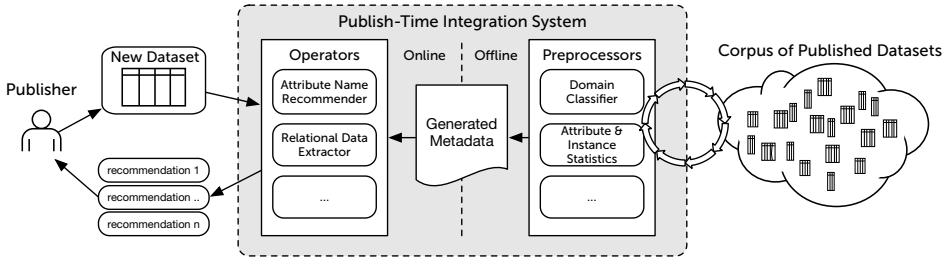



Abb. 5: Architektur einer PTDI-Plattform

priori geleistet werden, um damit den Nutzen dieser Plattformen als solches zu erhöhen. Aus diesem Grund wird in dieser Dissertation das neue Paradigma der *Publish-time Data Integration*, kurz PTDI, eingeführt. In Kern bedeutet dies, dass beim Veröffentlichen neuer Daten die Nutzer Integrationsempfehlungen erhalten, wie sie die Nachnutzbarkarkeit des Datensatzes maximieren können. Konkret wird dazu das Konzept der *PTDI-Operatoren* eingeführt (siehe die Architekturübersicht in Abbildung 5). Diese werden bei der Veröffentlichung neuer Daten angestoßen und erzeugen operatorspezifische Empfehlungen. Zur Unterstützung des Laufzeitsystems bei der Erzeugung der Vorschläge werden Metadaten und Statistiken von einem *PTDI-Präprozessor* gesammelt. Als konkrete Instanzierungen werden in der Dissertation zwei PTDI-Operatoren vorgeschlagen: Zum einen die *Erzeugung alternativer Attributnamen* und zum anderen das *Extrahieren relationaler Daten aus partiell strukturierten Dokumenten* wie zum Beispiel Tabellenkalkulations- bzw. HTML-Dateien.

Der erste Operator ermöglicht es, Empfehlungen alternativer Attibutnamen zu erzeugen, die besser zu den Schemata der bereits bestehenden Datensätze passen. Damit soll, trotz Fehlens eines globalen Schemas, die Heterogenität bei den Spaltenbezeichnern begrenzt werden. Die Empfehlungen werden basierend auf dem Grad der Schnittmenge zwischen den Ausprägungsmengen sowie unter Verwendung von Klassifikationsansätzen erzeugt. Für letzteren Ansatz werden die bestehenden Datensätze segmentiert so dass ähnliche Datensätze einen Cluster bilden. Zur Bewertung des Ansatzes zur Erzeugung alternativer Attributnamen wurde das Laufzeitverhalten und die Anzahl der generierten Empfehlungen sowie ihre Genauigkeit untersucht. Der zweite PTDI-Operator mit dem Namen *DeExcelerator* ermöglicht es, die Transformation partiell strukturierter Dokumente, d.h. Dokumente in denen strukturierte Daten mit beliebigen Textdaten und Layoutinformation vermischt werden, in Relationen erster Normalform zu überführen. Dazu wurde zunächst eine große Zahl von Tabellenkalkulationsdateien verschiedener Open-Data-Plattformen betrachtet und eine Menge typischer Denormalisierungen abgeleitet. Mit Denormalisierungen sind hier Muster gemeint, die eine Nachnutzung der Daten verhindern. Darauf aufbauend wird eine Operatorkette definiert, mit der sukzessiv Denormalisierungen entfernt und die Tabellenkalkulationsdateien transformiert werden. Die Relevanz der einzelnen Denormalisierungen und die Korrektheit der entsprechenden Transformationen wurde anhand einer repräsentativen Menge echter

Tabellenkalkulationsdateien evaluiert. Anhand der Experimente kann das Fazit gezogen werden, dass die vorgestellten Ansätze die Nachnutzbarkeit heterogener Dokumentsammlungen wesentlich erhöhen, ohne das dadurch die Einfachheit des Datenpublikationsprozesses leidet.

## Literaturverzeichnis

[Co09] Cohen, Jeffrey; Dolan, Brian; Dunlap, Mark; Hellerstein, Joseph M.; Welton, Caleb: MAD skills: new analysis practices for big data. VLDB, 2:1481–1492, August 2009.

[Eb15] Eberius, Julian: Query-Time Data Integration. `http://nbn-resolving.de/urn:nbn:de:bsz:14-qucosa-191560`, 2015.

[FHM05] Franklin, Michael; Halevy, Alon; Maier, David: From databases to dataspaces: a new abstraction for information management. SIGMOD Rec., 2005.

[La01] Laney, Doug: 3D data management: Controlling data volume, velocity and variety. META Group Research Note, 6:70, 2001.

[LJ12] Labrinidis, Alexandros; Jagadish, H. V.: Challenges and Opportunities with Big Data. Proc. VLDB Endow., 5(12):2032–2033, August 2012.

[Ma07] Madhavan, Jayant; Jeffery, Shawn R.; Cohen, Shirley; Dong, Xin Luna; Ko, David; Yu, Cong; Halevy, Alon: Web-scale Data Integration: You Can Only Afford to Pay As You Go. In: CIDR. 2007.

[MB12] McAfee, Andrew; Brynjolfsson, Erik: Big data: the management revolution. Harvard business review, 90(10):60—6, 68, 128, October 2012.

[MBC15] Mohanty, Hrushikesha; Bhuyan, Prachet; Chenthati, Deepak: Big Data: A Primer. Springer India, 2015.

[PS12] Pimplikar, Rakesh; Sarawagi, Sunita: Answering Table Queries on the Web using Column Keywords. In: VLDB. 2012.

[St13] Stonebraker, Michael; Beskales, George; Pagan, Alexander; Bruckner, Daniel; Cherniack, Mitch; Xu, Shan; Analytics, Verisk; Ilyas, Ihab F.; Zdonik, Stan: Data Curation at Scale: The Data Tamer System. In: CIDR. 2013.

[Ya12] Yakout, Mohamed; Ganjam, Kris; Chakrabarti, Kaushik; Chaudhuri, Surajit: InfoGather: entity augmentation and attribute discovery by holistic matching with web tables. In: SIGMOD. S. 97–108, 2012.

**Julian Eberius** studierte von 2005 bis 2010 Medieninformatik an der TU Dresden. Für seine Diplomarbeit zu selbstkonfigurierenden Schema Matching Systemen erhielt er den AMD Preis für die beste Diplomarbeit der Fakultät Informatik der Technischen Universität Dresden. Bis 2015 war er wissenschaftlicher Mitarbeiter an der Professur Datenbanken der TU Dresden und verteidigte im Dezember 2015 seine Dissertation zum Thema "Query-Time Data Integration". Derzeit arbeitet er bei einem TU Startup im Bereich Mobile Analytics und Big Data.