Datenqualität in Sensordatenströmen

Dissertation

zur Erlangung des akademischen Grades
Doktoringenieur (Dr.-Ing.)

vorgelegt an der
Technischen Universität Dresden
Fakultät Informatik

eingereicht von

Dipl.-Inf. Anja Klein
geboren am 19. Juni 1981
in Wolfen

Gutachter:

Prof. Dr.-Ing. Wolfgang Lehner
Technische Universität Dresden, Fakultät Informatik
Institut für Systemarchitektur
Lehrstuhl für Datenbanken
01062 Dresden

Prof. Dr. Peter Chamoni
Universität Duisburg-Essen
Mercator School of Management
Lehrstuhl für Wirtschaftsinformatik und Operations Research
47057 Duisburg

Tag der Verteidigung: 19. Juni 2009

Kurzfassung

Die stetige Entwicklung intelligenter Sensorsysteme erlaubt die Automatisierung und Verbesserung komplexer Prozess- und Geschäftsentscheidungen in vielfältigen Anwendungsszenarien. Sensoren können zum Beispiel zur Bestimmung optimaler Wartungstermine oder zur Steuerung von Produktionslinien genutzt werden. Ein grundlegendes Problem bereitet dabei die Sensordatenqualität, die durch Umwelteinflüsse und Sensorausfälle beschränkt wird.

Ziel der vorliegenden Arbeit ist die Entwicklung eines Datenqualitätsmodells, das Anwendungen und Datenkonsumenten Qualitätsinformationen für eine umfassende Bewertung unsicherer Sensordaten zur Verfügung stellt. Neben Datenstrukturen zur effizienten Datenqualitätsverwaltung in Datenströmen und Datenbanken wird eine umfassende Datenqualitätsalgebra zur Berechnung der Qualität von Datenverarbeitungsergebnissen vorgestellt. Darüber hinaus werden Methoden zur Datenqualitätsverbesserung entwickelt, die speziell auf die Anforderungen der Sensordatenverarbeitung angepasst sind. Die Arbeit wird durch Ansätze zur nutzerfreundlichen Datenqualitätsanfrage und -visualisierung vervollständigt.

Danksagung

Die vorliegende Dissertation entstand während meiner Tätigkeit bei SAP Research CEC Dresden. Eine ganze Reihe Menschen haben auf unterschiedliche Weise zum Gelingen der Arbeit beigetragen. Denen, die mich auf dem Weg ganz oder teilweise begleitet haben, möchte ich hiermit danken.

Meinem Doktorvater Prof. Dr.-Ing. Wolfgang Lehner gilt ein ganz besonderer Dank für seine Unterstützung und Begleitung sowie die Möglichkeit, meine Promotion an seinem Lehrstuhl durchzuführen. Er hat durch zahlreiche Anregungen und konstruktive Diskussionen wesentlich zur Entstehung dieser Arbeit beigetragen. Mein Dank gilt weiterhin Herrn Prof. Dr. rer. nat. habil. Dr. h. c. Alexander Schill und Herrn Prof. Dr. Peter Chamoni für ihre hilfreichen Hinweise sowie die Bereitschaft, das Koreferat zu übernehmen.

Während der Arbeit an dieser Dissertation habe ich viel Unterstützung von meinen Kollegen bei SAP Research und der TU Dresden erhalten. Dafür bin ich besonders Dr. Hong-Hai Do, der mich vor allem in der Anfangsphase durch fachliche und methodische Hinweise unterstützt hat, und Dr. Gregor Hackenbroich für seine stetige Motivation und Hilfe in mathematischen Fragen zu Dank verpflichtet.

Darüber hinaus danke ich Jürgen Anke, Kay Kadner und Jens Ebert für die fachliche und orthographische Durchsicht der Arbeit. Mit ihrer Hilfe ist es gelungen, Fehler zu beseitigen und die Verständlichkeit zu verbessern. Weiterhin danke ich Jan Frömberg, Katja Seidler und Patrick Mathey für ihre Unterstützung bei der prototypischen Realisierung der entwickelten Konzepte.

Herzlich danken möchte ich auch meiner Familie, meinen Freunden und Uwe, die mich während der Bearbeitung stets unterstützt und motiviert haben und besonders in der Endphase der Arbeit viel Verständnis für meinen Zeitmangel aufbrachten.

Dresden, im April 2009

Anja Klein

Inhaltsverzeichnis

1 Einführung

Das zentrale Thema dieser Arbeit ist das Management von Datenqualitätsinformationen zur Bewertung von Sensordaten in Smart-Item-Anwendungen. Sensoren überwachen Eigenschaften von Smart-Items und deren Umgebung, um Prozesse zu steuern und Entscheidungen zu unterstützen. Dafür werden Messdaten an den Smart-Items aufgezeichnet und in einem Datenstrom-Management-System verarbeitet. Die Qualität der Sensormessdaten sowie der abgeleiteten Informationen wird jedoch durch sensoreigene sowie externe Fehlerquellen beschränkt. Um Prozess- und Entscheidungsfehler auf Basis fehlerbehafteter Informationen zu verhindern, muss die Qualität der Sensordaten kontrolliert werden. Es müssen Möglichkeiten geschaffen werden, alle Aspekte der Sensordatenqualität von Smart-Items aufzuzeichnen, zu verarbeiten und dem Datenkonsumenten zur Bewertung der berechneten Informationen zur Verfügung zu stellen. Des Weiteren müssen Methoden zur Verbesserung der Datenqualität (DQ) entwickelt werden, falls die gegebene Datenqualität den Anforderungen der Anwendung oder des Nutzers nicht genügt. Die Spezifikation der Datenqualität hängt ebenfalls von den Anforderungen der Anwendung ab. Sie wird durch die Gesamtheit der Qualitätsaspekte, die die Nutzbarkeit der Daten zur Verfolgung eines bestimmten Zieles beschreiben, definiert.

Im Folgenden werden Smart-Items-Umgebungen basierend auf physischen Sensoren als Hintergrund der vorliegenden Dissertation vorgestellt. Die Aufgaben bei der Datenverwaltung und -verarbeitung werden beschrieben, um die Problemstellung abzuleiten und die vorliegende Arbeit zu motivieren. Nachfolgend werden die Ziele der Arbeit erläutert und der Lösungsansatz illustriert. Anschließend wird der Aufbau dieser Arbeit vorgestellt.

1.1. Hintergrund

Smart-Items sind physische Produkte, die mit eingebetteten Rechnersystemen ausgestattet sind. Sie sind mit Sensoren, zum Beispiel GPS-, RFID- oder physikalischen Sensoren (Druck, Temperatur, etc.), verbunden und ermöglichen die automatische Erfassung von Daten wie zum Beispiel Eigenschaften eines Produktes, den Betriebszustand einer Maschine oder Informationen über deren Umwelt.

Sensormessdaten stellen Rohdaten dar, die mit Hilfe verschiedener Verarbeitungsschritte ausgewertet werden müssen. Die gewonnenen Informationen dienen zur Automatisierung und Optimierung von vielfältigen Produktions- und Geschäftsprozessen. Mittels Sensoren können zum Beispiel Maschinen gesteuert oder die Qualität von hergestellten Produkten überprüft werden. Des Weiteren können die gewonnenen Informationen zur Unterstützung von komplexen Geschäftsentscheidungen, wie die Planung neuer Produktionslinien, herangezogen werden.

Für die Datenverarbeitung werden die Sensoren der Smart-Items in Netzwerken verknüpft. Sie bestehen aus meist heterogenen Datenübertragungspunkten und Rechnerknoten, die eine Integrations-Middleware für den Datentransfer von den Sensorknoten zu Software-Applikationen oder Zieldatenbanken betreiben. In allen Komponenten dieser Sensor-Umgebung ist eine Verarbeitung der Daten möglich.

Seit Mitte der 90er Jahre werden Datenstrom-Management-Systeme (DSMS) zur Verarbeitung kontinuierlicher Datenströme entwickelt. Auch Sensormessdaten stellen kontinuierliche Datenströme dar, die mit Hilfe eines DSMS verknüpft und verarbeitet werden. Ein Sensornetzwerk kann somit als verteiltes Datenstromsystem aufgefasst werden. Die frühzeitige Datenverarbeitung im Datenstromsystem ist besonders dann sinnvoll, wenn nur geringe Hardware-Ressourcen für die lokale Speicherung und Datenübertragung zur Verfügung stehen, wie es in Sensornetzwerken und Smart-Item-Anwendungen häufig der Fall ist.

Die verarbeiteten Sensordaten werden entweder direkt zur Steuerung von Produktions- oder Geschäftsprozessen genutzt oder zur weiteren Analyse in einer Zieldatenbank abgelegt. Hier können Verfahren des Data-Mining angewendet werden, um zum Beispiel Produktionsprozesse mit Hilfe historischer Daten zu optimieren oder Geschäftsstrategien anzupassen. Sensordaten sind damit die Grundlage sowohl der kurzfristigen Kontrolle und Steuerung von technischen Geräten, z.B. in Produktionsanlagen, als auch zur Planung langfristiger Geschäftsabläufe.

1.2. Problemstellung und Motivation

Ein grundlegendes Problem bei der sensorbasierten Prozesssteuerung und Entscheidungsunterstützung bereitet die beschränkte Qualität der zugrunde liegenden Sensordaten. Es gibt sensorinhärente, physikalische Einschränkungen, da alle Messverfahren und

Sensorkonstruktionen nur eine beschränkte Präzision der Sensormessung erlauben. Des Weiteren wird die Datenqualität durch Messfehler oder Sensorausfälle herabgesetzt.

Es treten unterschiedliche Fehlertypen auf, die sich durch eine Menge verschiedener Datenqualitätsdimensionen abbilden lassen. So bestimmt die Güteklasse des Sensors dessen *Genauigkeit*, das heißt die maximale Abweichung des Messwertes vom tatsächlichen Wert der physikalischen Größe. Die Qualitätsdimension *Vollständigkeit* beschreibt den Anteil fehlender Messwerte im Sensordatenstrom.

Während der Datenverarbeitung werden die initialen Fehler mit hoher Wahrscheinlichkeit verstärkt. Die Sensordatenströme werden analysiert, kombiniert und aggregiert, um bisher unbekannte Informationen und Zusammenhänge zu erkennen. Entsprechend den ausgeführten Verarbeitungsschritten werden auch die Fehler kombiniert und aggregiert. Sie durchlaufen die gesamte Datenverarbeitung und haben zur Folge, dass auch die gewonnenen Informationen fehlerbehaftet sind. Außerdem ist in vielen Anwendungen das Bestimmen einer Datenstichprobe (engl. sampling) zum Lastausgleich oder zur Anpassung der Datenraten notwendig. Dabei werden Datensätze aus dem Datenstrom entfernt. Der Informationsverlust führt zu zusätzlichen Fehlern und vergrößert damit die Abweichungen des Verarbeitungsergebnisses.

Sind aber die Informationen, die auf Basis der Sensoren erarbeitet wurden, fehlerhaft, so setzen sich diese Fehler in den getroffenen Entscheidungen fort. Zum Beispiel könnten ungeeignete Verarbeitungsprozesse eine Produktionslinie nicht optimal auslasten. Fehler in der Qualitätsprüfung können u.a. dazu führen, dass minderwertige Produkte zum Kunden versandt bzw. zu viele akzeptable Waren als Ausschuss deklariert werden. Dienen die Sensordaten zur Unterstützung von Entscheidungen auf Managementebene, können falsche Strategien oder Aktionen zu erhöhten Kosten für das Unternehmen führen.

Das größte Problem besteht im mangelnden Bewusstsein der Datenkonsumenten für Datenqualitätsprobleme. In fast allen datenverarbeitenden Anwendungen werden die gegebenen Daten und Informationen als wahr und fehlerfrei angenommen. Nach einer Studie der Unternehmensberatung PricewaterhouseCoopers [Pri04] messen nur 45% der Unternehmen in Australien, den USA und Großbritannien die Qualität ihrer Daten, obwohl fehlerhafte Daten den Unternehmensumsatz um bis zu 12% reduzieren. Nur etwa ein Drittel dieser Firmen sind mit der Datenqualität in ihrem Unternehmen zufrieden. Bis zu 60% der Ausgaben eines Unternehmens resultieren aus Datenqualitätsproblemen und könnten durch Methoden zur Qualitätsmessung und -verbesserung vermieden werden [Pri04]. Diese Kosten umfassen laut [Dat02] allein für US-Unternehmen 600 Mio. $ pro Jahr. Sogar der Challenger-Absturz 1986 und der irrtümliche Abschuss eines iranischen Airbus 1988 werden zum Teil auf Datenqualitätsprobleme zurückgeführt [FK01].

Auch in sensorgestützten Anwendungen werden Datenqualitätsprobleme oft ignoriert. Obwohl den Entscheidungsträgern die eingeschränkte Qualität der Sensorrohdaten oft bewusst ist, wird nicht an den berechneten Informationen gezweifelt. Es werden

keine Schritte zur Qualitätskontrolle oder -verbesserung unternommen. Dabei stellte DeMarco bereits 1982 fest: „*You cannot control what you cannot measure*“. Er wies in [DeM82] auf die Notwendigkeit hin, die Qualität von Software-Projekten zu messen, um das Aufwand-Nutzen-Verhältnis von Software-Verbesserungen abschätzen zu können.

Ein Ziel der vorliegenden Arbeit ist es, das Bewusstsein für Datenqualitätsprobleme in Sensordatenströmen und abgeleiteten Informationen zu erhöhen. Es werden Methoden zum Messen und Verbessern der Sensordatenqualität präsentiert.

1.3. Ziel der Arbeit

Es gibt zwei Ansätze, der eingeschränkten Sensordatenqualität zu begegnen. Die optimistische Herangehensweise vertraut auf Sensoren mit hoher Präzision und postuliert auftretende Fehler als vernachlässigbar klein für den entsprechenden Anwendungskontext. Dieser Ansatz erfordert sehr präzise Sensoren und Sensorabschirmungen, um Messfehler durch äußere Einflüsse zu vermeiden. Außerdem müsste ein sehr großes Datenvolumen verlustfrei übertragen werden; dies wäre mit sehr hohen Kosten verbunden. Auch wenn diese Anforderungen umgesetzt werden, so können doch nicht alle Fehler mit absoluter Sicherheit beseitigt werden. In realen Anwendungsszenarien wird nie ein fehlerfreies Messen von Sensordaten möglich sein (siehe Abschnitt 2.2).

Das zentrale Thema dieser Arbeit steht orthogonal zur optimistischen Herangehensweise. Die Datenqualität wird als wichtiges Merkmal eines Messdatenstroms betrachtet und erlaubt damit die umfassende Bewertung des extrahierten Wissens. Informationen zur Datenqualität werden hierbei an den Sensoren der Smart-Items aufgenommen, analog zu den Messdaten verarbeitet und zur Software-Anwendung oder Zieldatenbank propagiert.

Sensoren erlauben die automatische Erfassung enormer Datenmengen. Die zusätzliche Verarbeitung und Übertragung von Datenqualitätsinformationen kann einen gewaltigen Mehraufwand bei Datentransfer und -verwaltung verursachen, wenn keine effizienten Verarbeitungswege gefunden werden. Datenqualität stellt außerdem eine neue Klasse an Metadaten des Sensordatenstroms dar, deren Verarbeitung in aktuellen Forschungsarbeiten und Werkzeugen nicht adressiert wird. Damit keine Informationen verloren gehen, müssen die Schritte der Datenverarbeitungskette in der Datenqualitätsverarbeitung abgebildet werden.

Aus diesen Überlegungen ergeben sich die folgenden Forschungsfragen.

1. Welche Konzepte und Datenstrukturen müssen entwickelt werden, um die effiziente Verwaltung umfassender Qualitätsinformationen zur Beschreibung von Sensordaten in statischen und dynamischen Daten-Management-Umgebungen zur Verfügung zu stellen?

Es existiert noch kein Ansatz zur ressourcensparenden Propagierung und Speicherung von Metainformationen zur Beschreibung der Qualität von Sensormessdaten. In dieser Arbeit wird ein Konzept zum Management von Datenqualitätsinformationen erarbeitet, das sowohl die Charakteristiken eines flüchtigen Datenstromsystems als auch die Anforderungen der dauerhaften Speicherung in einer Zieldatenbank berücksichtigt.

2. Welche Transformationsregeln werden benötigt, um Datenverarbeitungsschritte auf die Qualitätsinformationen abzubilden?

Zur Beantwortung dieser Frage werden in der Literatur vorgestellte Datenqualitätsalgebren analysiert und zusammengeführt. Typische Datenstromoperatoren werden bezüglich ihres Einflusses auf verschiedene Datenqualitätsdimensionen untersucht. Bei der Verarbeitung von Sensordaten spielen auch Operatoren aus der Signalverarbeitung und numerischen Algebra eine wichtige Rolle. Für diese Operatoren wird ebenfalls eine Datenqualitätsalgebra entwickelt.

3. Welche Strategien sind erforderlich, um Nutzeranforderungen an die Datenqualität in den Pfad der Datenverarbeitung zu integrieren und die Datenqualität zu erhöhen?

Sollte die berechnete Datenqualität nicht den Anforderungen der Anwendung oder des Datenkonsumenten genügen, müssen Methoden zur Qualitätsverbesserung zur Verfügung stehen. Ziel ist es, nutzerdefinierte Qualitätsanforderungen zu integrieren und somit die gewünschte Qualität zu garantieren.

4. Wie können dem Nutzer die Ergebnisse der Datenqualitätsmessung und -verarbeitung dargestellt werden, um eine verbesserte Evaluierung der Sensordaten und abgeleiteter Informationen zu ermöglichen und das Bewusstsein für Datenqualitätsprobleme zu erhöhen?

Zur Beantwortung dieser Frage werden Visualisierungsmethoden zur graphischen Darstellung der Datenqualität von Datenstrom- und Datenbankanfrageergebnissen entworfen und zusammen mit den Konzepten und Methoden zur Datenqualitätsverwaltung in einer prototypischen Entwicklung mit nutzerfreundlicher Bedienoberfläche umgesetzt.

1.4. Lösungsansatz

Das in dieser Arbeit vorgeschlagene Lösungskonzept für das Datenqualitätsmanagement in Sensordaten ist in Abbildung 1.1 illustriert, die den Datenfluss zwischen Sensoren als Datenquellen und den Software-Anwendungen zur Prozesskontrolle oder Entscheidungsunterstützung zeigt. Im ersten Schritt der Datenqualitätsakquisition werden die Dimensionen Genauigkeit, Konfidenz, Vollständigkeit, Datenmenge und Aktualität entsprechend der Definition der Sensordatenqualität in Abschnitt 2.4 mit Hilfe der Eigenschaften der Sensoren initialisiert. Nachfolgend werden die Mess- und Qualitätsdaten in Datenströmen verarbeitet und zur Zieldatenbank oder Anwendung transferiert.

Dabei wird das Metamodell des Datenstroms erweitert, um Datenqualität zu modellieren. Des Weiteren werden Datenqualitätsoperatoren benötigt, um die Datenverarbeitung auf den Qualitätsinformationen nachzuvollziehen. Zur Speicherung von Datenqualität in persistenten Datenbanken muss das relationale Datenbankschema erweitert werden. Außerdem muss auf einen effizienten Import der Datenqualitätsinformationen vom Datenstrom in die Datenbank geachtet werden.

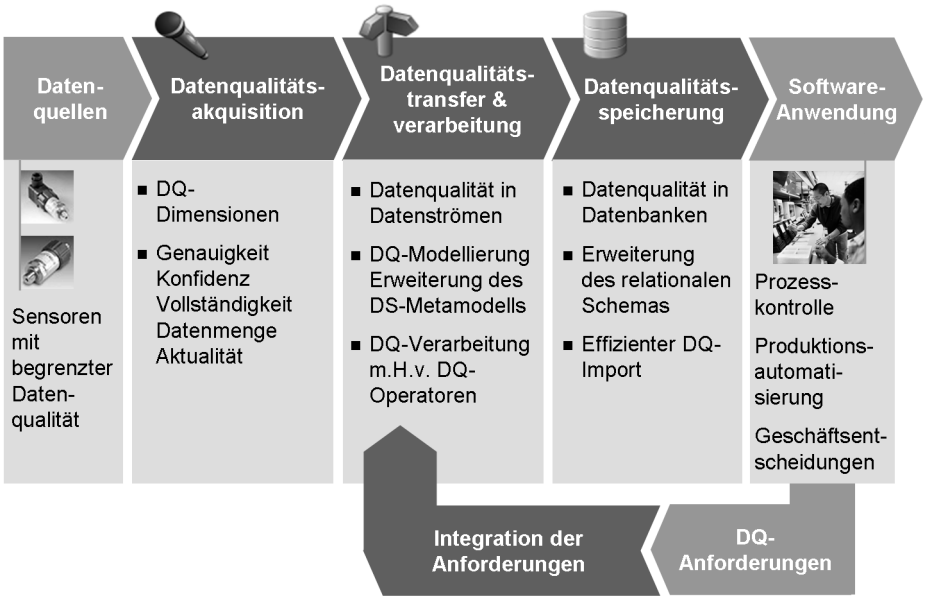


Abbildung 1.1.: Systemübersicht

Sensorgestützte Anwendungen können nun entweder kontinuierliche Messdaten des Datenstroms oder statische Daten der Datenbank nutzen. Sollten diese Daten nicht den Qualitätsanforderungen der Anwendung oder des Nutzers entsprechen, muss die Datenqualität verbessert werden. Dazu werden Datenqualitätsanforderungen in die Datenstromverarbeitung oder die Konfiguration der zugrunde liegenden Sensoren eingearbeitet.

In dieser Arbeit werden zum Beantworten der oben gestellten Forschungsfragen die folgenden Konzepte erarbeitet.

Effiziente Datenqualitätsmodellierung Ein neuer Ansatz zur effizienten Verwaltung von Datenqualitätsinformationen in Datenstrom- sowie Datenbank-Management-Systemen wird vorgestellt. Datenqualitätsfenster fassen Qualitätsinformationen der enthaltenen Messwerte zusammen, um das zusätzliche Datenvolumen sowohl in kontinuierlichen Datenströmen als auch in persistenten Datenbankrelationen zu reduzieren.

Methoden zur Datenqualitätsverarbeitung Um die Verarbeitung der Sensordaten auf die Qualitätsinformationen zu übertragen, wird ein Rahmenwerk zur Daten-

qualitätsverarbeitung entwickelt, das sowohl in Datenstromsystemen als auch in Datenbanken eingesetzt werden kann. Dafür wird ein umfassender Satz an Datenstromoperatoren von der traditionellen Datenstromverarbeitung als auch von weiteren sensorgestützten Anwendungsbeispielen abgeleitet und bezüglich des Einflusses auf alle relevanten Datenqualitätsdimensionen untersucht.

Verbesserung der Datenqualität Wenn die berechnete Datenqualität nicht den Anforderungen der Anwendung entspricht, müssen Maßnahmen zur Qualitätsverbesserung ergriffen werde. Dafür werden einerseits qualitätserhaltende Verfahren zum effizienten Ausgleich von Überlastsituationen entwickelt, die einen Hauptgrund der Qualitätsverschlechterung ausmachen. Andererseits wird die qualitätsbasierte Konfiguration der Datenverarbeitung als multikriterielles Optimierungsproblem definiert und Lösungsstrategien vorgestellt.

Visualisierung der Datenqualität Um Nicht-IT-Experten wie z.B. Technikern, Wartungsarbeitern oder Managern die umfassende Auswertung der zur Verfügung gestellten Sensor- und Datenqualitätsinformationen zu ermöglichen, muss eine nutzerfreundliche Plattform entwickelt werden. Sie unterstützt die graphische Modellierung von Datenstrom- bzw. Datenbankanfragen sowie die Visualisierung von Verarbeitungsergebnissen und deren Datenqualität.

Mit der Realisierung der genannten Konzepte bietet die vorliegende Arbeit eine Ende-zu-Ende-Architektur von Smart-Item-Sensoren zu Geschäftsanwendungen, die die transparente Propagierung, Speicherung und Verarbeitung von Datenqualitätsinformationen ermöglicht. Die entwickelten Werkzeuge erlauben die umfassende Evaluierung fehlerbehafteter Sensordaten, um falsche Entscheidungen zu reduzieren und im besten Falle ganz zu verhindern.

1.5. Aufbau der Arbeit

Kapitel 2 analysiert die Eigenschaften von Sensordaten und der Datenverarbeitung in Datenstromsystemen. Datenqualitätsdefinitionen verwandter Arbeiten werden verglichen, um die DQ-Dimensionen zu bestimmen, die zur Beschreibung der Sensordatenqualität eingesetzt werden müssen. Kapitel 3 diskutiert verwandte Arbeiten im Bereich Datenqualitätsmodellierung und -verarbeitung. Allgemeine Datenqualitätsmodelle und -methoden werden verglichen und Datenqualitätsaspekte in Informationssystemen und Datenströmen beschrieben.

Kapitel 4 befasst sich mit der Modellierung von Datenqualität. Es werden Datenstrukturen zur effizienten Propagierung von Datenqualitätsinformationen in der Datenstromverarbeitung vorgestellt. Außerdem wird die Erweiterung des relationalen Datenschemas zur persistenten Speicherung von Datenqualitätsdimensionen erläutert.

In Kapitel 5 wird die Verarbeitung von Datenqualitätsinformationen erklärt. Die Schritte der Datenverarbeitung müssen auf den Qualitätsinformationen nachvollzogen werden,

um die Auswirkungen der einzelnen Operatoren auf die Datenqualität abzubilden. Die entwickelte Datenqualitätsalgebra umfasst Operatoren aus der relationalen und numerischen Algebra sowie der Signalverarbeitung.

Kapitel 6 beschreibt zwei Ansätze zur Datenqualitätsverbesserung. Zum Einen wird die qualitätsgesteuerte Optimierung der Datenstromverarbeitung erläutert, mit deren Hilfe Nutzeranforderungen an die Datenqualität in den Verarbeitungsprozess integriert werden. Dazu wird die Konfiguration der Datenstromverarbeitung analysiert, um das Optimierungsproblem mit seinen Zielfunktionen zu definieren. Zur Lösung des Problems werden heuristische Optimierungsalgorithmen vorgestellt und diskutiert. Zum Anderen wird der qualitätsgesteuerte Lastausgleich vorgestellt, der in Überlastsituationen Datentupel minderer Qualität aus dem Datenstrom entfernt. Dadurch kann der durch den Datenverlust hervorgerufene Fehler kompensiert und die Datenqualität von Verarbeitungsergebnissen verbessert werden.

Die entwickelten Datenstrukturen und Algorithmen werden im Datenqualitätswerkzeug „QPIPEZ“ (Quality PIPEs & visualiZation) prototypisch umgesetzt, wobei die Visualisierung der Datenqualität besondere Beachtung findet. In Kapitel 7 wird dieser Prototyp beschrieben und anhand von künstlich generierten Datenströmen sowie Sensordaten realer Anwendungen validiert. Kapitel 8 fasst die Ergebnisse der vorliegenden Arbeit zusammen.

2
Anforderungsanalyse

In diesem Kapitel werden die Anforderungen an das zu entwickelnde Datenqualitätssystem analysiert. Zuerst werden in Abschnitt 2.1 vier Anwendungsszenarien vorgestellt, die in allen nachfolgenden Kapiteln zur Illustration der entwickelten Konzepte und Methoden aufgegriffen werden. In Abschnitt 2.2 werden die Eigenschaften von Sensoren und gemessenen Sensordaten untersucht und Methoden zur Messung der initialen Sensordatenqualität gegeben. Abschnitt 2.3 beschreibt die Datenverarbeitung in Datenstromsystemen. Hier werden die häufigsten Ressourcenbeschränkungen erläutert, um den Begriff der Datenqualität von der Dienstqualität in Datenstromsystemen abzugrenzen. Des Weiteren wird untersucht, welche Operatoren zur Sensordatenverarbeitung notwendig sind, um eine möglichst breite Menge an Anwendungen abzudecken. Im Anschluss erfolgt in Abschnitt 2.4 eine Literaturstudie zu Datenqualitätsdefinitionen und -dimensionen. Basierend auf den ermittelten Sensorfehlerquellen werden die Datenqualitätsdimensionen abgeleitet, die zur Beschreibung der Sensordatenqualität benötigt werden. Für jede Dimension werden Methoden zur Qualitätsmessung bzw. Informationsquellen zur Initialisierung der Datenqualität angegeben. Die ermittelten Anforderungen werden in Abschnitt 2.5 zusammengefasst.

2.1. Anwendungsszenarien

Im Folgenden werden Anwendungsszenarien beschrieben, um den Kontext der Sensordatenverarbeitung zu illustrieren und die vorliegende Arbeit zu motivieren. Außerdem dienen die enthaltenen Datenverarbeitungsschritte der Bestimmung der benötigten Datenstromoperatoren, für die eine Datenqualitätsverarbeitung definiert werden muss.

Szenario 1 - Vorausschauende Wartung

Bei der Wartung von Produktionsanlagen muss ein Kompromiss zwischen Wartungsaufwand und Maschinenausfällen gefunden werden. In den meisten Fällen wird eine periodische Wartung ausgeführt, so dass alle Maschinen in regelmäßigen Abständen überprüft werden. Dies kann dazu führen, dass die Produktion häufig wegen Wartungsarbeiten unterbrochen werden muss, obwohl kein Fehler vorliegt. Im Gegensatz dazu kann auf vorherige Wartung ganz verzichtet werden, so dass die Produktion nie gestört, aber nur auf Maschinenfehler reagiert wird. Tritt hier ein Fehler auf, ist eine Reparatur meist sehr teuer und längere Betriebsausfälle als bei regelmäßiger Wartung sind zu erwarten.

Einen Kompromiss bietet die vorausschauende Wartung, die mit Sensordaten gesteuert wird. Dafür werden Informationen über den Zustand der Produktionsmaschine aufgenommen und verarbeitet, um Fehlverhalten vorauszusehen und somit den idealen Wartungstermin zu bestimmen.

Im Beispielszenario wird ein Hydraulikbagger vorausschauend gewartet [MMS04]. Dafür werden vier Hydraulikzylindersysteme zur Steuerung des Baggerarms und der Schaufel auf Lecks oder Blockierungen geprüft. Um den Druck im Hydrauliksystem zu kontrollieren, werden Drucksensoren in den Hoch- und Niederdruckkammern jedes Zylinders angebracht. Eine Druckdifferenz über 200bar lässt auf eine Blockierung, eine Differenz unter 40bar auf ein Leck schließen. Bei starkem Druckanstieg oder -abfall muss eine Warnung ausgegeben und ein Wartungsauftrag ausgelöst werden. Des Weiteren werden Sensoren zur Messung der Öltemperatur, der Partikelverschmutzung, der Viskosität und des Wassergehalts verwendet, um das Alter des Hydrauliköls zu bestimmen [DB07]. Sinkt die verbleibende Lebenszeit unter 10% muss ein Ölwechsel eingeplant werden.

Szenario 2 - Effizientes Recycling

Die Altfahrzeugrichtlinie EU RL 200/53/EG der Europäischen Union verlangt, dass ab 2015 95% der Bauteile bei der Verschrottung eines PKWs wiederverwertet werden müssen. Um die Teile zu bestimmen, die den höchsten Restwert aufweisen, kann die „Lebensgeschichte“ der Autoteile mit Hilfe von Sensoren aufgezeichnet werden. Zum Beispiel können die Zahl der Kaltstarts, die gefahrenen Kilometer, das Geschwindigkeitsprofil der gefahrenen Strecken, der Verlauf der Öltemperatur und -verschmutzung sowie mögliche Unfälle den Restwert des Motorblocks, des Getriebes oder des Karosserierahmens beeinflussen.

Dazu wird das Auto mit den entsprechenden Sensoren und einer Datenverarbeitungseinheit versehen. Die gewünschten Sensordaten können aufgrund der beschränkten Speicherkapazität nicht über den gesamten Lebenszyklus des PKWs aufgezeichnet werden. Die Sensordaten werden fortlaufend verarbeitet, so dass nur der aktuelle Restwert pro Bauteil festgehalten wird. Dazu werden die Daten aggregiert, gewichtet und mit-

einander verknüpft. Beim Start des Recycling-Prozesses werden die berechneten Werte ausgelesen und unterstützen den Facharbeiter bei der Recycling-Planung.

Szenario 3 - Produktionskontrolle

Neben der Wartung von Produktionsanlagen ist auch die direkte Steuerung der Produktionsprozesse auf Basis von Sensordaten möglich. In vielen Fertigungshallen wird die Güte der hergestellten Produkte in mehreren Produktionsschritten kontrolliert, um Ausschuss so schnell wie möglich aus der Produktionslinie zu entfernen. Dies ist vor allem wichtig, wenn eine hohe Produktqualität erforderlich ist oder eine frühe Auslese die Produktionskosten senkt.

Im dritten Anwendungsszenario werden die Produktion von Kontaktlinsen betrachtet. Dabei wird die Dicke der Kontaktlinsenmitte und des Linsenrandes sowie die Axialverschiebung der Linsenkrümmung gemessen. Die Linsenqualität wird danach als gewichtete Summe dieser drei Messungen berechnet. Zum Einen wird die Qualität jeder Linse geprüft, um minderwertige Linsen aus der Produktion zu entfernen. Zum Anderen muss der Trendverlauf der Produktionsqualität überwacht werden, um den optimalen Zeitpunkt einer Rekalibrierung der Produktionsmaschinen einzuplanen.

Szenario 4 - Wetterprognose

Um Wettervorhersagen zu erstellen, muss eine Vielzahl von Sensorinformationen in kurzer Zeit ausgewertet werden. Zum Einen wird eine hohe Anzahl an verschiedenen Sensoren wie z.B. Temperatur, Luftfeuchte, Sonneneinstrahlung oder Luftdruck benötigt. Zum Anderen misst jeder dieser Sensoren mit einer hohen Datenrate, um feine Wetterschwankungen aufzuzeichnen. Das hohe Datenvolumen der Wetterinformationen stellt somit besondere Ansprüche an die Ressourcen des Datenstromsystems.

In der vorliegenden Arbeit werden Methoden der Wetterprognose mit Hilfe realer Wetterdatensätze [HWL08] simuliert, welche die Wolken- und Wetterinformationen über mehrere Jahre für eine Vielzahl von Wetterstationen zu Lande und auf Schiffen beinhalten. Durch geschickte Auswertung dieser Daten lassen sich interessante Wetterveränderungen der Vergangenheit ermitteln oder Prognosen für den Wetterverlauf validieren.

2.2. Eigenschaften von Sensordaten

In diesem Abschnitt wird der Aufbau eines Smart-Item-Sensors beschrieben. Im Anschluss werden Fehler, die beim Messen mit Sensoren auftreten können, untersucht.

Ein digitaler Sensor eines Smart-Items ist in Abbildung 2.1 schematisch dargestellt. Er bildet eine physikalische Eigenschaft (zum Beispiel Druck oder Luftfeuchtigkeit)

auf ein Spannungssignal ab, das nachfolgend abgetastet und somit diskretisiert wird. Um die rechnergestützte Datenverarbeitung zu ermöglichen, wird das Signal anschließend digitalisiert. Die Ausgabe eines Sensors ist somit eine diskretisierte, digitalisierte, numerische Messreihe, die einen zeitkontinuierlichen, analogen Messwert beschreibt.

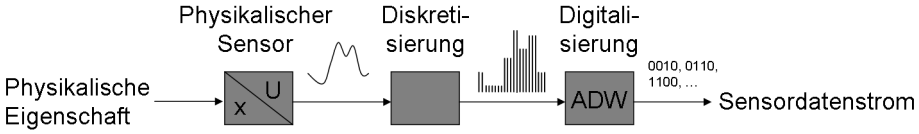


Abbildung 2.1.: Aufbau eines Smart-Item-Sensors

Dabei ist es nicht möglich, fehlerfrei zu messen. Die Differenz eines aus Messungen gewonnenen Wertes und des wahren Wertes der Messgröße wird Messabweichung (nach DIN 1319-1:1995) oder Messfehler genannt. Dabei können folgende Fehlerursachen auftreten [Str08].

Messgerätefehler als Folge der Unvollkommenheit der Konstruktion, Fertigung, Justierung (z. B. durch Werkstoffe oder Fertigungstoleranzen)

Verfahrensfehler bedingen Einflüsse infolge der Einwirkung der Messeinrichtung auf die Messgröße (z. B. Rückwirkungsabweichung durch Eigenverbrauch des Messgerätes)

Umwelteinflüsse als Folge von Änderungen der Einwirkungen aus der Umgebung (z.B. Temperatur, äußere elektrische oder magnetische Felder, Lage, Erschütterungen)

Instabilitäten des Wertes der Messgröße oder des Trägers der Messgröße (z. B. statistische Vorgänge oder Rauschen)

Beobachtereinflüsse infolge unterschiedlicher Eigenschaften und Fähigkeiten des Menschen (z. B. Aufmerksamkeit, Übung, Sehschärfe, Schätzvermögen oder Parallaxe)

Außerhalb der Diskussion stehen Messfehler durch Irrtümer des Beobachters [Bai83], Verfälschungen durch Wahl ungeeigneter Mess- und Auswerteverfahren sowie das Nichtbeachten bekannter Störgrößen.

Es gibt zwei Arten von Messabweichungen. Durch systematische Messfehler wird ein Messergebnis immer unrichtig. Systematische Fehler sind unter gleichen Umweltbedingungen bei gleicher Messanordnung immer gleich groß und besitzen das gleiche Vorzeichen. Sie lassen sich durch Methoden der Fehlerrechnung abschätzen und zum Teil korrigieren. Messgerätefehler und Verfahrensfehler verursachen systematische Messfehler. Außerdem können Umwelteinflüsse den Messwert systematisch verfälschen. Zum Beispiel können hohe Temperaturen zu überhöhten Druckmessungen führen.

Durch zufällige Messabweichungen wird ein Messergebnis unsicher. Dabei schwankt das unsichere Messergebnis um den wahren Messwert. Diese statistischen Fehler können

sowohl positiv als auch negativ sein. Durch ihren zufälligen Charakter können sie nicht korrigiert werden. Ursachen für statistische Fehler sind zufällige Umwelteinflüsse (z.B. Erschütterungen), Instabilitäten und Beobachtereinflüsse.

Messgerätefehler lassen sich durch Vergleich mit einem wesentlich besseren Messgerät bestimmen. Sie sind also systematischer Natur und im Prinzip - jedoch mit sehr hohem Aufwand - korrigierbar. Der Hersteller eines Sensors kann den systematischen Fehler in zwei Arten angeben. Die Fehlerkurve eines Messgerätes ist die Darstellung der Fehlerverteilung über den Messwertebereich in einem Diagramm oder einer Tabelle. Anhand der Fehlerkurve sind Betrag und Vorzeichen des Fehlers zu einem Messwert abzulesen. Da die Fehlerkurve den Fehler nur zu einem bestimmten Zeitpunkt und unter spezifischen Umweltbedingungen dokumentiert, wird meistens darauf verzichtet. Der Hersteller garantiert lediglich Fehlergrenzen unter gewissen Bedingungen (z.B. Temperatur: 0°C-40°C, Feuchtigkeit: 30%-50%). Dabei wird der Fehler als prozentualer Anteil des Maximalmesswertes des jeweiligen Messbereiches angegeben. [Hyd09] zeigt zum Beispiel die technischen Daten eines HYDAC Drucksensors.

Verfahrensfehler können im Vergleich mit anderen Messverfahren oder -methoden geschätzt werden. Sie können jedoch selten vollständig erfasst werden, da die meisten Messverfahren die zu messende Größe beeinflussen. Auch systematische *Umwelteinflüsse* lassen sich durch zusätzliche Vergleichsmessungen bestimmen. So könnte zum Beispiel die Fehlerkurve in Abhängigkeit der Umwelttemperatur bestimmt werden. Im Allgemeinen wird aufgrund des hohen Aufwandes jedoch darauf verzichtet.

Der *statistische Messfehler* wird mit Hilfe von Kenngrößen von Zufallsvariablen ausgedrückt. Dabei wird das Messergebnis x als Realisation einer Zufallsvariablen X aufgefasst. Diese wird vollständig durch ihre Wahrscheinlichkeitsdichtefunktion beschrieben, die auch Verteilungsdichtefunktion oder kurz Verteilung genannt wird. Die Wahrscheinlichkeitsdichtefunktion $Pr(\chi)$ beschreibt die Wahrscheinlichkeit p dafür, dass die wahre physikalische Größe $\hat{x}$ im Intervall $[x-a; x+b]$ liegt. Das Intervall wird Konfidenzintervall genannt; die Wahrscheinlichkeit p ist die Konfidenzwahrscheinlichkeit.

$$p = P\{x - a \leq \hat{x} \leq x + b\} \quad = \quad \int_{x-a}^{x+b} Pr(\chi) d\chi \tag{2.1}$$

Die weitaus häufigste Fehlerverteilung ist die Gaußverteilung, auch Normalverteilung genannt. Sie enthält zwei Parameter, den Erwartungswert μ, der den wahren Messwert $\hat{x}$ approximiert, und die Standardabweichung σ. Damit ergibt sich die Wahrscheinlichkeitsdichte aus Gleichung 2.1 in Abhängigkeit des Abstandes zwischen wahrem Wert $\hat{x}$ und Messwert x ($a = b$) wie folgt.

$$Pr(x) \quad = \quad \frac{1}{\sqrt{2\pi\sigma^2}} e^{-\frac{(x-\hat{x})^2}{2\sigma^2}} \tag{2.2}$$

Ziel ist nun die Bestimmung des Konfidenzintervalls $[x - \epsilon; x + \epsilon]$ bei einer gegebenen Konfidenzwahrscheinlichkeit p, wobei ϵ den statistischen Messfehler beschreibt. Die Standardabweichung muss mit dem $(1 - p/2)$-Quantil multipliziert werden. Tabelle 2.1 zeigt beispielhafte Lösungen für dieses Problem.

Konfidenz-wahrscheinlichkeit	68,3%	95%	95,4%	99%	99,7%	99,99%
Konfidenzintervall	$\epsilon = \sigma$	$\epsilon = 1,96\sigma$	$\epsilon = 2\sigma$	$\epsilon = 2,58\sigma$	$\epsilon = 3\sigma$	$\epsilon = 4,265\sigma$

Tabelle 2.1.: Vertrauensniveaus einer normal verteilten Messgröße

Im Folgenden werden statistische Fehler als normal verteilt angenommen. Die Aussagen in Kapitel 4, 5 und 6 gelten jedoch auch für jede andere Verteilungsfunktion.

Neben den numerischen Messfehlern stellen Sensorausfälle ein großes Problem in sensorgestützten Anwendungen dar. Besonders in mobilen Geräten kann es zu kurzzeitigen Ausfällen durch Erschütterungen oder Stöße kommen. Der Sensor liefert für die entsprechende Zeitspanne keine Datenwerte, so dass Null-Werte in der Datenstromverarbeitung erscheinen. Um eine kaskadierende Fortsetzung der Null-Werte in der Datenverarbeitung zu vermeiden, müssen die fehlenden Messwerte aufgefüllt werden. Dafür bieten sich folgende Möglichkeiten an.

- Einsetzen von Standardwerten
- Konstante Wiederholung des letzten Messwertes
- Schätzen des Messwertes

Die Schätzung des wahrscheinlichsten Messwertes kann auf unterschiedliche Art erfolgen. So kann der Durchschnittswert der bisherigen Messungen eingetragen werden oder eine Interpolation der fehlenden Messwerte stattfinden. Außerdem kann auf komplexere Methoden aus dem Data-Warehouse- und Data-Mining-Bereich zurückgegriffen werden [BK03], [MSO01], [QSB97], [SS07]. In der folgenden Arbeit werden fehlende Messwerte interpoliert, da die Interpolation eine gute Wertabschätzung mit akzeptablem Aufwand darstellt.

Sensordaten unterliegen drei verschiedenen Fehlerarten: *Systematische* und *statistische Messabweichungen* unterscheiden sich in ihrem Charakter und der Fehlerfortpflanzung [Pap06], [D'A95], [Gra05] und müssen daher getrennt betrachtet und verarbeitet werden. Außerdem müssen *Sensorausfälle* dokumentiert werden. Im Folgenden wird die Datenverarbeitung von Sensordaten beleuchtet, um weitere Fehlerarten oder Datenqualitätsprobleme aufzudecken.

2.3. Datenverarbeitung im Datenstromsystem

Datenstrom-Management-Systeme wurden entwickelt, um Ressourcenbeschränkungen in Smart-Item-Umgebungen zu begegnen. Im Datenstrommodell liegen die zu verarbeitenden Daten nicht persistent gespeichert auf einer Festplatte vor, sondern erreichen die Operatoren der Verarbeitungskette in einem oder mehreren fortlaufenden Datenströmen. Ein Datenstromsystem unterscheidet sich deshalb in den folgenden Punkten vom traditionellen relationalen Datenbankkonzept.

- Die Datenelemente des Stroms treffen zur Laufzeit am Verarbeitungspunkt ein.
- Die Länge eines Datenstroms kann unbegrenzt sein.
- Nachdem ein Datenelement verarbeitet wurde, wird es in der Regel verworfen und ist dann dem aktuellen Verarbeitungsknoten nicht mehr bekannt.

Damit sind nur bestimmte Algorithmen und Operatoren anwendbar. Auch die Auswertungszeit ist oft beschränkt, da zeitkritische Anwendungen schnelle Ergebnisse erwarten. In [BBD+02], [GO03] und [Mut05] werden Datenstrommodelle, Methoden zur Anfrageverarbeitung stetiger Datenflüsse und Beispielanwendungen vorgestellt. Anfragen an Datenstromsysteme werden in zwei Klassen unterteilt. Es gibt Anfragen, die (ähnlich traditionellen Datenbankanfragen) einmalig ausgeführt werden, und kontinuierliche Anfragen, die über einen bestimmten Zeitraum für stetig eintreffende Datenstromtupel bearbeitet werden. In beiden Fällen kann zwischen vordefinierten Anfragen und Ad-Hoc-Anfragen unterschieden werden, wobei vordefinierte Anfragen die Optimierung der Datenstromverarbeitung ermöglichen.

Kontinuierliche Sensormessreihen werden im DSMS als Datenströme verarbeitet. Sie werden im Folgenden als Sensordatenströme bezeichnet.

Definition 2.1 Ein Sensordatenstrom D ist als diskrete numerische Messreihe eines oder mehrerer physikalischer Sensoren definiert. Jedes Tupel τ eines Sensordatenstroms entspricht einem oder mehreren Messwerten, die zum Zeitpunkt t aufgenommen werden. Ein Sensordatenstromtupel besteht somit aus n Attributmesswerten $A_i (1 \leq i \leq n)$ und einem Zeitstempel t.

Die unterschiedlichen Anfragetypen nehmen keinen Einfluss auf die Datenqualität von Sensordatenströmen. Das in dieser Arbeit entwickelte Modell zur Verwaltung, Verarbeitung und Speicherung von Datenqualitätsinformationen in Sensordatenströmen ist unabhängig vom Zeitpunkt und der Zeitdauer der Datenstromauswertung und unterstützt somit alle vier Anfrageklassen. Im Folgenden sind einige Beispiele für Datenstromsysteme aufgelistet.

Aurora ist ein prototypisches Datenstrom-Management-System, das ein graphisches Werkzeug zum Erstellen von Operatorbäumen anbietet. [ACc+03a], [ACc+03b], [CcC+02]. Das Operatoren-Scheduling (dt. Planungserstellung) und Load-Shedding (dt. Lastausgleich) verbessert die Systemeffizienz.

STREAM (STanford stReam datA Manager) unterstützt deklarative, kontinuierliche Anfragen über Datenströmen und Relationen. Als Anfragesprache wurde im Rahmen dieses Projekts die Continuous Query Language (CQL) als Erweiterung der Datenbankanfragesprache Structured Query Language (SQL) entwickelt [ABB+03].

TelegraphCQ implementiert die Telegraph-Datenfluss-Engine basierend auf dem Datenbanksystem PostgreSQL. Die Anfragesprache Continously Adaptive Continous Queries (CACQ) adressiert Datenströme mit großem Datenvolumen und stark schwankenden Datenraten [CC03], [KCC+03].

QStream basiert auf einem Operatoren-Komponenten-Modell und wird in einer echtzeitfähigen Umgebung ausgeführt. QStream [SBL04], [SLSL05] bietet ein deterministisches Datenstromsystem, das eine vordefinierte Dienstqualität für Datenstromanfragen garantiert.

PIPES ist eine flexible, erweiterbare Architektur, die alle fundamentalen Komponenten zur Verfügung stellt, um ein Datenstrom-Management-System [KS04] zu implementieren. PIPES basiert auf der Java Bibliothek XXL [CHK+03] und erweitert sie um kontinuierliche Datenverarbeitung von autonomen Datenquellen.

Gigascope wurde entwickelt, um Datenströme mit sehr hoher Datenrate in Netzwerken zu kontrollieren. Um eine sehr schnelle Verarbeitung auf einfachen Prozessoren zu erlauben, wurde das Datenstrom-Management stark vereinfacht. So können zum Beispiel keine statischen Relationen verarbeitet werden. Fensteroperatoren wie der Verbund oder Aggregationen werden nicht unterstützt [CJSS03a], [CJSS03b], [CGJ+02].

Die vorgestellten Datenstromsysteme unterstützen den Lastausgleich in Überlastsituationen. Da Hardware-Ressourcen beschränkt sind und Datenstromraten schwanken können, kann es Situationen geben, in denen das System nicht alle einströmenden Datentupel verarbeiten kann. Um den Datendurchsatz zu stabilisieren und sehr große Verzögerungen der Datenverarbeitung zu vermeiden, müssen überzählige Tupel aus dem Datenstrom entfernt werden. Methoden sowie Vor- und Nachteile des Lastausgleichs werden in Abschnitt 3.3.1 detailliert erläutert.

Die Continuous Query Language, entwickelt im Rahmen des STREAM-Projektes [ABW06] stellt die umfassendste Datenstromalgebra dar. Abbildung 2.2 zeigt die drei Operatorklassen der CQL.

Strom-zu-Relation-Operatoren nutzen Fensterspezifikationen, um Datenstromfenster auf statische Relationen abzubilden. Datenstromfenster können zeitbasiert [Range 30 Seconds], tupelbasiert [Rows N], und partitioniert [Partition By A1,...,Ak Rows N] ähnlich der traditionellen Gruppierung der SQL definiert werden. Landscape-Fenster haben einen festen Startzeitpunkt und wachsen mit der Zeit. Zum Beispiel wird der Tagesumsatz eines Supermarktes beginnend jeden Morgen kontinuierlich über den ganzen Tag berechnet. Sliding-Windows (gleitende Fenster) hingegen werden mit Hilfe der Fensterlänge und der Schrittweite definiert. Für einen Fondsmanager ist der durch-

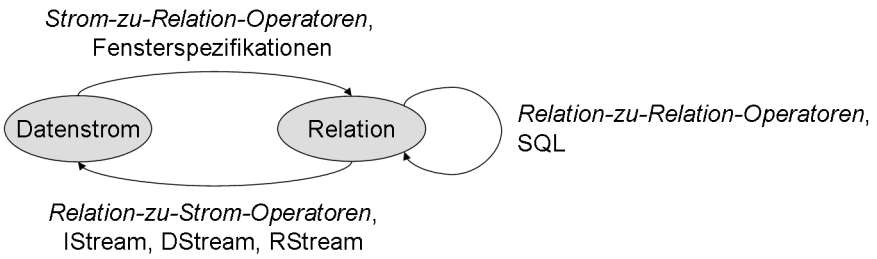


Abbildung 2.2.: Operatorenklassen der CQL

schnittliche Aktienwert der letzten Stunde von Interesse (Fenstergröße $1h$), der jede Minute aktualisiert wird (Schrittweite $1min$).

Datenstromfenster werden anschließend mit Hilfe der traditionellen relationalen Datenbankanfragesprache SQL verarbeitet werden. Tabelle 2.2 fasst die hierfür zur Verfügung stehenden *Relation-zu-Relation-Operatoren* der CQL zusammen.

CQL-Operator	**Definition**
Projektion	Duplikaterhaltende Auswahl bestimmter (Datenstrom-)Attribute
Duplikateliminierung	Entfernung von Duplikaten aus dem Ergebnisstrom
Verbund	Kombination zweier Datenströme auf Basis des Verbundattributes
Aggregation	Zusammenfassung einer Gruppe von Datensätzen zu einem Ergebniswert
Selektion	Entfernung von Datensätzen entsprechend dem Selektionsprädikat
Vereinigung	Vereinigung der Ergebnistupelmengen
Differenz	Differenz der Ergebnismengen
Schnittmenge	Schnittmenge der Ergebnistupel

Tabelle 2.2.: Relation-zu-Relation-Operatoren der CQL

Die relationale Projektion wählt eine bestimmte Menge an Attributen der Relation aus und führt anschließend eine Duplikateliminierung aus, so dass alle Datentupel der Ergebnismenge verschiedene Wertausprägungen aufweisen. Sie wird in der Datenstromverarbeitung der CQL in zwei separate Operatoren aufgespalten.

Der traditionelle relationale Verbund vergleicht alle Datentupel der Eingangsrelationen. Datentupel gleicher Wertausprägung im Verbundattribut werden beim inneren Verbund zu einem Ergebnistupel verknüpft. Der (rechte bzw. linke) äußere Verbund fügt

außerdem alle Datentupel der (rechten bzw. linken) Eingangsrelation hinzu, die keinen Verbundpartner gefunden haben. Beim Verbund zweier Datenströme können lediglich Ausschnitte der Ströme, d.h. Datenstromfenster, auf Verbundpartner untersucht werden, da der begrenzte Speicherplatz es nicht erlaubt, alle Stromdatensätze zwischenzuspeichern [SW04], [KNV02]. Ähnliche Verfahren werden beim Sequenzenvergleich (engl. Sequence Matching) angewendet [Moo06].

Aggregation und Selektion der SQL und CQL unterscheiden sich nicht in ihrer Ausführung. Bei der Aggregation von Datenströmen wird der Zeitstempel, d.h. die oben beschriebenen Datenstromfenster, als häufigstes Gruppierungsattribut gewählt [AW04]. Auch die Mengenoperatoren Vereinigung, Differenz und Schnittmenge der relationalen Algebra können ohne Änderungen auf Datenstromfenstern angewendet werden.

Die *Relation-zu-Strom-Operatoren* IStream, DStream und RStream erzeugen einen Ergebnisdatenstrom aus den Ergebnistupeln der relationalen Datenverarbeitung. Mit dem IStream-Operator wird in jedem Verarbeitungsschritt ein Ergebnistupel an den Datenstrom gesendet. Der DStream-Operator liefert alle Tupel, die während der Anfrageverarbeitung aus der Relation entfernt werden (z.B. alle Kundennummern gelöschter Kunden). RStream wandelt wiederum die gesamte Ergebnisrelation in einen Datenstrom um.

Strom-zu-Strom-Operatoren werden nicht eigenständig definiert, sondern aus diesen drei Klassen zusammengesetzt. Mit diesem Modell ist es möglich, alle Operatoren der relationalen Algebra auf Datenströme abzubilden und bekannte Optimierungsstrategien der relationalen Datenverarbeitung zu nutzen.

Sensordatenströme repräsentieren numerische Datenwerte, die mit Hilfe von mathematischen Operatoren miteinander verknüpft werden können. So wird zum Beispiel die Druckdifferenz (Szenario 1, Abschnitt 2.1) als Subtraktion von Hoch- und Niedrigdruck berechnet. Arithmetische Operatoren verarbeiten reellwertige Datensätze. Boolesche Operatoren der Aussagenlogik kombinieren die Aussagen *wahr* und *falsch*. Der Schwellwertvergleich bildet reellwertige Datentupel auf Boolesche Werte ab. Die in dieser Arbeit untersuchten numerischen Operatoren sind in Tabelle 2.3 zusammengefasst.

Numerische Operatoren	**Beispiele**
Arithmetik	Addition, Subtraktion, Quadratwurzel
Aussagenlogik	Negation, Disjunktion
Schwellwertvergleich	

Tabelle 2.3.: Numerische Operatoren

Des Weiteren entsprechen Sensordatenströme analogen Messsignalen, so dass auch Operatoren aus der Signalanalyse in der Datenstromverarbeitung Anwendung finden. Die Frequenzanalyse der Öltemperatur in einem Hydrauliksystem kann zum Beispiel Aussagen über periodische Temperaturschwankungen geben. Der Vergleich mit Fre-

quenzen der Ölpumpe, der Ventilfunktionen oder der Zylinderbewegung kann helfen, kritische Situationen zu erkennen und zu umgehen. Das Sampling wird zur Datenreduktion in Überlastsituationen benötigt. Schließlich wird die Interpolation genutzt, um die Datenraten asynchroner Datenströme für einen nachfolgenden Verbund anzupassen [SFL05]. Tabelle 2.4 fasst die Operatoren der Signalanalyse von Sensordatenströmen zusammen.

Quasi-analoge Operatoren	**Beschreibung**
Interpolation	Neue Datensätze werden an der Trendlinie bestehender Messwerte eingefügt.
Sampling	Eine zufällige Stichprobe der Datensätze wird erzeugt.
Frequenzanalyse	Die im Signal enthaltenen Frequenzen werden bestimmt.
Frequenzfilter	Bestimmte Frequenzbänder werden aus dem Signal entfernt.

Tabelle 2.4.: Operatoren der Signalverarbeitung

Sowohl bei der Aggregation als auch bei der Frequenzanalyse werden Daten eines Messattributes zu einer Synopse, einer kleineren Datenrepräsentation, zusammengefasst. Dabei ist die Synopse umso repräsentativer, umso größer die zugrunde liegende Datenmenge ist. So ist beispielsweise der durchschnittliche Druckverlust, der auf Druckmessungen mit der Datenstromrate $r = 1/s$ beruht, vertrauenswürdiger als der Druckverlust, der auf Basis von Messungen mit $r = 1/min$ berechnet wurde. Damit ist die Menge der Rohdaten, die von einem berechneten Wert repräsentiert wird, ein Qualitätsmerkmal des Datenstroms.

Sensordaten müssen so schnell wie möglich verarbeitet werden, um schneller auf Ereignisse reagieren zu können und/oder komplexe Rechnungen im vorgegebenen Zeitrahmen zu ermöglichen. Für Entscheidungen, die auf Basis von Sensordaten getroffen werden, ist das Alter der Daten von hoher Relevanz. So ist zum Beispiel der Arbeiter in der Kontaktlinsenproduktion an der gegenwärtigen Ausschussrate interessiert. Die Daten von vor einer Stunde sind zu alt, um eventuelle Produktions- oder Maschinenfehler rechtzeitig aufzudecken. Die Aktualität der Daten spielt somit auch im Sensordatenbereich eine wichtige Rolle.

Die *Menge der Rohdaten* und deren *Alter* werden somit zu der Menge der kritischen Datenqualitätsaspekte des *systematischen* und *statistischen Fehlers* sowie der Anzahl der *Sensorausfälle* zur Beschreibung der Sensordatenqualität hinzugefügt.

2.4. Definitionen der Datenqualität

Es existieren eine Vielzahl an Auffassungen und Definitionen des Begriffs Datenqualität. Je nach Anwendungskontext wird die Qualität von Daten unter einem anderen Blickwinkel bewertet, so dass unterschiedliche Prioritäten bei der Bestimmung von Datenqualität zu unterschiedlichen Definitionen führen. In diesem Abschnitt werden zuerst verschiedene Datenqualitätsdefinitionen verglichen, um dann eine Definition für Sensordatenqualität auf Basis der in Abschnitt 2.2 und 2.3 gefundenen Qualitätsmerkmale abzuleiten.

2.4.1. DQ-Definitionen in der Literatur

In allgemeinen Worten beschreibt die Datenqualität die Tauglichkeit (engl. fitness for use) der Daten für eine gegebene Anwendung [Jur88]. Dabei wird jedoch nicht angegeben, was die Tauglichkeit, das heißt die Angepasstheit oder Angemessenheit, der Daten beschreibt oder wie diese bestimmt werden kann. Ähnlich verhält es sich mit Datenqualitätsdefinitionen, welche die „Usability“ (dt. Brauchbarkeit) von Daten als Bewertungsmaßstab für Datenqualität ansetzen [ES07]. Hier werden Fragebögen und Nutzerbefragungen als Methode zur Bestimmung der Datenqualität vorgeschlagen.

Da die Datenverarbeitung in Sensor-Umgebungen mitunter sehr komplex sein kann, ist es dem Nutzer nicht möglich die Datenqualität des extrahierten Wissens auf Basis der Initialdatenqualität der Sensoren abzuschätzen. Oft tritt sogar der Fall ein, dass die Initialbelegungen der unterschiedlichen Datenqualitätsdimensionen dem Datenkonsumenten nicht bekannt sind. Eine Nutzerbefragung zur Bestimmung der Datenqualität führt in dem vorliegenden Anwendungskontext also nicht zum gewünschten Ziel. Diese schwierige Aufgabe der Qualitätsbestimmung soll dem Nutzer abgenommen werden. Die Datenqualitätsinformationen sollen nutzertransparent vom Sensor zur Software-Anwendung transferiert und entsprechend der durchgeführten Datenverarbeitung angepasst werden.

Datenqualität ist multidimensional. Sie wird mit Hilfe von verschiedenen Merkmalen, Aspekten oder Dimensionen beschrieben, die potenzielle DQ-Probleme charakterisieren. Datenqualitätsdefinitionen sind daher aus einer Menge von Datenqualitätsdimensionen zusammengesetzt. Dabei stellt die folgende Gruppe die im Datenbankbereich am häufigsten genutzten Datenqualitätsdimensionen dar.

- Genauigkeit (engl. accuracy)
- Vollständigkeit (engl. completeness)
- Konsistenz (engl. consistency)
- Aktualität (engl. timeliness, up-to-dateness)

Die Genauigkeit beschreibt dabei die Abweichung des in der Datenbank gespeicherten zum realen Datenwert. Die Genauigkeit von numerischen Werten kann mit Hilfe des euklidischen Abstandes, der Unterschied zwischen zwei Zeichenketten mit der Levenshtein-Distanz zwischen gespeichertem und realem Wert berechnet werden. Die Vollständigkeit erfasst, in wie weit alle relevanten Informationen vorliegen. Einzelne Attributwerte, aber auch ganze Datentupeleinträge können fehlen. Daten sind inkonsistent, wenn sie Widersprüche enthalten und nicht aktuell, wenn Änderungen in der Realwelt nicht in der Datenbank nachvollzogen wurden.

Tabelle 2.5 zeigt Beispiele der genannten Datenqualitätsprobleme. Im ersten Tupel ist „Königsplattz“ falsch geschrieben und damit ungenau. Der Straßenname des zweiten Tupels fehlt, so dass dieses unvollständig ist. Das dritte Tupel zeigt ein Beispiel für inkonsistente Daten, denn die Postleitzahl „40762“ liegt nicht in „Dresden“. Der Eintrag des vierten Tupels ist nicht aktuell; die „Knopstraße“ wurde am 19.11.2007 umbenannt.

ID	Nachname	Vorname	Straße	Stadt	Postleitzahl
1	Schmidt	Thomas	Königsplattz	München	80333
2	Weihmann	Dieter	null	Berlin	13353
3	Ludwig	Simone	Liebigstraße	Dresden	40762
4	Werner	Sabrina	Knopstraße	Leipzig	06118

Tabelle 2.5.: Datenqualitätsprobleme

Im Folgenden werden weitere Datenqualitätsdefinitionen aus der Literatur vorgestellt. Bereits 1990 wurde die Datenqualität beim jährlichen Treffen der Southern California Online User Group (SCOUG) thematisiert. In einer Brainstorming-Sitzung [Bas90] wurden Datenqualitätsdimensionen definiert, die vor allem auf Datenbankperformanz und Nutzersicht der Daten gerichtet sind, z.B. Antwortzeit, Verfügbarkeit, Dokumentation und Kundensupport.

Bei der Suche nach Maßeinheiten für die Nutzerzufriedenheit definiert Kriebel in [Kri78] und [Kri79] die Datenqualitätsdimensionen Genauigkeit, Aktualität, Präzision, Zuverlässigkeit, Vollständigkeit und Relevanz. Redman [Red97] erweitert diese Datenqualitätsdimensionen der Instanzebene um Dimensionen des konzeptuellen Datenschemas (Attributgranularität, Konsistenz, Robustheit, Gültigkeitsbereiche, etc.) sowie Dimensionen der Formatebene (u.a. Interpretierbarkeit, Portierbarkeit, Speichereffizienz).

Chen et al. [CZW98] fügen weitere DQ-Dimensionen hinzu. Sie betrachten die Qualität von Webanfragen unter Berücksichtigung zeitrelevanter Kriterien wie Antwortzeit, Netzwerklatenzzeit, etc. Die Datenqualität aus Prozess- bzw. Systemsicht wird von Weikum in [Wei99] untersucht. Auch hier werden technische Dimensionen wie Verifizierbarkeit und Latenzzeit fokussiert. In [AA87] hingegen wird die Zuverlässigkeit der Daten betrachtet. Sie bestimmt, in welchem Ausmaß die vorhandenen Daten mit den Nutzeranforderungen und/oder der Realität übereinstimmen.

Basierend auf einer Befragung von Datenkonsumenten in größeren Unternehmen stellen Wang und Strong eine initiale Liste von 179 Datenqualitätsdimensionen auf [WS96], die nicht nur einzelne Datenwerte, sondern komplexe Datenmengen beschreiben. Sie analysieren diese Liste unter Betrachtung der verschiedenen Sichten auf die Datenqualität und extrahieren die in Tabelle 2.6 genannten 15 Qualitätsmerkmale. Die Qualität der Rohdaten (z.B. Genauigkeit, Vollständigkeit) wurde mit Nutzeranforderungen (z.B. Verständlichkeit, Zugänglichkeit) kombiniert.

In [SLW97] wird diese Kategorisierung näher diskutiert. Für jede DQ-Kategorie werden Problemmuster angegeben und Hinweise für Verantwortliche im Datenqualitätsmanagement abgeleitet. Jarke und Vassiliou [JV97] wenden die DQ-Kategorien auf Data-Warehouse-Umgebungen an, um unter anderem die Qualität von aggregierten Daten zu bestimmen.

Kategorie	**Dimension**
Intrinsische Datenqualität	Genauigkeit Glaubhaftigkeit Objektivität Reputation
Kontextuelle Datenqualität	Vollständigkeit Relevanz Aktualität Datenmenge Mehrwert
Repräsentative Datenqualität	Interpretierbarkeit Verständlichkeit Konsistenz Prägnanz
Zugriffsqualität	Zugänglichkeit Sicherheit

Tabelle 2.6.: Kategorien der Datenqualität nach [WS96]

In [Nau02] werden die bisher vorgestellten Dimensionen und Datenqualitätsdefinitionen zusammengefasst, verglichen und klassifiziert. Naumann teilt die vorliegenden DQ-Dimensionen in inhaltsbezogene Merkmale (Genauigkeit, Vollständigkeit, Relevanz, Interpretierbarkeit), technische Merkmale (Verfügbarkeit, Antwortzeit, Latenz), intellektuelle, subjektive Merkmale (Glaubwürdigkeit, Reputation) und instanzbezogene Merkmale (z.B. Datenmenge, Verständlichkeit, Verifizierbarkeit).

Rahm et al. betrachten in [RD00] auftretende Datenqualitätsprobleme. Sie unterscheiden Probleme, die einer Datenquelle entstammen, von denen, die durch Integration mehrerer Datenquellen entstehen (siehe Abbildung 2.3). Des Weiteren klassifizieren

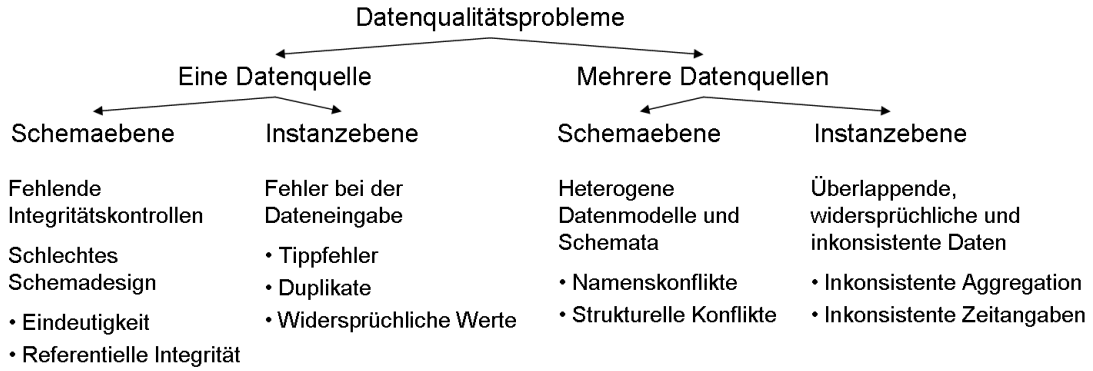


Abbildung 2.3.: Klassifikation von Datenqualitätsproblemen [RD00]

sie Datenqualität bezüglich der Ebenen, an denen Fehler auftreten können. So gibt es Datenfehler auf Schemaebene (Namens- , Integritätskonflikte) sowie auf Instanzebene (Duplikate, fehlende Werte, Inkonsistenzen).

Ein zweiter Zugang zu Datenqualität über die auftretenden Probleme wird in [LC02] diskutiert, wobei eine evolutionäre Sichtweise auf Datenqualität vorgestellt wird (siehe Abbildung 2.4). Im Datenverarbeitungsprozess treten zuerst Kollektionsprobleme auf, die Dimensionen wie Genauigkeit und Vollständigkeit beeinträchtigen. Dann wird die Organisationsqualität (Konsistenz, Navigierbarkeit) zum Beispiel durch Systemfehler verschlechtert. Die Präsentationsqualität kann durch Auswahl- oder Interpretationsfehler in Mitleidenschaft gezogen werden. Schließlich verschlechtern weitere Beschränkungen und die Alterung der Daten die resultierende Anwendungsqualität.

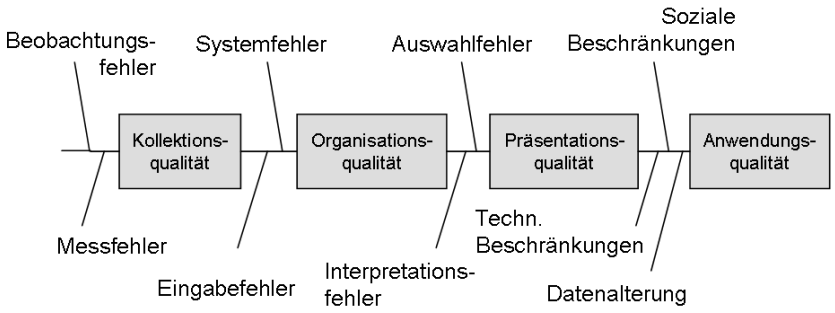


Abbildung 2.4.: Evolution der Datenqualität nach [LC02]

Datenqualitätsdimensionen sind oft voneinander abhängig bzw. stehen in Konflikt miteinander, so dass domänenspezifisches Wissen benötigt wird, um die Gesamtqualität zu bewerten. Jang et al. stellen ein System [JIW95] vor, das dem Datenkonsumenten erlaubt, dieses domänenspezifische Wissen über Relationen zwischen DQ-Dimensionen einzubringen. In Anlehnung an Arbeiten im Bereich der künstlichen Intelligenz wurde

ein Qualitäts-Calculus entwickelt, der die Gesamtqualität aus den Nutzereingaben und den einzelnen DQ-Dimensionen ableitet. Die im folgenden Abschnitt vorgestellten Dimensionen zur Beschreibung der Sensordatenqualität sind jedoch so gewählt, dass keine Abhängigkeiten bestehen. Sie sind orthogonal zueinander und beleuchten unterschiedliche Aspekte der Datenqualität. Soll eine Gesamtqualität berechnet werden, müssen die einzelnen Dimensionen mit Hilfe einer einfachen Form des Qualitäts-Calculus gewichtet werden.

2.4.2. Definition der Sensordatenqualität

Datenqualität wird oft subjektiv wahrgenommen, da die Qualitätsbewertung abhängig vom Anwendungskontext und der Einschätzung des Datenkonsumenten ist. Die Bewertung der Sensordatenqualität soll jedoch so objektiv wie möglich gestaltet werden, so dass unterschiedliche Anwendungen und Nutzergruppen bedient werden können. Deshalb werden zur Bewertung der Datenqualität von Sensordatenströmen vor allem Datenqualitätsdimensionen betrachtet, die Eigenschaften der Rohdaten beschreiben und an den Sensorknoten aufgenommen werden.

Die Kollektionsqualität (siehe Abbildung 2.4) der Rohdaten wird durch Mess- und Beobachtungsfehler beeinträchtigt. Wie bereits in Abschnitt 2.2 erläutert, werden keine groben Nutzerfehler wie zum Beispiel Eingabe-, Interpretations- oder Auswahlfehler betrachtet. Jedoch fügen Systemfehler (z.B. Sensorausfälle), technische Beschränkungen (z.B. Ressourcenknappheit) und die Datenalterung weitere Datenqualitätsverluste hinzu, die die finale Anwendungsqualität bestimmen.

Im Kontext der Sensordatenqualität ist der Fokus auf die Datenqualitätsprobleme einer Datenquelle auf Instanzebene gerichtet (siehe Abbildung 2.3). Das zu entwickelnde System wird eine beliebige Anzahl von Sensoren als Datenquellen unterstützen, jedoch werden die Ausprägungen der Datenqualitätsdimensionen für jeden einzelnen dieser Sensoren unabhängig voneinander initialisiert. Bei der Kombination mehrerer Datenströme wird keine Konsistenzprüfung stattfinden. Diese kann und soll als Teil der Datenverarbeitungskette modelliert werden. Die Korrektheit des Datenschemadesigns geht über den Rahmen dieser Arbeit hinaus und wird nicht untersucht.

Datenqualitätsdimensionen, die Nutzerzufriedenheit ausdrücken, können nicht automatisiert erfasst werden und werden daher nicht untersucht. Dimensionskandidaten für die Beschreibung von Sensordatenqualität finden sich folglich nur in den Kategorien der intrinsischen und kontextuellen Datenqualität (siehe Tabelle 2.6). Im Folgenden werden die in den vorherigen Abschnitten aufgedeckten Datenqualitätsprobleme den Datenqualitätsdimensionen dieser Kategorien zugeordnet, um Sensordatenqualität zu definieren.

Die intrinsische Dimension *Genauigkeit* beschreibt die maximale, relative oder absolute Abweichung eines Datenwertes vom realen Wert. Im Kontext von numerischen Sensordaten entspricht diese Abweichung dem *systematischen Messfehler*. Die *Glaubhaftigkeit*,

auch als Konfidenz bezeichnet, gibt das Vertrauen in einen Datenwert wieder. Der maximale *statistische Messfehler* definiert das Intervall zufälliger Messschwankungen auf Basis der Wahrscheinlichkeit, dass der wahre Wert in diesem Intervall liegt. Je größer der statistische Fehler ist, umso wahrscheinlicher (glaubhafter) liegt der wahre Wert darin. Der statistische Messfehler wird deshalb der Dimension der Konfidenz zugeordnet. Die intrinsische *Objektivität* und *Reputation* eines maschinellen Sensors wird vorausgesetzt und muss daher nicht in die Betrachtung einbezogen werden.

Die kontextuelle Datenqualität geht über die Analyse der Qualität eines einzelnen Datenwertes hinaus. Die *Vollständigkeit* trifft Aussagen über fehlende Werte einer Datengruppe und kann somit zur Beschreibung von Sensorausfällen in Sensordatenströmen herangezogen werden. Die *Datenmenge* beschreibt Datenvolumina und wird zur Definition der repräsentierten Rohdatenmenge eines berechneten Sensordatenstromtupels genutzt. Das Alter der Sensordaten wird der *Aktualität* zugeordnet, die den zeitlichen Kontext eines Datenwertes in Beziehung zum vergangenen Zeitraum seit der Datenaufnahme beurteilt. Die *Relevanz* der Daten und der *Mehrwert*, der mit ihrer Hilfe erbracht werden kann, hängen wiederum von Einschätzungen des Nutzers ab und sollen deshalb hier nicht untersucht werden.

Definition 2.2 Die Datenqualität Q eines Sensordatenstroms D wird durch die Datenqualitätsdimensionen Genauigkeit a, Konfidenz ϵ, Vollständigkeit c, Datenmenge d und Aktualität u definiert.

Zur vollständigen Begriffsklärung werden im Folgenden Definitionen dieser Datenqualitätsdimensionen vorgestellt. Zuerst werden Dimensionen zur Beschreibung eines einzelnen Datenwertes betrachtet. Danach werden die Dimensionen bestimmt, die sich auf eine Datengruppe beziehen.

Genauigkeit

Die Genauigkeit eines Sensors und somit der gemessenen Daten wird in der Güteklasse oder Genauigkeitsklasse des Sensors angegeben. Diese ist der Herstellerbeschreibung zu entnehmen und bezeichnet den maximalen absoluten Fehler als prozentualen Anteil der oberen Messbereichsgrenze. Ein Drucksensor der Güteklasse 1 mit einem Messbereich von 0 bis $200bar$ weist zum Beispiel einen absoluten Fehler von $2bar$ auf. In der Fehlerberechnung wird diese sensorinhärente Abweichung als systematischer Fehler bezeichnet.

Definition 2.3 Die Genauigkeit eines numerischen Datenwertes bezeichnet den maximalen, absoluten, systematischen Messfehler a, so dass der reale Wert $\hat{x}$ mit Sicherheit im Intervall $[x - a; x + a]$ um den gemessenen Wert x liegt.

Konfidenz

Die Konfidenz beschreibt den statistischen Messfehler, der durch zufällige Einflüsse der Umwelt (zum Beispiel Vibrationen, Temperaturschwankungen, etc.) entsteht und um den wahren Messwert streut. Das Ausmaß dieser Streuung wird in der Konfidenz festgehalten.

Definition 2.4 Die Konfidenz eines Sensordatenstroms beschreibt den maximalen, absoluten, statistischen Fehler ϵ, so dass der reale Wert $\hat{x}$ mit der Konfidenzwahrscheinlichkeit p im Intervall $[x - \epsilon; x + \epsilon]$ um den gemessenen Wert x liegt.

Die Konfidenz definiert die Grenzen der zufällig verteilten Streuung basierend auf der Standardabweichung σ der Messwerte (siehe Abschnitt 2.2). Aufgrund der statistischen Verteilung gilt $\epsilon = \infty$ für $p = 100\%$. Tabelle 2.1 auf Seite 14 definiert die Konfidenz ϵ für unterschiedliche Konfidenzwahrscheinlichkeiten $p < 100\%$. Zum Beispiel ergeben die Messwerte $\{18,7; 17,2; 21,7; 21,3; 19,8\}$ eine Konfidenz von $\epsilon = 2,58 \cdot \sigma = 1,7$ bei einer Konfidenzwahrscheinlichkeit von $p = 99\%$.

Aktualität

Es gibt zwei Interpretationen der Datenqualitätsdimension Aktualität. Zum Einen kann die Aktualität das Alter des jeweiligen Datenwertes als Differenz der aktuellen Systemzeit und des Zeitstempels der Datenerfassung ausdrücken. Zum Anderen kann die Aktualität als Rechtzeitigkeit im Hinblick auf den Anwendungskontext beurteilt werden. Nur unter Kenntnis der Anwendungsanforderungen lässt sich dabei bestimmen, ob ein Datenwert oder ein Berechnungsergebnis rechtzeitig ausgegeben wurde. Da die Bewertung der Rechtzeitigkeit eine Beurteilung bzw. die vorherige Festlegung von Regeln durch den Nutzer beinhaltet, wird die Aktualität im Rahmen dieser Arbeit als Alter eines Datenwertes interpretiert.

Definition 2.5 Die Aktualität u eines Datenwertes lässt sich als Differenz der aktuellen Systemzeit *clock* und dem Zeitpunkt der Datenaufnahme, gegeben durch den Zeitstempel t, im Datenstrom berechnen.

Die Aktualität nimmt damit unter den Datenqualitätsdimensionen eine Sonderstellung ein. Sie kann im Gegensatz zu allen anderen DQ-Dimensionen während der Laufzeit aus der Systemzeit und den Zeitstempeln des Datenstroms berechnet werden. Die Aktualität muss daher nicht im Datenstrom propagiert und verarbeitet werden, sondern wird direkt am Endpunkt der Datenverarbeitung für den Nutzer berechnet und angezeigt. Um eine vollständige Betrachtung der Datenqualität zu gewährleisten, wird sie in den weiteren Ausführungen berücksichtigt, wobei ihr Sonderstatus aber Beachtung finden muss.

Datenmenge

Die Datenmenge beschreibt den untersuchten Datenumfang. Wird im Verlauf der Datenstromverarbeitung eine Menge an d Datentupeln zusammengefasst (z.B. durch Aggregation), so beruht das Ergebnis auf der Auswertung nicht nur eines Datenwertes, sondern der Verrechnung dieser d Datentupeln. Die Anzahl der zusammengefassten Datentupel wird durch die Datenqualitätsdimension der Datenmenge widergespiegelt.

Definition 2.6 Die Datenmenge d definiert die Anzahl der Datensätze x_i für $1 \leq i \leq d$, die zur Bestimmung eines Datenwertes $x' = f(x_i)$ herangezogen wurden.

Vollständigkeit

Die Vollständigkeit adressiert fehlende Werte im Datenstrom aufgrund von Sensorfehlern oder -ausfällen. Es existieren vielfältige Strategien zur Behandlung fehlender Datenwerte beim Import in Datenbanken und Data-Warehouse-Systemen [LLLK99]. In den Extraktions-, Transformations- und Ladeprozessen (engl. Extraction Transformation Loading, ETL) wird zumeist die Schätzung oder Interpolation fehlender Werte angestrebt. Diese Strategien werden auch zur Wiederherstellung fehlender Sensordaten verwendet. Die Datenqualitätsdimension Vollständigkeit c dient zur Unterscheidung gemessener Sensordaten $x \in X$ und geschätzter bzw. interpolierter Werte $\tilde{x} \in \tilde{X}$.

Definition 2.7 Die Vollständigkeit c beschreibt den Anteil der gemessenen Datenwerte $|X|$ an der Gesamtmenge des Datenstroms $m = |X| + |\tilde{X}|$.

Im Rahmen dieser Arbeit werden die vorgestellten fünf Datenqualitätsdimensionen zur Beschreibung eines Sensordatenstroms fokussiert. Details zu deren Initialisierung werden analysiert und Methoden zur Abbildung der Datenverarbeitung werden erarbeitet. Ebenfalls wird die Integration von Nutzeranforderungen auf Basis dieser Dimensionen untersucht. Darüber hinaus besteht jedoch das Ziel, alle Konzepte so generisch wie möglich zu halten, um die transparente Erweiterung durch zusätzliche Datenqualitätsdimensionen zu ermöglichen.

2.5. Abgeleitete Anforderungen

Basierend auf den vorangegangenen Überlegungen werden nun die Anforderungen formuliert, die das zu entwickelnde System zum Management von Datenqualitätsinformationen in Sensordatenströmen erfüllen muss. Durch Analysen der Eigenschaften von Sensormessdaten und existierender Datenqualitätsdefinitionen konnten Dimensionen zur Beschreibung der Datenqualität von Sensordatenströmen extrahiert werden.

Anforderung 1 Zur Bewertung der Datenqualität von Sensordatenströmen werden die Datenqualitätsdimensionen *Genauigkeit*, *Konfidenz*, *Vollständigkeit*, *Datenvolumen* und *Aktualität* herangezogen.

In Abschnitt 2.3 wurden die Ressourcenbeschränkungen in sensorgesteuerten Anwendungssystemen diskutiert. Das Volumen der Datenqualitätsinformationen und damit des gesamten übertragenen Sensordatenstroms muss so gering wie möglich gehalten werden.

Anforderung 2 Die Datenqualitätsinformationen müssen mit geringem Speicherplatzaufwand im Datenstrom-Management-System verwaltet werden.

In einigen Anwendungen müssen die Ergebnisse der Datenstromverarbeitung in einer Datenbank abgelegt werden. So können Untersuchungen über den zeitlichen Datenverlauf erstellt und komplexe Data-Mining-Analysen ausgeführt werden. Dazu sind Datenstrukturen zur Qualitätsspeicherung sowie Methoden zum Einfügen und Verarbeiten der Datenqualität notwendig.

Anforderung 3 Das relationale Datenmodell muss erweitert werden, um Datenqualitätsinformationen persistent zu speichern und den effizienten Import der Messdaten aus dem Datenstromsystem in die Datenbank zu gewährleisten.

Wenn Sensordatenströme bzw. Datenbankrelationen verarbeitet werden, müssen die ausgeführten Operationen wie Aggregation, Verbund oder Addition auch auf den Datenqualitätsinformationen nachvollzogen werden, um die Qualität der Verarbeitungsergebnisse zu berechnen. Die Operatoren der Anfrageverarbeitung in Datenströmen sowie Datenbanken müssen analysiert werden, um ihren Einfluss auf verschiedene Datenqualitätsdimensionen zu bestimmen.

Anforderung 4 Alle in Abschnitt 2.3 aufgelisteten Operatoren der Sensordatenstromverarbeitung müssen hinsichtlich ihres Einflusses auf Datenstromtupel und Datenqualitätsinformationen untersucht werden, um Funktionen zur Datenqualitätsverarbeitung für jede Dimension aus Anforderung 1 abzuleiten.

Anforderung 5 Die Datenqualitätsverarbeitung muss ebenfalls für Datenbankanfragen ermöglicht werden, um die Qualität des Anfrageergebnisses berechnen zu können.

Wenn die Anforderungen 2 bis 5 erfüllt sind, stehen dem Nutzer Datenqualitätsinformationen sowohl im Datenstromsystem als auch in der Zieldatenbank zur Verfügung. Die Nutzer von Sensormessdaten sind typischerweise keine Informatiker, sondern Wartungsmitarbeiter, Techniker, Ingenieure, aber auch Manager oder Akademiker anderer Forschungszweige. Sie verfügen über Erfahrungen im Umgang mit gängigen Computerprogrammen. Kenntnisse der Datenbankanfragesprache SQL können jedoch nicht vorausgesetzt werden. Trotzdem soll diesen Benutzergruppen die Anfrage und Interpretation von Datenqualitätsinformationen ermöglicht werden.

Anforderung 6 Es müssen nutzerfreundliche Funktionen und Methoden zum Abfragen und Visualisieren der Datenqualitätsinformationen in Zusammenhang mit den Sensormessdaten im Datenstrom- und Datenbank-Management-System geschaffen werden.

Durch schwankende Datenraten in Sensordatenströmen kann es zu Überlastsituationen kommen, so dass nicht alle eintreffenden Sensormesswerte mit den gegebenen Ressourcen verarbeitet werden können. Um Verzögerungen der Datenverarbeitung oder Speicherüberläufen vorzubeugen, müssen überzählige Sensordatentupel aus dem Datenstrom entfernt werden. Durch diesen Informationsverlust wird die Datenqualität nachfolgender Verarbeitungsergebnisse herabgesetzt. Numerischen Messfehler werden erhöht, die Vollständigkeit des Anfrageergebnisses verringert. Um dem entgegen zu wirken und die Qualität der Anfrageergebnisse zu verbessern, müssen neue Ansätze zum Lastausgleich entwickelt werden, die diese eingeführten Fehler minimieren und eventuell kompensieren.

Anforderung 7 Es müssen Load-Shedding-Verfahren zum Ausgleich von Überlastsituationen in Sensordatenströmen entwickelt werden, die die Qualität von Verarbeitungsergebnissen auf Basis von vorhandenen Datenqualitätsinformationen optimieren.

Bei der Verarbeitung von Sensordaten kann es trotzdem vorkommen, dass die aktuelle Datenqualität der Verarbeitungsergebnisse nicht den Anforderungen der Anwendung bzw. des Nutzers genügt. In diesem Fall soll es ermöglicht werden, für jede DQ-Dimension Datenqualitätsanforderungen zu definieren, die eine nachfolgende Qualitätsverbesserung steuern.

Anforderung 8 Es müssen Möglichkeiten zur Integration und Gewährleistung von Nutzeranforderungen an die Datenqualität von Verarbeitungsergebnissen geschaffen werden. Die Datenstromverarbeitung muss konfiguriert werden, um die resultierende Datenqualität zu verbessern.

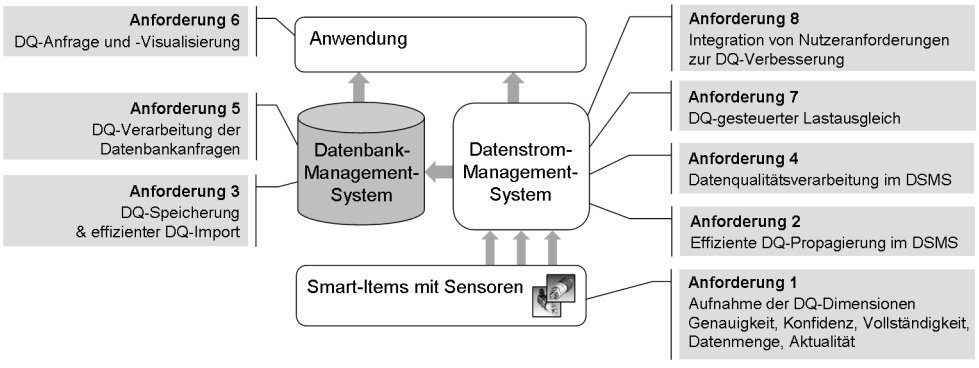


Abbildung 2.5.: Anforderungen an das Datenqualitätssystem

Abbildung 2.5 fasst die Anforderungen an das zu entwickelnde System zum Datenqualitätsmanagement in Sensordatenströmen zusammen.

In diesem Kapitel konnten die vier Fragen der Zielstellung der vorliegenden Arbeit detailliert werden. Die Konzepte und Datenstrukturen zur Erfüllung der Anforderungen 1, 2 und 3 beantworten die erste Frage der Zielstellung. Mit Erfüllung der Anforderungen 4 und 5 werden Funktionen gegeben, um Datenverarbeitungsschritte auf Qualitätsinformationen abzubilden. Damit wird das zweite Ziel dieser Arbeit erfüllt. Die dritte Frage nach Verbesserungsmöglichkeiten der Sensordatenqualität wird durch die Anforderungen 7 und 8 abgedeckt. Anforderung 6 steht im Kontext der vierten Frage nach einer nutzerfreundlichen Visualisierung der Datenqualitätsergebnisse. Diese wird durch die entwickelte Benutzeroberfläche im Zuge der prototypischen Umsetzung und Validierung beantwortet.

3
Verwandte Arbeiten

Dieses Kapitel diskutiert verwandte Arbeiten auf dem Gebiet des Datenqualitätsmanagements, um Grundlagen zur Erfüllung der aufgestellten Ziele und Anforderungen zu legen. Zunächst werden in Abschnitt 3.1 allgemeine Modelle und Methoden zur Verwaltung von Datenqualitätsinformationen vorgestellt. Abschnitt 3.2 erläutert Qualitätsaspekte in Informationssystemen. Dabei werden vor allem Data-Cleaning-Verfahren und die Integration heterogener Informationsquellen beleuchtet. Abschnitt 3.3 beschreibt den Konflikt zwischen der Dienstqualitätsoptimierung traditioneller Datenstromsysteme und der in dieser Arbeit angestrebten Datenqualitätsverbesserung. Schließlich werden verwandte Arbeiten vorgestellt, die sich speziell mit der Qualität von Sensordaten beschäftigen.

3.1. Datenqualitätsmodelle und -methoden

In Abschnitt 2.4 wurden verschiedene Definitionen des Begriffs Datenqualität gegenüber gestellt, um geeignete Dimensionen zur Beschreibung der Sensordaten abzuleiten. Im Folgenden werden die Sichten der Datenkonsumenten und Datenerzeuger auf die Datenqualität untersucht und Metriken und Methoden zur Datenqualitätsmessung vorgestellt. Anschließend werden verschiedene Modelle zur Abbildung von Datenqualitätsinformationen in einem relationalen Datenbankschema veranschaulicht. Außerdem wird die Integration von Qualitätsaspekten in semistrukturierte Datenstrukturen dargestellt und der Datenproduktionsprozess modelliert. Dieser Abschnitt endet mit einer Diskussion verschiedener Datenqualitätsalgebren, die die Verarbeitung von Qualitätsinformationen in relationalen Datenbankanfragen erlauben.

3.1.1. Verschiedene Sichten auf Datenqualität

Es gibt so viele verschiedene Sichten auf das Thema Datenqualität wie es unterschiedliche Nutzerrollen in Datenbank- und Informationssystemen gibt. Entsprechend der Arten des Datenzugriffs werden vor allem Erzeuger von Daten (schreibender Zugriff) und Konsumenten bzw. Endbenutzer der Daten (lesender Zugriff) unterschieden. Je nach Nutzerrolle treten andere Datenqualitätsprobleme auf. Im Zuge der Datenqualitätsverwaltung wird eine dritte Rolle hinzugefügt. Der Datenqualitätsmanager kontrolliert die Datenqualität und korrigiert eventuell auftretende Datenfehler, wenn dies möglich ist.

Ausgehend von der „Tauglichkeit“ bzw. „Nutzbarkeit“ der Daten wurden in Abschnitt 2.4 verschiedene Datenqualitätsdimensionen diskutiert und zu Datenqualitätsdefinitionen zusammengefasst. Im Folgenden werden weitere Arbeiten vorgestellt, die sich speziell mit der Nutzersicht auf Datenqualität befassen.

Datenkonsumenten verfolgen immer ein bestimmtes Ziel bei der Betrachtung von Informationen und deren Qualität. Genügt die Qualität einer Datenquelle einer gestellten Aufgabe, kann sie doch in einem anderen Kontext als zu gering eingestuft werden. Even et al. [ES07] schlagen deshalb die nutzengesteuerte Bewertung von Datenqualität vor, die den Kontext der jeweiligen Anwendung einbezieht. Sie präsentieren Metriken zur Berechnung der Vollständigkeit, Genauigkeit und Gültigkeit, die auf dem Maß des Geschäftswertes beruhen, der mit den Daten im spezifischen Anwendungskontext verbunden ist.

In [IOB83] werden mehrere Ansätze zur Bewertung der Nutzerzufriedenheit untersucht. Eine Befragung von Produktionsmanagern ergab die DQ-Dimensionen Zuverlässigkeit, Prädikativ- und Konzeptgültigkeit als guten Bewertungsmaßstab der Informationsqualität. Eine Formularvorlage wird vorgestellt, mit der Qualitätsbefragungen mit geringem zeitlichen Aufwand durchgeführt werden können.

In [CSBP99] wird analysiert, welche Arten von Datenqualitätsinformationen im Kontext der Entscheidungsfindung (engl. decision making) von Bedeutung sind. Im Versuch wurde festgestellt, dass Datenqualitätsinformationen im Intervallformat am besten zur Entscheidungsunterstützung geeignet sind. Es wurden keine starken Abweichungen zwischen Entscheidungen mit und ohne DQ-Information festgestellt. Doch je mehr Qualitätsdaten zur Verfügung stehen, umso konsistenter und schneller wird entschieden.

Missier et al. befassen sich in [MEG+06] mit der aktiven Bestimmung von Datenqualitätssichten (engl. quality view) der Nutzer. Die Zielgruppe bilden Forscher in der Wissenschaft und Medizin. Sie stellen ein Rahmenwerk vor, das es Datenkonsumenten ermöglicht, relevante Datenqualitätsinformationen zu definieren, die halbautomatisch in den Datenverarbeitungsprozess eingepflegt werden.

Aufgabe des *Datenerzeugers* ist es nun, die Tauglichkeit der Daten zu gewährleisten und dem Endbenutzer Daten mit hoher Datenqualität in allen gewünschten Dimensionen zur Verfügung zu stellen. In [Nau07] gibt Naumann einen Überblick über den IT-aspekt

der Datenqualität und schlägt damit eine Brücke zwischen Datenverbraucher und -erzeuger. Er stellt erst unterschiedliche Begriffsdefinitionen und Klassifikationen der Datenqualität vor und gibt dann einen kurzen Einblick in die Messung und Verbesserung von Datenqualität in Datenbanken (z.B. durch Duplikaterkennung). In [TB98] fassen Tayi et al. weitere wichtige Arbeiten auf dem Gebiet der Datenqualität unter dem Gesichtspunkt des Datenqualitätsmanagement zusammen.

Orr [Orr98] schlägt die Installation von Feedback-Kontrollsystemen zur Sicherstellung der Datenqualität in Informationssystemen vor. Er stellt damit den Nutzer als ständigen Kontrolleur der gepflegten Daten in den Vordergrund und definiert sechs Regeln, um die Datenqualität zu bestimmen und auf einem zufriedenstellenden Niveau zu halten. Der Nutzer der Daten erhält damit die Möglichkeit der Datenqualitätsverbesserung, übernimmt aber auch aktiv die Rolle des DQ-Verantwortlichen.

In [ME05] wird untersucht, wie Methoden zur Definition der Dienstqualität (engl. quality of service) zur Bestimmung von Datenqualitätsanforderungen in serviceorientierten Anwendungen genutzt werden können. Missier et al. stellen ein Modell zur Datenqualitätsabsprache zwischen Dienstanbieter (Datenerzeuger) und -nutzer (Datenverbraucher) vor und zeigen auf, wie die Qualitätsanforderungen in den Prozessablauf des Datendienstes einbezogen werden können. Sie evaluieren ihr Konzept anhand der DQ-Dimension der Vollständigkeit, die mit Hilfe einer wahren Referenztabelle berechnet wird.

In [Wan98] präsentiert Wang unter dem Begriff „Total Data Quality Management“ (TDQM) Methoden zum Datenqualitätsmanagement von Informationsprodukten. Rohdaten werden aus der Produktperspektive wahrgenommen und über Verarbeitungsketten (IP-Maps) zu Informationsprodukten transformiert. Basierend auf Grundgedanken des Manufacturing werden Methoden zur Definition, Messung, Analyse und Verbesserung von Informationsprodukten abgeleitet. Pearson et al. befragten DQ-Manager, die das TDQM anwenden, und fassten die Ergebnisse in [PMH95] zusammen. Nachdem die TDQM-Prozesse mindestens drei bis vier Jahre genutzt wurden, konnte in allen befragten Unternehmen die Produkt- und Dienstqualität und damit die Kundenzufriedenheit verbessert werden.

In sensorgestützten Anwendungssystemen übernehmen die Sensoren die Rolle der Datenerzeuger. Anwendungen bzw. Nutzer sind die Konsumenten der Daten. Außerdem übernehmen sie die Rolle des Datenqualitätsmanagers, der über Güte und Tauglichkeit der angebotenen Qualität entscheidet. Sie werden dabei durch das zu entwickelnde System zur Datenqualitätspropagierung und -verarbeitung unterstützt, das alle notwendigen Datenqualitätsinformationen zur Verfügung stellt. Des Weiteren wird die Korrektur von Datenqualitätsfehlern durch die automatische Integration von Nutzeranforderungen übernommen.

3.1.2. Modellierung der Datenqualität

Im Folgenden werden Modelle vorgestellt, die das traditionelle Metadatenmodell eines Datenbank-Management-Systems erweitern, um Datenqualitätsinformationen in Datenbanken zu verwalten. Außerdem wird die Integration von DQ-Informationen in semistrukturierte XML-Dateien diskutiert. Zum Schluss wird die IP-Map beschrieben, mit deren Hilfe der Informationsproduktionsprozess und damit mögliche Datenqualitätsbeeinträchtigungen abgebildet werden können.

Erweiterung des ER-Modells

Traditionelle Datenmodelle können auf konzeptioneller und logischer Ebene mit Strukturen zur Repräsentation und Analyse von Datenqualität angereichert werden. Storey und Wang entwerfen das „Quality ER Model“ als eine Erweiterung des Entity-Relationship-Modells [SW98], [SW01]. Auf konzeptioneller Ebene werden spezielle Qualitätsentitäten für jede DQ-Dimension eingeführt. Abbildung 3.1 zeigt den Zusammenhang zwischen Standardentitäten (*Produkt* mit dem Attribut *Preis*) und den Qualitätsentitäten (*DQ-Dimension*) mit den Attributen *Name* und *Wert*, zum Beispiel Genauigkeit=0,8. Das *DQ-Maß* interpretiert die gegebenen DQ-Werte durch Zuordnung einer *Beurteilung* (z.B. ausreichend), so dass eine Qualitätsbewertung möglich wird.

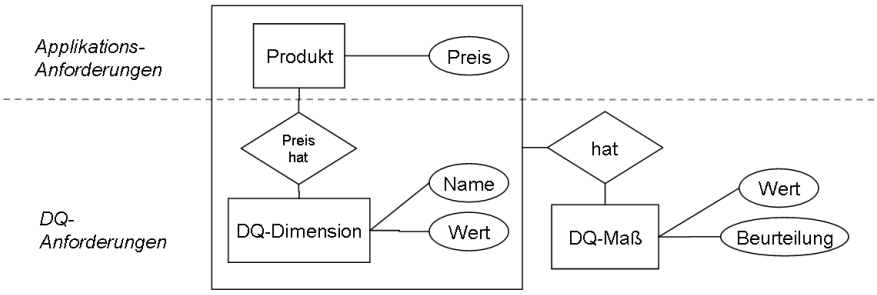


Abbildung 3.1.: Erweitertes ER-Modell

Attributbasiertes Modell

In [WSF95] wird eine Erweiterung des Relationenmodells auf logischer Ebene vorgeschlagen. Es werden zusätzliche Attribute (Qualitätsindikatoren) zur Beschreibung der Datenqualität eingeführt (siehe Abbildung 3.2). Mit Hilfe der Qualitätsindikatoren ist eine erweiterte Anfrage möglich. Zum Einen kann die Qualität der Daten abgefragt werden, zum Anderen können gespeicherte Qualitätswerte zur Selektion, Vereinigung und Differenzberechnung von Daten genutzt werden (siehe Abschnitt 3.1.3).

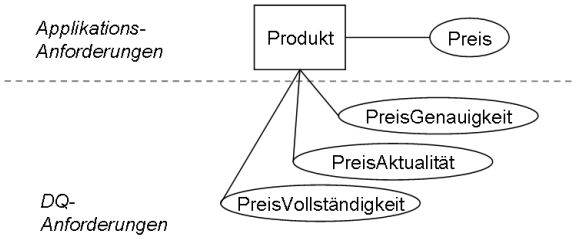


Abbildung 3.2.: Datenqualitätsattribute

Polygen-Modell

Das attributbasierte Modell kann auch zur Umsetzung des Polygen-Datenmodells [WM90] genutzt werden. Hier werden die zusätzlichen Attribute zur expliziten Repräsentation der Herkunft von Daten in verteilten, heterogenen Datenbanksystemen genutzt. Das Polygen-Relationenschema umfasst globale Polygen-Attribute, die jeweils durch die lokale Datenbank, das lokale Schema und den Namen des lokales Attributs beschrieben werden. Die Qualität eines Anfrageergebnisses wird unter der Annahme, dass zuverlässige Datenquellen zu einer hohen Datenqualität führen, durch die Herkunft der Daten definiert. In Abschnitt 3.1.3 wird die Anfrage-Algebra des Polygen-Modells näher erläutert.

D^2Q-Modell

Neben der Qualitätsbestimmung strukturierter Daten soll die Bewertung semistrukturierter Informationen möglich sein. Dafür schlagen Scannapieco et al. in [SVM+04b] ein erweitertes XML-Modell vor: D^2Q (Data and Data Quality), das im Rahmen des DaQuinCIS-Projekts entwickelt wurde. Abbildung 3.3 zeigt die Verbindung des traditionellen Datenschemas und des Qualitätsschemas mit den Qualitätstypen Genauigkeit und Vollständigkeit. Die Qualitätsknoten werden den Datenknoten über Qualitätsfunktionen zugewiesen.

IP-Map: Modell der Informationsprodukte

Im vorangegangenen Abschnitt wurde die Sicht auf Daten als Informationsprodukte beschrieben. Im Folgenden wird die IP-Map [SWZ00] zur Modellierung des Informationsproduktionsprozesses vorgestellt. Die Verarbeitung von Rohdaten zum Informationsprodukt (IP) ähnelt den Fertigungsprozessen in der Güterproduktion. Die IP-Map bietet Methoden zur graphischen Modellierung der Schritte und Abläufe des Datenproduktionsprozesses. Wichtige Komponenten der IP-Fertigung sind dabei u.a. Datenquellen, Konsumenten des Informationsproduktes, Verarbeitungsoperatoren und Entscheidun-

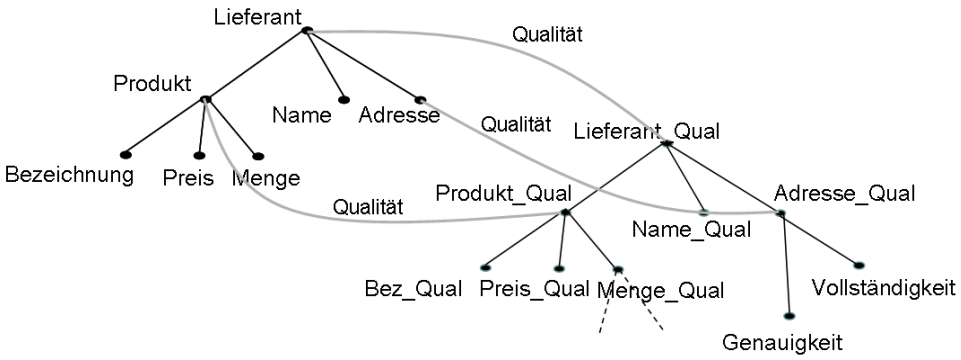


Abbildung 3.3.: Datenqualität in semistrukturierten Daten

gen. Die Handhabung von Datenqualitätsinformationen erfolgt explizit in Komponenten der Datenqualitätskontrolle. Zum Beispiel können invalide Daten aus dem Datenfluss gefiltert oder die Datenqualität eines Attributes bestimmt werden, um anschließend eine Entscheidung abzuleiten. Mit Hilfe der IP-Map können Dateneigentümer, Verantwortlichkeiten sowie potenzielle Datenqualitätsprobleme identifiziert und bearbeitet werden.

In [Pie04] wird die Erweiterung der IP-Map um Kontrollmatrizen vorgestellt. Sie verbinden Datenprobleme und Qualitätskontrollen, um erstere während des Informationsproduktionsprozesses automatisch aufzuspüren und zu korrigieren. Sie werden von Experten erstellt und dienen in aggregierter Form zur Bestimmung der Gesamtqualität der beteiligten Informationsprodukte.

Eine weitere Anwendung der IP-Map wird in [SC06] beschrieben. Shankaranarayanan und Cui stellen ein Decision-Support-System vor, dass die aktive Einschätzung der Datenqualität mit Hilfe der IP-Map-Funktionen ermöglicht. Das Werkzeug „IPView“ setzt dieses Rahmenwerk am Beispiel der Vollständigkeit um.

Schließlich wird der Ansatz des Informationsproduktes in [Arn92] aufgegriffen. Arnold formalisiert das Konzept des Information Manufacturing (dt. Informationsproduktion) als Prozess des Schreibens maschinenlesbarer Dateien bis hin zur Einarbeitung des Nutzerfeedbacks. Dabei bestimmen der Preis, die Genauigkeit, die Präsentation und die notwendigen Kompromisse (z.B. Vollständigkeit gegen Kosten) die Qualität der produzierten Informationen.

Zusammenfassung

Sowohl im erweiterten ER-Modell als auch im attributbasierten Modell werden Qualitätsinformationen auf Attributebene gespeichert. Das Datenvolumen wird um die Anzahl $\vartheta = |Q|$ der betrachteten Datenqualitätsdimensionen $q \in Q$ erhöht. Der Qualitätsknoten

im D^2Q-Modell besteht aus Unterknoten für alle betrachteten DQ-Dimensionen, so dass ein Eintrag in der ursprünglichen XML-Datei ϑ Knoten im Qualitätsschema assoziiert. In der IP-Map wird ebenfalls jedes Informationsprodukt durch Datenqualitätsdimensionen beschrieben, die in der Datenqualitätskontrolle ausgewertet werden.

In der Qualitätsverwaltung von Sensordaten wird diese Annotationsebene aufgegriffen. Jedes Datenstromattribut, d.h. jeder eingehende Sensordatenstrom, wird mit Hilfe von ϑ Dimensionen bewertet. Andere Möglichkeiten wären eine tupel-, spalten- oder relationenbasierte DQ-Beschreibung [SB04].

Die vorgestellten Ansätze haben gemein, dass die Datenqualitätsinformationen für jeden einzelnen Datenwert gespeichert bzw. betrachtet werden. Datenstromsysteme arbeiten in der Regel unter stark eingeschränkten Ressourcen, so dass das Datenvolumen begrenzt werden muss. Es ist nicht ohne großen Zusatzaufwand möglich jedem Messwert ϑ DQ-Informationen zur Seite zu stellen. Kapitel 4 widmet sich deshalb der effizienten Verwaltung von Datenqualitätsinformationen in Datenstromsystemen und Datenbanken.

3.1.3. Verarbeitung von Qualitätsinformationen

In diesem Abschnitt werden verschiedene Ansätze diskutiert, die eine Datenqualitätsalgebra für relationale Datenbankoperatoren definieren. Sie fokussieren unterschiedliche DQ-Dimensionen und Operatorenklassen.

Attribut-basiertes Modell

Neben der Erweiterung der relationalen Datenstrukturen durch Datenqualitätsattribute, präsentieren Wang et al. in [WSF95] eine Algebra für die gespeicherten DQ-Indikatoren. Sie beschreiben die Selektion und Projektion über den Datenqualitätsattributen. Bei einer Selektion von Datentupeln werden automatisch die entsprechenden DQ-Indikatoren selektiert. Die Projektion auf eine Menge von Attributspalten wird auf die diese Spalten beschreibenden DQ-Indikatoren erweitert. Diese Operatoren können auch umgekehrt verwendet werden. Das heißt, eine Selektion bzw. Projektion von DQ-Indikatoren führt zur zusätzlichen Selektion bzw. Projektion der entsprechenden Datentupel. Außerdem werden Relationenvereinigung und -differenz auf den DQ-Attributen nachvollzogen, wenn diese mengenkompatibel sind.

Diese Datenqualitätsalgebra erlaubt das Zusammenführen mehrerer Relationen sowie die Auswahl von Teilmengen der Daten. Sie ermöglicht allerdings keine weitergehende Verarbeitung der Rohdaten, so dass keine höherwertigen Informationen berechnet werden können.

Polygen-Modell

Das Polygen-Modell unterstützt das Data-Lineage (dt. Verfolgung der Datenabstammung) auf Schemaebene. Mit dem Wissen über „schlechte“ Datenquellen kann die Qualität der abgeleiteten Daten oder Informationen bewertet werden. In [WM90] werden dem attributbasierten Modell ähnliche Operatoren: Selektion, Projektion, Vereinigung, Differenz und kartesisches Produkt. Außerdem schlagen sie einen zusätzlichen Coalesce-Operator vor, der zwei relationale Spalten vereinigt.

Die Verfolgung der Herkunft von Daten auf Instanzebene gestaltet sich schwieriger. Transformationen der Daten, wie zum Beispiel Datenbereinigungen, ETL-Prozesse oder Aggregationen, müssen bei der Qualitätsberechnung berücksichtigt werden. Cui und Widom legen in [CW01] Transformationsklassen fest (Dispatcher, Aggregator, Black-Box), für die sie Tracing-Algorithmen aufstellen, mit deren Hilfe Sequenzen von Transformationen ausgeführt werden können. In [GFS+01] werden ähnliche Tracing-Algorithmen für Klassen von Data-Cleaning-Operatoren erläutert.

Beim Data-Lineage werden keine direkten Datenqualitätsinformationen verarbeitet, sondern die Herkunft und die Verarbeitungswege der Daten ermittelt. Diese Informationen werden dem Nutzer zur Verfügung gestellt, um ihm die Bewertung der Qualität, zum Beispiel der Glaubwürdigkeit oder der Aktualität der Daten zu ermöglichen. Das Data-Lineage bietet also eine gute aber indirekte Unterstützung für die Evaluation der Daten. Eine Datenqualitätsalgebra wurde jedoch nicht erarbeitet.

Berechnung der Genauigkeit

Eine ausführliche Betrachtung der DQ-Dimension Genauigkeit erfolgt in [WZL01]. Es werden sowohl gleichförmig verteilte Fehler, als auch Best- und Worst-Case-Fälle für ungleichförmige Fehlerverteilungen untersucht. Die vorgestellte Datenqualitätsalgebra umfasst die relationale Selektion und Projektion.

In [PSJ99] wird die DQ-Dimension Genauigkeit aufgegriffen und um zwei Dimensionen erweitert. Ein Datentupel ist *ungenau*, sobald ein Attributwert nicht dem wahren Wert entspricht. Fehlt ein Attributwert, ist es *unvollständig*. Es wird als *falsches Element* (engl. mismember) bezeichnet, wenn es keine Entsprechung der Attributbelegungen in der Realität gibt. In Sensordatenströmen entsprechen die Attributbelegungen Messwerten aus kontinuierlichen Messbereichen. Es sind keine Messwerte außerhalb des vom Sensor vorgegebenen Messbereiches möglich, so dass keine „Mismember“ in Sensordatenströmen entstehen können.

In [PSJ04] erstellen Parssian et al. Metriken zur Berechnung der oben genannten DQ-Dimensionen für die relationalen Operatoren Selektion, Projektion und kartesisches Produkt. In [PSJ02] wird der relationale Verbund-Operator diskutiert. In beiden Arbeiten gehen sie von einer gleichförmigen Fehlerverteilung aus, bei der die Attribute unabhängig voneinander verfälscht werden.

In [Par06] wird eine DQ-Algebra für die Aggregationen *Count*, *Sum*, *Average*, *Max* und *Min* und die Datenqualitätsdimensionen Vollständigkeit und Genauigkeit vorgestellt. Mit Hilfe von Stichproben werden Maximum-Likelihood-Schätzungen für jeden definierten Attributdatentyp bestimmt. Auf dieser Basis können die Auswirkungen unterschiedlicher Fehlerraten auf das Aggregationsergebnis vorhergesagt werden.

In allen vorgestellten Arbeiten wird die Genauigkeit mit Hilfe einer Referenztabelle berechnet, die die wahren Datenwerte enthält. Zuweilen ist auch die Hochrechnung auf Basis einer Stichprobe möglich. Ein Referenzsensor, der vollständig fehlerfrei misst, existiert jedoch nicht. Außerdem müssten sämtliche Umwelteinflüsse vermieden werden, dies ist in realen Industrieanwendungen nicht möglich. Deshalb wird die Genauigkeitsabschätzung mit Hilfe eines hochpräzisen Referenzsensors durch den Sensorhersteller vorgenommen. Die Ergebnisse werden als Fehlerkurve oder -klasse angegeben (siehe Abschnitt 2.2).

In Sensordatenanwendungen kann die Genauigkeit also nicht direkt berechnet, sondern nur auf Basis von Referenzmessungen abgeschätzt werden. Auf diese Weise wird nur die Genauigkeit der gemessenen Rohdatenströme bestimmt. Für die Berechnung der Qualität eines Verarbeitungsergebnisses fehlt die Referenz. Hierfür müssen andere Methoden entwickelt werden.

Berechnung der Vollständigkeit

In [SB04] wird die Berechnung der Vollständigkeit von relationalen Datenbanken analysiert. Scannapieco und Batini konzentrieren sich auf die Auswirkungen der relationalen Operatoren Vereinigung, Differenz und kartesisches Produkt. Die angegebenen Formeln sind allerdings nur unter der Annahme eines offenen Systems ohne fehlende Werte (Null-Werte) gültig und das Verhältnis der Eingangsrelationen (Disjunktheit, Überlappung, etc.) muss bekannt sein. In Sensordatenströmen sind die vorgeschlagenen Methoden nicht anwendbar, da durch Sensorausfälle Null-Werte entstehen können. Außerdem sind die vorgestellten Mengenoperatoren nicht ausreichend, um höherwertiges Wissen zu extrahieren.

Motro und Rakov [MR97] konzentrieren sich auf die DQ-Dimensionen *Stichhaltigkeit* (engl. soundness) und *Vollständigkeit* (engl. completeness). Zur initialen Belegung dieser DQ-Dimensionen wird eine Referenzdatenbank benötigt, die die wahren Werte enthält. In [MR98] stellen sie ein Maß der Homogenität vor, um die Gesamtqualität mit Hilfe von Stichproben schätzen zu können.

Die *Stichhaltigkeit* zählt die Tupelwerte, die vom wahren Wert abweichen. Da in realen Messungen immer Messfehler auftreten, weichen Sensormessdaten immer vom wahren Wert ab. Die Stichhaltigkeit ist somit keine repräsentative Datenqualitätsdimension in Datenstromanwendungen. Die DQ-Dimension *Vollständigkeit* ist auf Sensorströme übertragbar. Allerdings wird wieder eine Stichprobe der wahren Werte in einer Referenzdatenbank benötigt, um die Vollständigkeit zu berechnen. Diese ist in einer Anwendung

mit Sensordatenströmen nicht vorhanden. Als Referenz kann nur ein Sensor dienen, der ohne Ausfälle arbeitet. Doch Sensoren sind hochempfindliche Geräte, so dass Sensorausfälle in den angestrebten Anwendungsbereichen der Industrieproduktion und in mobilen Geräten aller Voraussicht nach nicht vermeidbar sind.

In [BCSW06] werden ebenfalls Stichproben genutzt, um die Qualität eines Anfrageergebnisses abzuschätzen. Ballou definiert die Vollständigkeit sowie die Akzeptanz bzw. Eignung der Daten für die Anfragebeantwortung, die sich aus Genauigkeit, Aktualität und Konsistenz zusammensetzt.

Diese stichprobenbasierten Arbeiten geben jeweils eine einfache Datenqualitätsalgebra an. Komplexe Datenbankanfragen werden als Kette von Basisoperationen modelliert, die um Funktionen zur Berechnung der Datenqualität erweitert werden. So kann die Qualität des Anfrageergebnisses parallel zur eigentlichen Datenbankanfrage berechnet und ausgegeben werden. Dieser Ansatz wird in Kapitel 5 aufgegriffen, um Verarbeitungsprozesse im Datenstromsystem zu modellieren. Motrov et al. diskutieren die DQ-Berechnung für das kartesische Produkt und eine Kombination aus Projektion und Selektion. Ballou fügen Relationenverbund, Vereinigung und Mengendifferenz hinzu. Um die Datenqualität in komplexen Datenstromverarbeitungen verfolgen zu können, muss auch diese DQ-Algebra umfassend erweitert werden.

In [NFL04] wird die Vollständigkeit von integrierten Informationsquellen betrachtet. In Web-Anwendungen werden verschiedene heterogene Datenquellen unterschiedlicher Güte kombiniert. Naumann et al. präsentieren drei neue Operatoren zur Datenintegration sowie ein Modell der Vollständigkeit der integrierten Antworten. Die *Vollständigkeit* wird mit Hilfe der *Dichte* (engl. density) und der *Abdeckung* (engl. coverage) bestimmt. Die Datenqualitätsalgebra bestimmt die Dichte einer Datenquelle als den Anteil der Nicht-Null-Werte am Datenvolumen dieser Quelle. Dies entspricht der Sensordatenvollständigkeit. Die Abdeckung setzt das Volumen einer Datenquelle in Bezug zur Größe der Universalrelation, die die Kombination aller Quellen beschreibt. Da Sensordatenreihen unabhängig voneinander betrachtet werden, spielt dieses Maß zunächst keine Rolle. Erst während der Datenverarbeitung werden Sensordaten verknüpft.

Die Berechnung der Vollständigkeit der integrierten Datenmenge wird für disjunkte und überlappende Datenquellen angegeben. Sie kann zum Beispiel in sicherheitsrelevanten Anwendungen genutzt werden, in denen redundante Messungen etwaige Sensorausfälle kompensieren. Kritische Messgrößen, wie zum Beispiel Gaskonzentration in der Luft, werden mit mindestens drei Sensoren überwacht. Um die resultierenden Datenströme zu verknüpfen, können die von Naumann et al. vorgeschlagenen Mengen-Integrations-Operatoren genutzt werden.

[BP03] bietet einen weiteren Ansatz zur Integration von Informationen über deren Vollständigkeit. Mit Hilfe einer Gewichtung der konkurrierenden DQ-Dimensionen Voll-

ständigkeit und Konsistenz[1] soll der Datennutzer die Auswahl von Datenquellen, die zur Beantwortung einer Anfrage genutzt werden, steuern. Dabei wird die Vollständigkeit nicht nur durch eine Zählung der Null-Werte in einer Relation definiert. Es werden weitere kontextbezogene Informationen hinzugezogen, um die *inhaltliche Vollständigkeit* zu bestimmen. Zum Beispiel wird die Vollständigkeit der Aussage „30°C“ höher eingeschätzt als die der Aussage „heiß“. Ballou erörtert jedoch nur die Quellenpriorisierung auf Basis der Gewichtung und gibt keine Datenqualitätsalgebra zur Beschreibung des Anfrageergebnisses an.

Zusammenfassung

Tabelle 3.1 fasst die Arbeiten zum Thema Datenqualitätsverarbeitung in relationalen Datenbanken zusammen. Es werden Datenqualitätsalgebren für die Dimensionen Vollständigkeit und Genauigkeit vorgeschlagen. Dabei werden vor allem Mengenoperatoren betrachtet, mit deren Hilfe spezifische Datenbereiche ausgewählt werden können.

In der Verarbeitung von Sensordatenströmen spielen diese Operatoren eine große Rolle. Mengenoperatoren dienen zur Auswahl wichtiger Tupelmengen. Mittels Projektion und Selektion ist eine Auswahl von relevanten Messwerten möglich. Der Verbund dient zum Zusammenführen mehrerer Sensordatenströme. Sie sind wichtige Bestandteile der Continuous Query Language zur Verarbeitung von Datenströmen und müssen in der zu entwickelnden Datenqualitätsalgebra vertreten sein.

Die Überlegungen zu Mengenoperatoren und Projektion (ohne Duplikateliminierung) können ohne Änderungen auf Sensordaten übertragen werden. Vereinigung, Differenz und Projektion eines Messwertattributes müssen auf alle Datenqualitätsinformationen erweitert werden (siehe Abschnitt 5.4.1 und 5.4.2).

Die bisherigen Methoden zur Berechnung der Vollständigkeit und Genauigkeit von Aggregation, Selektion und Verbund sind nicht auf Sensordatenströme anwendbar, da sie eine Referenztabelle mit den wahren Datenwerten voraussetzen. Aggregation und Selektion von Sensordatentupeln werden in den Abschnitten 5.4.3 bzw. 5.4.5 eingeführt. Die Datenqualitätsberechnung beim Datenstromverbund wird in Abschnitt 5.4.4 näher erläutert.

Die Qualitätsverarbeitung bei der Datenintegration wird in Tabelle 3.2 zusammengefasst. Auch hier spielt die Vollständigkeit eine wichtige Rolle. Des Weiteren können die Herkunftsinformationen des Data-Lineage zur indirekten Qualitätsbewertung herangezogen werden. Die qualitätsgesteuerte Datenintegration und das Data-Cleaning wurden zur Vereinigung und Bereinigung hochvolumiger, persistenter Datenquellen entwickelt. In sensorgestützten Anwendungen kommen sie nicht zum Einsatz, da sie einen sehr hohen Ressourcenbedarf haben und die Verarbeitungszeit erheblich verlängern würden.

[1] Vollständigkeit und Konfidenz stehen in Konflikt zueinander. Je mehr Datenquellen zur Beantwortung einer Anfrage hinzugezogen werden, umso höher ist die Wahrscheinlichkeit eines vollständigen Ergebnisses, doch umso höher ist auch die Wahrscheinlichkeit von Inkonsistenzen.

Referenz	DQ-Dimensionen	Operatoren
Wang [WSF95]		Selektion, Projektion, Vereinigung, Differenz
Wang [WM90]	Data-Lineage	Selektion, Projektion, Vereinigung, Differenz, Kartesisches Produkt, Spaltenverbund
Wang [WZL01]	Genauigkeit	Selektion, Projektion
Parssian [PSJ99] [PSJ02] [PSJ04]	(Un-)Genauigkeit, (Un-)Vollständigkeit	Selektion, Projektion, Kartesisches Produkt, Verbund
Parssian [Par06]	Genauigkeit, Vollständigkeit	Aggregation
Scannapieco [SB04]	Vollständigkeit	Vereinigung, Differenz, Kartesisches Produkt
Motro [MR98] [MR97] [MR97]	Vollständigkeit	Selektion, Projektion, Kartesisches Produkt
Ballou [BCSW06]	Vollständigkeit, Akzeptanz	Selektion, Projektion, Vereinigung, Differenz Kartesisches Produkt Verbund

Tabelle 3.1.: Datenqualitätsverarbeitung im relationalen Modell

Das Data-Cleaning und die qualitätsgesteuerte Integration heterogener Daten bilden jedoch große Forschungsthemen auf dem Gebiet der Datenqualität und werden daher im folgenden Abschnitt beschrieben.

3.2. Datenqualität in Informationssystemen

In [Red98] beschreibt Redman den negativen Einfluss fehlerhafter Daten in Informationssystemen auf betriebswirtschaftliche Unternehmen, um das Bewusstsein für schlechte Datenqualität zu erhöhen. Inkorrekte und inkonsistente Daten sowie das Fehlen von Daten für bestimmte Aufgaben oder Entscheidungen sind für ihn für schlechte Datenqualität verantwortlich. Er dokumentiert die Auswirkungen auf das tägliche Firmengeschäft (steigende Kosten, sinkende Kundenzufriedenheit), die taktischen Entscheidungen (langwierig, fehlerhaft) und die langfristige Unternehmensstrategie (Schwierigkeiten bei der Strategieplanung und - ausführung, Ablenkung der Aufmerksamkeit des Managements).

Referenz	DQ-Dimensionen	Operatoren
Data-Warehouse-Systeme		
Cui [CW01]	Data-Lineage	Dispatcher, Aggregator, Black-Box
Garhardas [GFS+01]	Data-Lineage	Data-Cleaning
Integrierte Informationssysteme		
Naumann [NFL04]	Vollständigkeit, Dichte, Überdeckung	Verbund
Ballou [BP03]	Vollständigkeit, Konsistenz	Quellenpriorisierung, Merging

Tabelle 3.2.: Qualitätsverarbeitung bei der Datenintegration

Nachfolgend werden Informationssysteme klassifiziert und das Data-Cleaning in Data-Warehouse-Systemen sowie die Online-Integration heterogener Datenquellen näher untersucht.

3.2.1. Klassifizierung von Informationssystemen

Um die Behandlung von Datenqualität in Informationssystemen (IS) detailliert diskutieren zu können, ist eine Klassifizierung der Typen von Informationssystemen notwendig. Für jede Systemart sind andere Qualitätsprobleme und -methoden von Bedeutung. Abbildung 3.4 zeigt die Klassifikation nach Verteilungsgrad, Heterogenität und Autonomie.

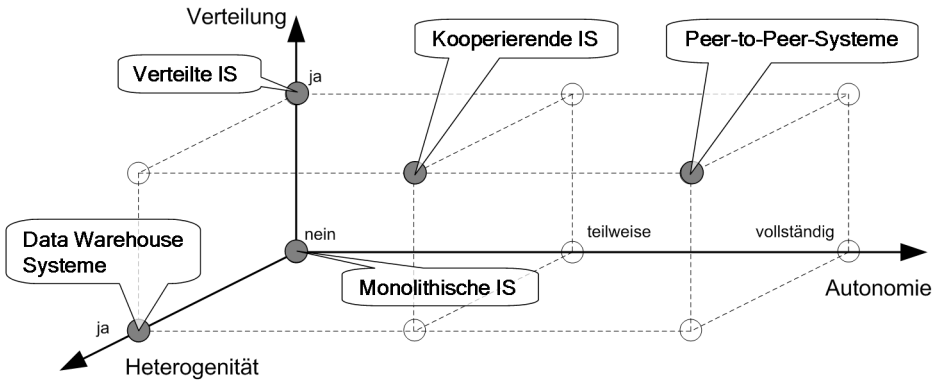


Abbildung 3.4.: Arten von Informationssystemen [BS06]

In *monolithischen Systemen* ist die Datenhaltung und Anwendungslogik auf einem zentralen Rechnerknoten installiert. Es gibt nur eine Datenquelle, wodurch die Behandlung von DQ-Problemen vereinfacht wird. Die Modellierung und Anfrageverarbeitung von Datenqualitätsinformationen, die in Abschnitt 3.1 diskutiert wurden, liegen im Vordergrund. *Verteilte Informationssysteme* [RW95] arbeiten ebenfalls auf einer homogenen Datenbasis. Die örtliche Verteilung der Daten erschwert das Datenmanagement und die Anfragebearbeitung, beeinflusst jedoch nicht die prinzipielle Behandlung von Datenqualitätsinformationen.

Data-Warehouse-Systeme (DWH) sind zentrale Datensammlungen, deren Inhalt sich aus Daten unterschiedlicher Quellen zusammensetzt [Leh03]. Sie werden meist zur Unterstützung von Geschäftsentscheidungen auf Management-Ebene eingesetzt. Das aus Datenqualitätssicht größte Problem bildet die Zusammenführung heterogener Datenquellen in der Vorverarbeitungsphase. Um konsistente und qualitativ hochwertige Daten zu erzeugen, ist ein Data-Cleaning (siehe Abschnitt 3.2.3) notwendig. Das Online-Analytical-Processing (OLAP) wird zur Analyse von Warehouse Daten genutzt. In [BDJ+05] und [BDJ+06] werden traditionelle OLAP-Operatoren erweitert, um die Verarbeitung unsicherer und unpräziser Daten zu ermöglichen.

In *kooperierenden Informationssystemen* [MDJ+97] werden viele Datenquellen unterschiedlicher Herkunft und Struktur zur Beantwortung einer Anfrage verbunden. Im Gegensatz zu Data-Warehouse-Systemen erfolgt die Quellenauswahl und -vereinigung und damit die Datenbereinigung hier zur Laufzeit der Anfrageverarbeitung. Dabei kann die Datenqualität von der Wahl der „besten“ Datenquelle zur Anfragebeantwortung profitieren. Andererseits besteht die Gefahr, dass die Qualität durch die eingeschränkten Kontrollmechanismen in autonomen und heterogenen kooperierenden Informationssystemen schnell nachlässt. Außerdem stellt die Integration vieler heterogener, zum Teil inkonsistenter Datenquellen eine sehr komplexe Aufgabe dar. Abschnitt 3.2.4 widmet sich diesem Problem.

3.2.2. Diskussion allgemeiner Ansätze

Firmendaten, d.h. Informationen, die angelegt, genutzt und geteilt werden um Geschäftsprozesse zu steuern, sind kritische Geschäftsressourcen, deren Qualität analysiert und kontrolliert werden muss. In diesem Abschnitt wird eine Auswahl an Arbeiten vorgestellt, die umfassende Systeme zur Datenqualitätsverwaltung und -verbesserung in Informationssystemen beschreiben.

Redman beleuchtet in [Red97] alle Aspekte des Datenqualitätsmanagements in Informationssystemen. Datenqualitätsprobleme und deren Auswirkungen werden analysiert, mögliche Lösungen vorgestellt und beschrieben, wie sie zu einem vollständigen Programm zur Datenqualitätsverbesserung kombiniert werden können. Um die Qualität über den gesamten Datenlebenszyklus zu gewährleisten, werden Rollen und Verantwortlichkeiten, Hauptprobleme und -aktivitäten vorgestellt.

In [UKN+99] werden die Ergebnisse einer Datenqualitätsstudie im Umfeld der Telekommunikationsindustrie vorgestellt. Es wird gezeigt, wie Datenqualitätsanforderungen definiert, wie die Datenqualität gemessen, welche kommerziellen Werkzeuge zur DQ-Verbesserung genutzt und welche Erfolge erzielt werden. Basierend auf den gesammelten Erfahrungen wird ein Rahmenwerk für das Datenqualitätsmanagement vorgeschlagen, das die auszuführenden Schritte und Prozeduren sowie Checklisten und DQ-Werkzeuge umfasst.

Kahn et al. arbeiten ebenfalls mit Befragungen von Datenerzeugern und -konsumenten [KSW02]. Sie betrachten die Datenqualitätsdimensionen Nutzbarkeit, Stichhaltigkeit, Nützlichkeit und Vertrauenswürdigkeit, um Benchmarks zu definieren. Damit kann die Datenqualität in verschiedenen Informationssystemen gemessen werden, um die jeweils angewendeten DQ-Werkzeuge zu vergleichen. In [KKPP98] präsentieren sie ein Decision-Support-System, das Datenqualitätsverantwortlichen die Planung und Kontrolle des Datenqualitätsmanagements erleichtern soll. Mit Hilfe des Systems können Testläufe zur DQ-Messung definiert und der minimale Satz an Kontrollprozeduren zur Sicherung der Zuverlässigkeit von Daten bestimmt werden.

Wand und Wang [WW96] definieren Ontologien, um Informationssysteme zu beschreiben und deren Datenqualität als Differenz zwischen gespeicherter Datenrepräsentation und der Realwelt zu bestimmen. Sie beschränken ihre Betrachtungen allerdings auf das Messen der Datenqualität. Es werden keine Verbesserungsvorschläge gegeben.

In [LSKW02] wird die „Assessment and Improvement Methodology“ vorgestellt, mit der Informationsqualität erhoben und verglichen werden kann. Neben dem Modell der Informationsqualität wird ein Fragebogen zur Messung der Qualität entwickelt. Außerdem werden Analysetechniken zur Interpretation der Informationsqualität erläutert, um Defizite aufzudecken und die besten Angriffspunkte für Verbesserungsaktivitäten aufzufinden.

Benedikt et al. vereinen Data-Cleaning-Techniken, in denen hoch-qualitative Daten zur Fehlerkorrektur genutzt werden, mit der qualitätsgesteuerten Datenintegration, bei der Qualitätsinformationen zur Quellenauswahl beitragen. Sie entwerfen ein Modell der Quellenzuverlässigkeit, das mit Hilfe der Qualitätsinformationen aktualisiert wird, wenn Daten zur Anfragebearbeitung aus den verschiedenen Quellen ausgelesen werden. So werden dem Datenkonsumenten Wahrscheinlichkeitsmodelle über unsichere Daten zur Verfügung gestellt, mit deren Hilfe er die beste(n) Quelle(n) für eine Anfrage auswählen und damit die Qualität des Anfrageergebnisses verbessern kann. Die automatische Integration unterschiedlicher Datenquellen auf Basis von DQ-Informationen wird in Abschnitt 3.2.4 untersucht.

Die vorgestellten Systeme unterstützen Datenerzeuger bei der Aufgabe des Datenqualitätsmanagements. Die Datenkonsumenten werden vom Datenqualitätsmanager mit Hilfe von Fragebögen in den Prozess der Datensicherung einbezogen. Entsprechend ihrer Bewertung werden Prozesse und Aktionen des Data-Cleaning zur Datenqualitätsverbesserung geplant.

3.2.3. Data-Cleaning

Das Data-Cleaning (dt. Datenbereinigung) dient zum Erkennen und Beseitigen von Inkonsistenzen, Widersprüchen und anderen Fehlern in Daten mit dem Ziel der Qualitätsverbesserung. Es ist Teil des ETL-Prozesses in Data-Warehouse-Systemen und nimmt bis zu 80% des Aufwandes ein. Data-Cleaning-Schritte sind unter anderem die Schematransformation zur Standardisierung und Normalisierung, die Fehlerkorrektur auf Instanzebene, Duplikat- und Ausreißererkennung sowie das Sampling bzw. Zusammenfassen von Datenquellen. Datentypen und Wertebereiche, das Vorkommen von Nullwerten, Eindeutigkeiten und Muster (z.B. Datumsformate) werden untersucht. Außerdem werden Abhängigkeiten zwischen Attributen einer Relation (z.B. funktionale Abhängigkeiten) sowie potenzielle Primärschlüssel geprüft und Überlappungen zwischen Attributen verschiedener Relationen auf Redundanzen oder Fremdschlüsselbeziehungen getestet.

Obwohl bereits 1988 erschienen, bieten das „Quality Control Handbook“ [Jur88] und die Sammlung „Data Quality Control Theory and Pragmatics“ [LU90] gute Grundlagen für das Datenqualitätsmanagement in Data-Warehouse-Systemen. In [Jur88] werden drei Phasen des Qualitätsmanagements unterschieden. In der Phase der Qualitätsplanung werden die Datenstrukturen und -importfunktionen basierend auf den gesammelten Nutzeranforderungen entworfen. In der Datenqualitätsverbesserungsphase werden die Prozesse implementiert und optimiert. In der Phase der Qualitätskontrolle muss bewiesen werden, dass die Prozesse die Nutzeranforderungen unter realen Arbeitsbedingungen erfüllen.

Ballou et al. entwerfen in [BT99] ein Konzept, um DQ-Verbesserungsmaßnahmen zu bewerten und zu planen. Basierend auf den Aktivitäten, die durch das Data-Warehouse unterstützt werden, und den zugrunde liegenden Daten wird die aktuelle und die erforderliche Datenqualität für jede relevante Datenqualitätsdimension berechnet. Anschließend wird für jede DQ-Verbesserungsmaßnahme der Kostenaufwand bestimmt und der Nutzen abgeschätzt. Damit können die Maßnahmen priorisiert und ein Projektplan zur Verbesserung der Datenqualität erstellt werden.

Im Folgenden werden weitere Arbeiten auf dem Gebiet des Data-Cleaning diskutiert. Zuerst wird die Integritätsanalyse untersucht. Danach werden verschiedene Verfahren zum Erkennen von Ausreißern und abweichenden Datenmustern beschrieben. Anschließend wird auf die Modellierung von Datenqualitätsinformationen in Data-Warehouse-Systemen zur weiteren DQ-Verarbeitung und -Verbesserung eingegangen. Ein allgemeiner Überblick über aktuelle Verfahren findet sich in [JIM00].

Integritätsanalyse und Duplikaterkennung

In [Sva88] wird die Integritätsanalyse zum Auffinden und Korrigieren von Inkonsistenzen in sieben Schritte geteilt. Nachdem Konsistenz- bzw. Integritätsregeln definiert wurden, wird eine Stichprobe der Datenbank erstellt. Dann wird die spezifische Inte-

gritätsanalyse ausgewählt und passende Datenqualitätsdimensionen definiert, die die Ausgabewerte der Datenanalyse beschreiben. Schließlich wird das Analyseprogramm implementiert und die Integritätsanalyse ausgeführt. Eine spezifische Integritätsanalyse wird zum Beispiel in [RLSG07] erläutert. Ziel ist es, so wenig Datenfelder wie möglich zu ändern, um eine konsistente Sicht auf die Daten zu erreichen. Dieses Optimierungsproblem (Minimalproblem) wird mit Hilfe eines Branch-and-cut-Ansatzes gelöst, der auf Benders Dekomposition [Geo72] beruht.

In [LLLK99] werden ebenfalls Methoden zur Integritätsanalyse vorgestellt. Es sollen Datentupel erkannt werden, die dieselbe Entität in der realen Welt referenzieren. Um Duplikate bei der Integration unabhängiger Datenquellen zu erkennen und zu eliminieren, werden die Datensätze vorverarbeitet und sortiert, so dass sich wahrscheinliche Duplikate in nächster Nachbarschaft zu einander befinden und verschmolzen werden können. Weitere Ansätze zur Duplikaterkennung werden u.a. in [EIV07] und [HS98] beschrieben.

Da Sensoren und deren Installation sehr teuer sind, wird in realen Anwendungen auf doppelte Messungen oft verzichtet. Allein in Anwendungen, die Sensorinformationen für sicherheitsrelevante Aufgaben einsetzen, werden mehrere Sensoren zur Überwachung desselben Messwertes genutzt. Multiple Messungen einer Variable können relativ einfach über den Durchschnitt aller Datentupel des gleichen Zeitstempels zusammengefasst werden. Komplexe Verfahren zur Duplikaterkennung sind nicht notwendig.

Ausreißererkennung

Kübert et al. [KGH05] stellen ein Verfahren zur regelbasierten Ausreißersuche in großen Datenbanken vor, das sowohl mit von Experten vorgegebenen Gültigkeitsregeln (Geschäftsregeln) als auch mit automatisch erzeugten Assoziationsregeln eingesetzt werden kann.

Data-Mining-Methoden werden zur Wissensgewinnung aus großen Datensammlungen genutzt. Diese Methoden können auch zur Messung und Verbesserung der Datenqualität herangezogen werden. In [VdP02] werden Data-Mining-Techniken vorgestellt, die zur Merkmalsextraktion, Fallselektion und Ausreißererkennung eingesetzt werden können, um die Datenqualität zu verbessern.

Lübbers stellt in [LGJ03] ein Datenprüfsystem vor, das Techniken des Maschinenlernens nutzt, um Ausreißer, die nicht den semantischen Strukturen der Daten entsprechen, als fehlerhafte Daten zu bestimmen und zu korrigieren. Ein ähnliches Modell wird in [GH01b] präsentiert. Auch Hinrichs nutzt in [GH01a] Data-Mining-Methoden zur Ausreißererkennung und stellt ein Prozessmodell zum Datenqualitätsmanagement vor, das den Anforderungen des ISO-9001-Standards genügt.

Die genannten Verfahren identifizieren Ausreißer oder ungewöhnliche Wertkombinationen und -muster und stufen sie als Fehler ein. Allerdings können Ausreißer auch richtige und wichtige Datensätze sein, die auf außergewöhnliche Ereignisse oder beson-

ders interessante Vorkommnisse hinweisen. Deshalb ist immer die manuelle Kontrolle der Daten durch einen Experten notwendig. Sonst wird die Datenqualität durch das Löschen der Ausreißer verschlechtert. Wie bereits erwähnt, ist die manuelle Kontrolle von Sensordaten nicht möglich. Außerdem ist es selbst für Experten schwer, zum Beispiel einen Sprung in der Motoröltemperatur einem Messfehler oder einem Motorfehler zuzuordnen. In Sensordatenströmen müssen andere Methoden zur Messung und Verbesserung der Datenqualität gefunden werden.

DQ-Modellierung in Data-Warehouse-Systemen

Im Zuge des „Data-Warehouse-Quality"-Projekts werden Grundlagen für die Datenqualitätsverwaltung in Data-Warehouse-Systemen gelegt. Semantische Modelle der Data-Warehouse-Architektur werden mit expliziten Datenqualitätsmodellen verknüpft. Ein Überblick über die Projektziele, das Architekturkonzept und die unterschiedlichen Forschungsbeiträge wird in [JV97] gegeben.

Das traditionelle Data-Warehouse-Metadatenmodell besitzt keine Komponenten, um Aspekte der Datenqualität auszudrücken. In [JJQV98] werden die Metadaten durch Geschäftsmodelle zur Abbildung von DQ-Informationen erweitert. Außerdem werden existierende Funktionen zur Messung und Verbesserung der Datenqualität in Data-Warehouse-Systemen mit Hilfe der Goal-Question-Metric [VB99], [BJ06] aus dem Software-Qualitätsmanagement in einem generischen, konzeptionellen Rahmenwerk zusammengefasst. In [CPPS01] wird erklärt, wie Datenqualitätsaspekte bereits im Data-Warehouse-Entwurf einfließen können. Auch Calero kombiniert dabei verschiedene DQ-Metriken und dimensionale Modelle in einem formalen Rahmenwerk, das in Anlehnung an die Softwarequalität entwickelt wurde.

Werkzeuge zur Qualitätsverbesserung

Die vorgestellten Verfahren des Data-Cleaning wurden in verschiedenen Werkzeugen umgesetzt, um die Effizienz und Ergebnisqualität von Datenanalysen zu verbessern. Eine Übersicht über Datenqualitätswerkzeuge findet sich in [BG05a]. In [Hin01] wird ein technischer Ansatz für das Data-Cleaning vorgestellt, um kommerzielle Werkzeuge zur Datenmigration und -bereinigung mit domänenspezifischen Komponenten zu verbinden, um domänenspezifisches Wissen in das Data-Cleaning einfließen zu lassen.

Im Folgenden wird eine Auswahl kommerzieller Data-Cleaning-Werkzeuge zur Verbesserung der Datenqualität in Data-Warehouse-Systemen vorgestellt. Sie umfassen unterschiedliche Methoden, Techniken und Nutzerinteraktionen.

Trillium [Tri09] bietet Methoden des Data-Cleaning u.a. für die Verwaltung von Personendaten, zum Beispiel in der Kundenverwaltung. Es enthält Parser, Extraktoren, Transformatoren und vordefinierte Geschäftsregeln, um Namens- und Adressdaten zu bereinigen. Das „Trillium Software Systems" kann zur Standardisierung

von ETL-Prozessen, sowie zur Verknüpfung und Anreicherung von Eingangsdaten genutzt werden.

WizRule [Wiz09] dient zum Aufdecken von Abhängigkeiten zwischen Attributwerten und dem Erkennen von Ausnahmen. Formeln zur Berechnung von Attributwerten und Wenn-Dann-Regeln können definiert werden. Auch die Festlegung von korrekten Schreibweisen ist möglich. Damit kann WizRule genutzt werden, um Inkonsistenzen in Daten aufzudecken und zu korrigieren.

Potter's Wheel [RH01] ist ein interaktives Data-Cleaning-System, das Transformationen (Add, Drop, Copy, Fold, Merge, Split) eng mit der Konsistenzprüfung und Fehleranalyse verbindet. In einer Spreadsheet-ähnlichen Benutzerschnittstelle können Transformationsregeln auf nutzerfreundliche Weise mit Hilfe von Beispielen definiert werden. Formatverletzungen und Diskrepanzen werden sofort aufgedeckt und angezeigt.

AJAX [GFSS00b], [GFSS00a] bietet eine deklarative Sprache zur Definition von Datenflussgraphen mit Cleaning-Operatoren. Transformationen der logischen Ebene werden auf spezialisierte SQL-Operatoren abgebildet. Beim *Mapping* wird ein Datentupel in ein oder mehrere Tupel zerlegt. Das *Matching* führt einen Ähnlichkeitsverbund mit Hilfe von Clustering durch. Zuletzt verschmilzt das *Merging* verschiedene Tupel zur Duplikateliminierung.

Data Transformation Services [CRK00] ist ein ETL-Tool für SQL Server. Datentransformationen können zum Beispiel als Sequenz unterschiedlicher SQL-Skripte oder Data-Mining-Tasks mit bestimmten Datenquellen und -senken definiert werden. Er unterstützt die Erstellung und Ausführung von ETL-Prozessen, bietet allerdings keine Hilfe beim Auffinden von DQ-Fehlern und Bestimmen der optimalen Data-Cleaning-Schritte.

CLIQ (Data Cleansing with Intelligent Quality Management) [Hin09], [HA01] bezeichnet ein Vorgehensmodell für die Datenqualitätsverwaltung in Data-Warehouse-Systemen. Es bietet Methoden zur Datenqualitätsbewertung und -verbesserung. Die CLIQ-Workbench umfasst die Verwaltung von Fachdaten und Metadaten, Komponenten für einzelne Schritte der Datenintegration (Qualitätsbestimmung, Cleaning, etc.), Funktionen zur Ablaufplanung und -steuerung sowie zur Einbeziehung von Fremdsystemen (z.B. DTS, Microsoft Repository).

Bellmann System [Sri06], [DJMS02] dient zur Kontrolle und Extraktion von Datenbankstrukturen. Es bietet Techniken, um gleiche oder ähnliche Werte in Datenbankspalten zu schnell identifizieren. So werden Verbundpfade, -richtungen und -größen ermittelt. Diese Informationen können dem besseren Verständnis des Datenbankinhalts dienen, aber auch für Data-Mining- und Schema-Matching-Verfahren sowie zur Planung der Datenqualitätsverbesserung genutzt werden.

Business Objects [SAP09] bietet Methoden für das unternehmensweite Datenqualitätsmanagement in großen und mittelständischen Firmen. Eine hohe Nutzerfreund-

lichkeit wird durch vordefinierte Data Quality Blueprints^{TM} zur Datenqualitätsvisualisierung und durch Wizards zum Erstellen von Datenqualitätslösungen auf Basis zentralisierter Qualitätsrichtlinien gewährleistet.

Business Objects und der Datenqualitätsbrowser Bellmann integrieren Data-Cleaning-Methoden zur Datenbereinigung mit Nutzeroberflächen zur Visualisierung der erreichten Datenqualität. Weitere Werkzeuge, die sich auf die Qualitätsvisualisierung spezialisiert haben sind der Data Quality Visualiser „DaVis“ [SEG05] und der Visual Quality Evaluator „Viqtor“ [FN09], die die qualitativen Eigenschaften der Daten in der Tabellendarstellung durch farbliche Hinterlegung und/oder Schriftfarben hervorheben. [App04] und [GS05] visualisieren Unsicherheiten u.a. über die Nutzung von bis dahin ungenutzten grafischen Eigenschaften, wie Schärfe, Farbe oder Textur.

3.2.4. Online-Integration verteilter, heterogener Datenquellen

Während das Data-Cleaning vor der eigentlichen Anfrageverarbeitung ausgeführt werden kann, gibt es viele Anwendungen, in denen heterogene Datenquellen erst zur Laufzeit miteinander verknüpft werden. Somit kann die Datenqualitätsberechnung und -verbesserung erst während der Anfrageverarbeitung erfolgen. Im Folgenden wird eine Auswahl an Arbeiten vorgestellt, die sich mit der qualitätsgesteuerten Anfrageverarbeitung und -planung mittels Referenzmodellen, Websemantiken, Ontologien und anderen Technologien befassen, um möglichst hochqualitative Ergebnisse zu erzielen.

Allgemeine Ansätze

In Anlehnung an die kostengesteuerte Anfrageplanung bezieht Naumann in [NLF99] die Qualität als zusätzlichen Kostenparameter bei der Datenverarbeitung mit ein. Die Verarbeitungspläne werden über mehrere DQ-Dimensionen (u.a. Vollständigkeit, Aktualität, Genauigkeit) geordnet, wobei Pläne mit Ergebnissen geringer Qualität verworfen werden. Die Konzepte werden in einem Datenverarbeitungsprozessor umgesetzt, der multiple Datenbanken bei der Anfragebearbeitung integriert.

Dai et al. präsentieren die Spaltenheterogenität [DKO^{+}06] als neuartige Datenqualitätsdimension, um DQ-Probleme bei der Datenintegration heterogener Informationssysteme zu bemessen. Sie definieren Qualitätslevel, die die Spaltenheterogenität erfüllen muss und stellen die Cluster-Entropie als Qualitätsmaß vor. Je eindeutiger das Clustering ausfällt, umso höher wird die Datenqualität bewertet.

In [BJ07] wird das Datenqualitätsmanagement an Geschäftspraktiken ausgerichtet. Zuerst werden verlässliche Quellen und Integrationsbeziehungen identifiziert, um Verbundmodelle abzuleiten. Dann werden Regeln zur Vereinigung von überlappenden und/oder disjunkten Datenmengen gefunden. Weiterhin werden Datenänderungen unterschiedlicher Frequenz in die Berechnung der Qualität einbezogen.

Leser et al. schlagen einen Branch-and-Bound-Algorithmus vor, um die Anfrageplanung zu beschleunigen [LN00]. So wird verhindert, dass eine exponentielle Anzahl an Anfrageverarbeitungsplänen erarbeitet und bewertet wird. Mit Hilfe von strikten Qualitätsgrenzen, die die Pläne erfüllen müssen, können eine Vielzahl von Plänen mit schlechter Qualität in kurzer Zeit von der weiteren Verarbeitung ausgeschlossen werden.

Management-Informationssysteme (MIS) stellen eine Untergruppe der verteilten Informationssysteme dar. In [Mor82] werden Methoden zur Datenqualitätskontrolle in einem MIS vorgestellt. Sollen Datenänderungen im MIS vorgenommen werden, wird automatisch die Korrektheit der neuen Datensätze geprüft. Dabei werden einige als fehlerhaft bewertet und müssen manuell kontrolliert werden. Die von diesen Änderungen betroffenen Daten werden bis zur Überprüfung als fehlerhaft und nicht aktuell gekennzeichnet. Leider können Änderungen die automatische Kontrolle passieren, obwohl die neuen Datensätze Fehler beinhalten. Deshalb schlagen Morey et al. Kontrollmechanismen vor, um die Genauigkeit der MIS-Daten abzuschätzen.

Integration von Webdatenquellen

Das World-Wide-Web ist das wichtigste Medium zur Informationsverbreitung in nahezu allen Bereichen. Doch der Zugang wird durch die Suche nach relevanten Datenquellen zur Lösung eines bestimmten Problems erschwert. Nachdem eine Menge an relevanten Quellen gefunden wurde, müssen sie geordnet werden, um diejenigen zu identifizieren, die die spezifische Aufgabe am besten erfüllen. Dabei muss beachtet werden, dass Web-Datenquellen starke Qualitätsunterschiede (u.a. Vollständigkeit, Aktualität, Granularität) aufweisen.

In [CZW98] wird die qualitätsbasierte Auswahl von Quellen auf semistrukturierte Web-Informationen angewendet, um ein qualitätsbasiertes Ranking von Anfrageergebnissen zu erzielen und unvorhersehbare Antwortzeiten, irrelevante Ergebnisse und veraltete Daten zu vermeiden. Dazu müssen die Datenquellen ihre Qualitätsinformationen bereitstellen oder Methoden zur Qualitätsmessung gefunden werden. Zum Beispiel könnte die Qualität einer Webseite mit Hilfe der enthaltenen Rechtschreibfehler zu bewertet werden [AGB08]. Außerdem wird ein Modell zur Anfrageverarbeitung mit Qualitätskontrolle präsentiert, das ein DQ-Protokoll und Scheduling-Algorithmen beinhaltet.

Mihaila et al. präsentieren in [MRV00] ein Metadatenmodell ähnlich der DQ-Modellierung in Data-Warehouse-Systemen, um Informationen über den Inhalt und die Qualität von Web-Datenquellen anzulegen. Die Metadaten werden dabei so organisiert, dass eine effiziente Anfrageverarbeitung möglich wird. Des Weiteren stellen sie eine Anfragesprache zur Quellenselektion vor, die strikte und unscharfe Qualitätsanforderungen unterstützt.

Das HiQIQ-Projekt (High-Quality Information Querying) befasst sich mit der qualitätsgesteuerten Anfrage von dezentralen Informationssystemen, die über das Internet zugänglich sind [Nau09], [Nau02]. Kriterien der Informationsqualität, wie zum Beispiel

Vollständigkeit und Aktualisierungshäufigkeit, werden genutzt, um vor der Datenverarbeitung gute Datenquellen zu identifizieren, die Anfrageplanung zu verbessern und zu beschleunigen, den besten Plan zur Ausführung zu wählen und Anfrageergebnisse verschiedener Datenquellen zu einer hochqualitativen Antwort zu verknüpfen. Dabei wird ein Mediator-Ansatz verfolgt, der die Nutzeranfragen an die verteilten Datenquellen weiterleitet und Ergebnisse aggregiert.

Die vorgestellten Verfahren nutzen Datenqualitätsinformationen, um die Integration heterogener Webdatenquellen zu steuern und Anfrageergebnisse zu bewerten. Im Folgenden wird die Bewertung mit Hilfe anderer Kriterien skizziert. Der PageRank-Algorithmus [BP98] bewertet und gewichtet eine Menge verlinkter Dokumente, wie beispielsweise das World-Wide-Web, anhand ihrer Struktur und Linkpopularität. Das Gewicht einer Datenquelle (Webseite) ist umso höher, je mehr Links auf diese Seite verweisen und je höher das Gewicht der verweisenden Seiten ist. Entsprechend dem Gewicht werden die Datenquellen in der Ergebnisliste sortiert. Die semantische Suche [Man07] bewertet Webdatenquellen nicht auf Basis von Schlüsselbegriffen, sondern schätzt deren Bedeutung für die Suchanfrage mit Hilfe von zusätzlichen Informationen in Form von Annotationen oder Ontologien [GLC05]. Sie verknüpft unterschiedliche Informationsobjekte und unterstützt den Nutzer bei der Suche.

Kooperierende Informationssysteme

In kooperierenden Informationssystemen ist die Qualität der Daten, die von unterschiedlichen Quellen zur Verfügung gestellt und ausgetauscht wird, von hoher Bedeutung. Durch die lose Kopplung können sich schnell Fehler einschleichen und die Datenqualität in allen beteiligten Systemen mindern. Andererseits bietet der kooperierende Datenaustausch Möglichkeiten zur Korrektur und dient so der Verbesserung der Gesamtdatenqualität.

Laudon et al. untersuchen organisationsübergreifende, kooperierende Informationssysteme am Beispiel des Strafregistersystem der USA [Lau86]. Sie nutzen die Qualitätsdimensionen Genauigkeit, Vollständigkeit und Eindeutigkeit, um die Qualität der FBI-Strafregister der Bundesstaaten zu bewerten. Dabei stellen sie fest, dass nur etwa 25-50% der Daten vollständig und korrekt sind. Politische Bedenken verhindern die notwendige staatenübergreifende Qualitätskontrolle; und doch dienen die Strafregister weiterhin als Basis für viele Justizentscheidungen. Dieses Beispiel zeigt deutlich, wie wichtig die organisationsübergreifende Datenqualitätskontrolle in kooperierenden Informationssystemen ist.

In [SVM+04a] wird eine Architektur zur Datenqualitätsverwaltung in kooperierenden IS vorgestellt. Sie besteht aus zwei Modulen. Der Data-Quality-Broker ermöglicht die Anfrage und Verbesserung der Datenqualität. Der Quality-Notification-Service verbreitet geprüfte Datenänderungen im kooperierenden System über eine Publish/Subscribe-Kommunikation.

Peer-to-Peer-Systeme

Peer-to-Peer-Systeme bieten eine dezentrale, dynamische, datenzentrische Koordination von autonomen Organisationen. Ein großes Problem bei der Anfragebeantwortung in Peer-to-Peer-Systemen stellt die Konsistenz dar. Inkonsistente Daten in einer einzigen Quelle führen zu Antworten mit inakzeptabler Qualität.

Lenzerin et al. stellen in [Len04] qualitätsbasierte Semantiken zur Integration inkonsistenter Daten in Peer-to-Peer-Systemen vor. Informationen über Daten- und Peer-Qualität werden herangezogen, um die Datenquellen zu wichten und die inkonsistenten Anfrageergebnisse zu sortieren. Genügen die vorhandenen DQ-Informationen nicht für eine eindeutige Bestimmung der wahrscheinlich richtigen Antwort, wird jede Datenquelle einzeln betrachtet und eine Menge von möglichen Antworten geliefert.

Sensornetzwerke

In [RL07] wird ein qualitätsgarantierender, energieeffizienter Algorithmus (QGEE) für die Datenverarbeitung in drahtlosen Sensordatenbanksystemen vorgestellt. Der QGEE-Algorithmus verbindet die qualitätsgesteuerte Anfrageverarbeitung - bekannt aus der Integration heterogener Informationssysteme - mit der Verarbeitung von Sensordaten. Er arbeitet mit Methoden zur Datenverarbeitung in Sensornetzen [MFHH05], [RGT07], [DNGM07] und bestimmt die erreichte Genauigkeit mit Hilfe von Konfidenzintervallen. Damit wird der bezüglich Energieeffizienz und Ergebnisqualität optimale Anfrageplan berechnet. So können Ergebnisverfälschungen durch Basisdaten mit großen Fehlern vermieden werden.

Ren et al. bieten einen Auswahlmechanismus an, der die für die Anfragebearbeitung optimale Teilmenge der Sensoren in einem Sensornetzwerk identifiziert. Dieses Verfahren kann nur angewendet werden, wenn genug Sensoren zur Auswahl stehen, die überlappende Informationen zur Verfügung stellen. In sensorgesteuerten Anwendungen werden meist nur die minimal notwendigen Sensoren eingesetzt, da ihre Wartung sowie die Datenübertragung und -verarbeitung sehr teuer sind. In Anwendungen zur Sicherheitsüberwachung werden zwar redundate, jedoch gleichwertige Sensoren genutzt, um einen Messwert mehrfach zu kontrollieren, so dass auch hier keine optimale Teilmengenauswahl notwendig ist. Um die Datenqualität zu verbessern, müssen andere Möglichkeiten gefunden werden. Dazu werden in Kapitel 6 verschiedene Konfigurationsmöglichkeiten diskutiert.

3.3. Datenqualität in Datenströmen

In Abschnitt 2.3 wurden verschiedene Datenstrom-Management-Systeme vorgestellt. Um den Beschränkungen der Hardware-Ressourcen in Datenstromumgebungen zu entsprechen, werden Daten in Stromfenstern verarbeitet und ein Lastausgleich durch

Entfernen überzähliger Datentupel in Überlastsituationen ausgeführt. Um die Dienstqualität, wie Geschwindigkeit der Anfrageverarbeitung (Datenrate) oder Durchsatz, zu verbessern, werden Anfrageergebnisse näherungsweise berechnet. Dadurch sinkt Datenqualität der Verarbeitungsergebnisse. Im Abschnitt 3.3.1 werden Load-Shedding-Verfahren zum Lastausgleich vorgestellt und der Konflikt zwischen Dienst- und Datenqualität erörtert. In Abschnitt 3.3.2 wird auf die speziellen Eigenschaften und Methoden des DQ-Managements in Sensordatenströmen eingegangen. Es werden Forschungsarbeiten des Sensor-Monitoring und der drahtlosen Sensorkopplung untersucht.

3.3.1. Dienstqualität und Datenqualität

Load-Shedding zielt auf die Reduzierung des zu verarbeitenden Datenvolumens. In Überlastsituationen strömen mehr Datensätze in das Datenstromsystem als dieses verarbeiten kann. Die Verarbeitung aller eintreffender Datentupel würde zu signifikanten Verzögerungen und einem Speicherüberlauf führen. Die Dienstqualität (eng. quality of service, QoS) würde sich verschlechtern. Deshalb muss die Last verringert werden, indem bestimmte Anteile der Datentupel gelöscht werden bis ein verarbeitbares Datenvolumen oder eine akzeptable Verzögerung erreicht ist.

Es existiert eine große Auswahl an Load-Shedding-Verfahren, die alle die folgenden Fragen beantworten. *Wieviele* Datentupel müssen aus den Eingangsströmen entfernt werden? *Wo* muss der Lastausgleich im Verarbeitungspfad stattfinden? *Welche* Tupel müssen gelöscht werden?

Die ersten zwei Fragen werden detailliert in früheren Arbeiten hinsichtlich der Lastverteilung zwischen mehreren parallelen Anfragen und der optimalen Load-Shedding-Platzierung diskutiert. Die Load-Shedding-Rate, die den Anteil zu löschender Tupel bestimmt, wird auf Basis von Statistiken zu Datenstrom- und Operatoreigenschaften (Datenraten, Selektivitäten, etc.) und den verfügbaren Ressourcen berechnet. Zur Beantwortung der zweiten Frage wurden Verfahren zur Optimierung der Load-Shedding-Struktur vorgeschlagen, die alle auf derselben Grundlage arbeiten. Zur optimalen Verteilung der Load-Shedding-Operatoren müssen geteilte Segmente in Datenverarbeitungspfaden paralleler Anfragen gefunden werden [ACc+03b], [TZ06]. Der kontrollbasierte Ansatz [TLPY06] sieht weiterhin eine Feedback-Schleife zur dynamischen Adaption der Load-Shedding-Struktur bei geänderten Anfragen, Datenraten, etc. vor. In Abschnitt 6.1.2 wird der Ansatz von Babcock et al. [BDM04] als Basis des datenqualitätsgesteuerten Lastausgleichs genauer beschrieben, der existierende Arbeiten durch die Integration von Qualitätsinformationen signifikant erweitert.

Im Folgenden werden verschiedene Strategien zur Beantwortung der letzten Frage verglichen, um den Konflikt zwischen Dienst- und Datenqualität herauszustellen.

Der einfachste Ansatz des Lastausgleichs [TcZ+03] erzeugt eine Zufallsstichprobe der eintreffenden Datenstromtupel. Dadurch wird die Dienstqualität verbessert. Die Antwortzeit wird verkürzt und der Tupeldurchsatz erhöht. Das Löschen einiger Datentupel

führt jedoch zu einem Datenverlust, so dass nachfolgend Verarbeitungsergebnisse nur approximiert werden können. Aggregationsergebnisse werden ungenau und wichtige Verbundpartner können entfernt werden, bevor sie zur Ergebnismenge beitragen können, so dass diese unvollständig ist. Je mehr Tupel aus dem Strom entfernt werden, umso besser ist die Dienstqualität, doch umso schlechter fällt die Ergebnisdatenqualität aus.

Komplexe Strategien des semantischen Load-Sheddings wurden entwickelt, um die Genauigkeit und/oder Vollständigkeit der Anfrageergebnisse mit Hilfe gewichteter Stichproben zu verbessern. Kang et al. stellen Strategien zur Speicherplatzverwaltung vor, um die Ausgabemenge von Verbundoperatoren auf Basis unbegrenzter Fenster und Häufigkeitsverteilungen der Eingangsdatenströme zu maximieren [KNV02]. Ein entgegengesetzter Ansatz wird in [SW04] verfolgt, der Datenstromtupel mit Hilfe einer altersbasierten Statistik verwaltet. Longbo et al. folgen einer korrelierten Load-Shedding-Strategie für mehrere Eingangsdatenströme [LZZM07]. Die Ergebnismenge wird maximiert, indem die Domäne der Verbundattribute partitioniert und Tupel entsprechend ihrer Domänenzugehörigkeit gefiltert werden. Obwohl diese Algorithmen die Ergebnismenge von Verbundoperatoren maximieren, gehen einige Verbundpartner unwiederbringlich verloren. Außerdem werden Unvollständigkeiten durch a priori fehlende Werte nicht beachtet, da Informationen über die Datenqualität nicht berücksichtigt werden.

[TZ06] richtet sich auf Aggregationsanfragen in gleitenden Datenstromfenstern. Um Überlast zu reduzieren, werden vollständige Fenster entfernt, so dass korrekte, aber unvollständige Aggregationsergebnisse ausgegeben werden. Im Gegensatz dazu präsentieren Babcock et al. einen Load-Shedding-Algorithmus, um Aggregationen zu approximieren und ungenaue, aber vollständige Ergebnisse zu berechnen [BDM04]. Der durch den Datenverlust entstandene Fehler wird dabei mit Hilfe von Datenstromstatistiken minimiert.

Während die bisher geschilderten Algorithmen Datenstromstatistiken nutzen, führt [OZH06] die „Wichtigkeit“ (engl. importance) von Datentupel ein, um die Load-Shedding-Wahrscheinlichkeit zu bestimmen. Zur Verbesserung nachfolgender Aggregationsergebnisse verbleiben wichtige Tupel im Strom. Des Weiteren kann die „Nützlichkeit“ (engl. utility) mit Statistikinformationen verknüpft werden [ACc+03b], um das Load-Shedding zu wichten. Neben tupelbasiertem Load-Shedding können einzelne Attribute des Datenstromtupels [AN07] entfernt werden, so dass nur informative Attribute im Datenstrom erhalten bleiben. Der Triage-Ansatz [RH05] begegnet Überlastsituationen, indem Tupel nicht gelöscht, sondern zu Synopsen zusammengefasst werden. Allerdings führt auch hier die Vorverarbeitung der Daten einen Genauigkeits- und Vollständigkeitsfehler in nachfolgende Verarbeitungsschritte ein.

Zusammenfassend lässt sich festhalten, dass verschiedene Load-Shedding-Algorithmen entwickelt wurden, die für die Ausführung von Verbund und/oder Aggregationen optimiert sind. Alle existierenden Techniken fügen durch die Reduzierung des zu verarbeitenden Datenvolumens Fehler in die Verarbeitungsergebnisse ein. Entweder fehlen

Tupel der Ergebnismenge (reduzierte Vollständigkeit) oder Aggregationsergebnisse werden approximiert (reduzierte Korrektheit). Je stärker die Überlast, umso mehr Tupel müssen entfernt werden, um die Dienstqualität zu gewährleisten und umso größer ist der eingeführte Fehler in der Datenqualität.

Der in Abschnitt 6.1 vorgestellte qualitätsgesteuerte Lastausgleich löst dieses Problem. Werden vor allem Tupel von geringer Datenqualität aus dem Strom gelöscht, kann die durchschnittliche Qualität des Datenstroms und damit der Anfrageergebnisse verbessert werden. Der eingefügte Fehler wird minimiert oder sogar kompensiert. Der Konflikt zwischen Dienst- und Datenqualität kann so zum Teil aufgehoben werden. Die Dienstqualität wird durch Reduzierung der Überlast gewährleistet, zeitgleich verbessert das Entfernen der Tupel die Datenqualität.

3.3.2. Sensordatenqualität

In diesem Abschnitt werden Forschungsarbeiten diskutiert, die die Datenqualität von Sensordaten beschreiben. Zuerst wird die Qualität von RFID[2]-Daten untersucht. Danach wird eine Arbeit von Biswas et al. vorgestellt, die sich mit der Vollständigkeit von Sensordatenströmen befasst. Obwohl hier nur die Vollständigkeit als Ausschnitt der Datenqualität betrachtet wird, liegt eine sehr große Themennähe zur vorliegenden Dissertation vor. Schließlich wird eine Arbeit von Tatbul vorgestellt, in der Sensorsysteme auf Basis von Datenqualitätsinformationen konfiguriert werden.

Qualität von RFID-Sensordaten

RFID [Fin06], [CKR07] wird u.a. zur automatisierten, berührungslosen Identifizierung und Lokalisierung von Gegenständen und für die automatische Erfassung und Speicherung von Daten genutzt. RFID-Lesegeräte sind spezielle Sensoren zum Auslesen der RFID-Kennung.

Die zur Objekterkennung aufgenommenen Rohdatenströme können verschiedene Fehler aufweisen. Sie können fehlerhafte Daten (engl. false positives) beinhalten oder Objekte übersehen (engl. false negatives). Die Fehlerkorrektur basiert auf zeitlichen und örtlichen Relationen und Eigenschaften der Realwelt. Doch nicht alle Fehler (z.B. false negatives) können auf diese Weise behoben werden. In [SJFW06] wird die Fehlerabschätzung bei der Objekterkennung adressiert. Die Konfidenz dient zur Bewertung der fehlerhaft gelesenen Daten. Die Abdeckung bestimmt den Anteil verpasster Objekte.

In der vorliegenden Arbeit ist der Fokus auf physikalische Sensoren und quasikontinuierliche Signale gerichtet. Die Objekterkennung mit Hilfe von RFID-Sensoren wird nicht untersucht. RFID-Systeme bieten jedoch eine Vielzahl von Einsatzmöglichkei-

[2] RFID (Radio Frequency Identification) beschreibt die automatische Objekterkennung mit Hilfe von elektromagnetischen Wellen.

ten, so dass ein umfassendes Datenqualitätsmanagement bei der Objekterkennung die Informationsqualität in vielen Anwendungen verbessern kann.

Vollständigkeit von Sensordaten

Biswas, Naumann und Qiu stellen in [BNQ06] ein System zur Berechnung der Vollständigkeit von Sensordaten vor. Sie analysieren vier verschiedene Operatorenklassen, um den Informationsverlust bei der Verarbeitung und Propagierung im Sensornetz zu bestimmen. Sie evaluieren ihre Konzepte im Kontext der Überwachung von Patienten oder hilfebedürftigen Menschen in sogenannten „Smart Spaces“. Mit Hilfe von Bewegungssensoren, Lichtschranken und Pulsmessgeräten können zum Beispiel folgende Fragen beantwortet werden. In welchem Raum befindet sich die Person? Läuft sie mit ruhigen Schritten? Ist die Person aufgeregt?

Biswas et al. definieren vier Datenraten zur Bestimmung der Vollständigkeit. Die Systemrate ist die maximale Datenrate aller Sensoren im überwachten System. Die Samplingrate eines Sensors ist die Abtastrate, mit der der Sensor Umweltdaten aufnimmt. Die Sensordatenrate bestimmt die Rate, mit der ein Sensor Informationen an das System zur Datenverarbeitung übermittelt. Die Anfragedatenrate ist die Rate, mit der Informationen aus dem System abgefragt werden.

Die Sensordaten füllen eine virtuelle Relation, deren Tupelzahl durch die Systemrate angegeben wird. Ist eine Sensordatenrate kleiner als die Systemdatenrate, entstehen Null-Werte in der virtuellen Universalrelation. Die Systemvollständigkeit wird als Durchschnitt der Sensordatenraten in Relation zur Systemrate definiert. Sie ist ein Maß für die Anzahl der Null-Werte der Universalrelation.

Disjunktive Operatoren geben einen Datenwert zurück, wenn mindestens ein eingehender Sensorwert einen Datenwert (Nicht-Null-Wert) aufweist. Die Vollständigkeit wird als Maximum der eingehenden Vollständigkeiten berechnet. Bei konjunktiven Operatoren, zum Beispiel binäre mathematische Operatoren wie Addition oder Subtraktion, kann nur ein Ergebniswert berechnet werden, wenn kein einziger Null-Wert vorhanden ist. Hier wird die Vollständigkeit als das Minimum der eingehenden Sensorvollständigkeiten berechnet. Bei Kompressionsoperatoren muss die Vollständigkeit durch die Kompressionsrate geteilt werden. Bei Aggregationsoperatoren wird der Durchschnitt der eingehenden Vollständigkeiten gebildet.

Um alle Anfragen vollständig beantworten zu können, muss die Anfragevollständigkeit größer oder gleich eins sein. Sie besteht aus der Systemvollständigkeit des Anfrageergebnisses geteilt durch die Anfragedatenrate. Somit wird die notwendige Systemvollständigkeit über die Anfragedatenrate berechnet. Biswas et al. gehen jedoch nicht darauf ein, wie diese Systemvollständigkeit auf die notwendigen Sensordatenraten aufgeteilt wird, wenn mehr als zwei Sensordatenströme miteinander verrechnet werden.

Die Systemvollständigkeit unterscheidet sich durch die Herkunft der Null-Werte von der Definition der Vollständigkeit, die in der vorliegenden Arbeit verwendet wird.

Biswas et al. nutzen die maximale reguläre Sensordatenrate, um ein Maß für die Vollständigkeit abzuleiten. Sensorausfälle, die die Datenrate temporär herabsetzen und weitere Null-Werte einfügen, werden nicht betrachtet. Um diese Unvollständigkeiten messen zu können, wird in der vorliegenden Arbeit jeder Sensor einzeln überwacht. Erst im Verlauf der Datenverarbeitung werden die Datenraten der Sensorströme für den Verbund angepasst, so dass die Vollständigkeit verschiedener Datenströme verknüpft werden muss.

Außerdem werden fehlende Werte in der vorliegenden Arbeit nicht direkt in die Datenverarbeitungskette propagiert, sondern sofort interpoliert, um kaskadierende Null-Werte zu vermeiden. Die Vollständigkeit gibt damit nicht über den Anteil der Null-Werte, sondern über den Anteil interpolierter Datensätze Auskunft.

Trotz dieser Unterschiede können Überlegungen zur Propagierung der Systemvollständigkeit genutzt werden, um Anforderung 4 zu erfüllen. Es muss geprüft werden, ob die in Abschnitt 2.3 aufgelisteten Operatoren in die Operatorenklassen eingeordnet werden können und wie die Funktionen der Operatorenklassen an die hier vorliegende Definition der Vollständigkeit angepasst werden können.

Qualitätsgesteuerte Konfiguration der Sensoren

Tatbul et al. stellen in [TBH+04] ein Monitoring-System zur Überwachung der Lebenszeichen von Soldaten vor. Dafür werden Sensoren in eine „Smart Uniform“ integriert, die u.a. den Pulsschlag, die Temperatur, und die Unversehrtheit der Uniform überwachen. Die physikalische Umwelt (Bewegung, mögliche Schäden oder Verletzungen im Kampf, etc.) führt zu Messfehlern oder -ausfällen. Außerdem muss die niedrige Bandbreite auf die verschiedenen Sensordatenströme aufgeteilt werden. Die entstandenen Ungenauigkeiten reduzieren die Zuverlässigkeit der Daten, die einen Grenzwert nicht unterschreiten darf. Tatbul et al. schlagen deshalb die qualitäts- bzw. konfidenzgesteuerte Auswahl der aktiven Sensoren vor.

Zuerst muss die Konfidenz aus drei Eigenschaften des Sensorsystems bestimmt werden. Das „Modell“ beschreibt die Menge der Sensoren. Je mehr Sensoren genutzt werden, umso höher ist die Zuverlässigkeit des Ergebnisses. Die „Latenzzeit“ der Sensormessungen ist ein weiterer Faktor. Mit dem Alter der Messwerte sinkt deren Relevanz und Beitrag zur Konfidenz. Der dritte Faktor ist der berechnete „Zustand“ des Soldaten. Die Antwort „Alles ist in Ordnung.“ kann mit höherer Konfidenz bestimmt werden als eine mögliche Verletzung.

Am Beispiel der Körpertemperaturüberwachung werden sechs Modelle entwickelt. Die Temperaturmessung ist mit sechs Sensorgruppen möglich, von einfacher Hauttemperaturmessung an mehreren Körperstellen bis zur Temperaturmessung mit Hilfe einer verschluckbaren Pille. Tatbul et al. ermitteln dann experimentell, welches Modell bei welchem Körperzustand die höchste Konfidenz erzielt und die geringste Energie verbraucht.

Das vorgestellte Verfahren ist ein erster Ansatz, um ein Sensorsystem mit Hilfe von Qualitätsinformationen zu konfigurieren. Um die Qualitätsanforderungen einer hohen Konfidenz zu erfüllen, werden je nach Zustand andere Sensoren aktiviert. Die Optimierung beruht auf einer experimentellen Ermittlung der erreichbaren Konfidenz, die wiederum nur mit Hilfe der Modellparameter abgeschätzt werden kann. Das Verfahren ist nur für die Körpertemperaturmessung mit den vorgegebenen Sensoren anwendbar. Es kann nicht auf andere Anwendungsgebiete übertragen werden.

In der vorliegenden Dissertation soll dagegen eine Methode zur qualitätsgesteuerten Anpassung der Sensordatenverarbeitung entwickelt werden, die in allen Sensoranwendungen genutzt werden kann (siehe Anforderung 8). Kapitel 6 stellt diese Methode der Qualitätsoptimierung vor.

3.4. Zusammenfassung

In diesem Kapitel wurden existierende Konzepte und Methoden zur Datenqualitätsmodellierung, -verarbeitung und -verbesserung vorgestellt und hinsichtlich ihrer Anwendbarkeit auf Sensordatenströme untersucht. Abschnitt 3.1 beschreibt verschiedene Metamodelle zur Datenqualitätsspeicherung. Sie eignen sich gut für die Qualitätsverwaltung in gängigen Datenbankanwendungen. Allerdings wird ein zu großes Datenvolumen benötigt, um den Ressourcenbeschränkungen in Datenstromsystemen zu genügen.

Das Datenqualitätsmanagement in Datenströmen muss entsprechend Anforderung 2 Speicherplatz sparender gestaltet werden. Anforderung 3 verlangt außerdem einen effizienten Import von Rohdaten oder Verarbeitungsergebnissen in eine Zieldatenbank. Um dies zu ermöglichen, muss das Speicherkonzept in relationalen Datenbanken angepasst werden.

Gegenwärtige Arbeiten analysieren die Datenqualitätsverarbeitung vor allem aus Sicht der relationalen Algebra. Die vorgestellten Ansätze sind auf relationale Mengenoperatoren wie Selektion, Union oder Verbund beschränkt. Einzig Naumann et al. untersuchen die Fortpflanzung von Vollständigkeitsinformationen für eine größere Zahl an Operatoren. Ziel der vorliegenden Arbeit ist es, diese Ansätze signifikant zu erweitern (siehe Anforderung 4 und 5). Es wird eine umfassende Datenqualitätsalgebra entwickelt, die sowohl Operatoren der relationalen Algebra, der Signalanalyse und der Mathematik unterstützt.

Anforderung 6 bezieht sich auf die Visualisierung der Sensordatenqualität. Bisherige Werkzeuge stellen Datenqualität von String-Daten in Tabellenform dar bzw. beschreiben geografische Daten mit grafischen Texturen z.B. auf Karten. Meist werden nur wenige Dimensionen wie Konsistenz oder Genauigkeit unterstützt. Zur Visualisierung der Qualität strömender numerischer Sensormessdaten ist eine dynamischere Darstellung notwendig, die alle Dimensionen aus Anforderung 1 umfasst.

Um Überlastsituationen in Datenstromsystemen zu begegnen, wurden verschiedene Load-Shedding-Verfahren in Abschnitt 3.3 erläutert. Das Entfernen überzähliger Datentupel führt einen Fehler ein, der sich in der Korrektheit von Aggregationsergebnissen oder der Vollständigkeit von Verbundmengen bemerkbar macht. Entsprechend Anforderung 7 soll eine neue Load-Shedding-Strategie unter Berücksichtigung der verfügbaren Datenqualitätsinformationen entwickelt werden, die die Ergebnisqualität im Vergleich mit existierenden Verfahren verbessert.

Ein weiteres Ziel ist die Integration von Nutzeranforderungen zur Datenqualitätsverbesserung (siehe Anforderung 8). Zur Qualitätsverbesserung in Data-Warehouse-Systemen werden in Abschnitt 3.2 verschiedene Data-Cleaning-Methoden und -Werkzeuge diskutiert. Sie sind jedoch auf Qualitätsprobleme wie Inkonsistenzen oder Datenduplikate gerichtet, die in Sensordatenströmen nicht vorkommen. Außerdem erfolgt die Datenbereinigung vor der Anfrageverarbeitung und benötigt zum Teil den vollständigen Datensatz. Dies widerspricht der kontinuierlichen Datenverarbeitung in Datenströmen. Während der Online-Integration heterogener Datenquellen richtet sich die Datenverarbeitung nach der jeweiligen Datenqualität der Einzelquellen. Auch bei der qualitätsgesteuerten Sensorkonfiguration (Abschnitt 3.3.2) wird die Anfrageverarbeitung an Qualitätsgegebenheiten und -forderungen angepasst. Diese Idee wird aufgenommen, um Nutzeranforderungen bezüglich der Sensordatenqualität zu integrieren. Die Datenverarbeitung wird mit Hilfe von Optimierungsalgorithmen angepasst, um die gewünschte Datenqualität in allen Dimensionen zu gewährleisten.

4

Modellierung der Sensordatenqualität

Die Analyse verwandter Arbeiten zeigt, dass die Verwaltung von Datenqualitätsinformationen in Datenströmen weder auf Modellebene von existierenden Metadatenstrukturen noch auf konzeptioneller Ebene von bestehenden Datenstrom-Management-Systemen unterstützt wird. Deshalb wurde das traditionelle Datenstrom- und Datenbankmetamodell um Komponenten zur Modellierung von Datenqualitätsinformationen erweitert. Das entwickelte Datenqualitätsmodell wird DQMx (Data Quality Model extension) genannt.

In diesem Kapitel wird zuerst die ressourcensparende Modellierung und Propagierung von Qualitätsinformationen in Datenströmen untersucht. Danach werden die gewonnenen Kenntnisse auf die effiziente Speicherung von Datenqualität in relationalen Datenbanksystemen übertragen. Abschließend werden Mechanismen zur Adaption und Optimierung der Modellparameter vorgestellt.

4.1. Datenqualitätspropagierung im Datenstromsystem

Im Folgenden werden zwei Ansätze zur flexiblen Datenqualitätspropagierung vorgestellt und bezüglich der in Kapitel 2.5 aufgestellten Anforderungen in Datenstromsystemen verglichen. Anschließend werden die mathematischen Grundlagen des Datenqualitätsmodells gelegt.

4.1.1. Naive Datenqualitätsannotationen

Ein einfacher Ansatz zur Propagierung von Datenqualitätsannotationen besteht darin, jeden Datenstromwert mit Datenqualitätsinformationen für alle relevanten Dimensionen anzureichern, wie es in [SW98] und [WSF95] vorgeschlagen wird. Qualitätsinformationen werden dabei mit der Datenrate des zu beschreibenden Datenstroms gesendet und verarbeitet. Abbildung 4.1 zeigt einen Ausschnitt aus der Druckverlustmessung aus Anwendungsszenario 1. Die Datenqualitätsdimensionen Genauigkeit, Konfidenz, Vollständigkeit und Datenmenge werden für jeden Datenwert im Datenstrom mitgeliefert.

Zeitstempel	…	210	211	212	213	214	215	216	217	218	219	220	221	222	223	224	225	226	227	228	229	…
Druckverlust	…	180	178	177	175	176	181	189	201	204	190	194	192	189	183	215	210	211	199	187	184	…
Genauigkeit	…	3,5	3,5	2,9	2,1	3,0	2,7	4,2	5,1	3,8	3,7	2,3	2,6	1,9	3,7	3,4	2,9	2,7	3,6	3,2	1,9	…
Konfidenz	…	12,3	12,5	13,0	12,8	14,0	13,6	13,6	13,2	13,4	12,9	12,4	12,7	13,1	10,9	11,4	12,4	12,3	12,4	12,0	11,5	…
Vollständigkeit	…	0.9	0.9	0.8	1	0.9	0.85	0.8	0.75	0.8	0.8	0.9	0.9	0.9	0.8	1	1	1	1	0,95	1	…
Datenmenge		2	2	2	2	2	2	2	2	2	2	2	2	2	2	2	2	2	2	2	2	

Abbildung 4.1.: Naive Datenqualitätsannotationen

Ein Datenstrom D umfasst m Tupel $\tau(j)$ mit $1 \leq j \leq m$ bestehend aus je n Attributwerten A_i mit $1 \leq i \leq n$ und einem Zeitstempel $t(j)$. Für jedes Attribut können die verwendeten DQ-Dimensionen unabhängig ausgewählt werden, so dass jeder Attributwert A_i nicht mehr nur durch den Datenwert $x_i(j)$, sondern zusätzlich durch ϑ_i Datenqualitätswerte $q_i(j)$ beschrieben wird.

Der naive Ansatz der Datenqualitätsverwaltung widerspricht Anforderung 2 nach geringem Datenstromvolumen, um den Ressourcenbeschränkungen in realen sensorgestützten Anwendungen zu begegnen. Die naive Datenqualitätsannotation sollte daher nur Verwendung finden, wenn die Kosten zur Datenübertragung und -verarbeitung sehr gering sind.

4.1.2. Datenqualität in Datenstromfenstern

Um Ressourcenkonflikte in Datenstromsystemen zu lösen, wird die fensterweise Datenverarbeitung verwendet. Verbundoperatoren oder Aggregationen werden nicht über den gesamten Datenstrom, sondern in gleitenden Fenstern ausgeführt.

In der Datenqualitätsmodellierung werden Datenstromfenster genutzt, um einen Kompromiss zwischen Ressourcenbelastung und Granularität der DQ-Informationen zu finden. Im Gegensatz zur Datenstromverarbeitung, die zumeist auf gleitenden Fenstern basiert, wird der Datenstrom attributweise in aufeinander folgende, nicht überlappende Fenster einer gegebenen Fenstergröße geteilt.

Wie in Abbildung 4.2 dargestellt, wird die Datenqualität nicht für jedes einzelne Datenelement, sondern aggregiert am Ende eines jeweiligen Fensters übertragen. Die

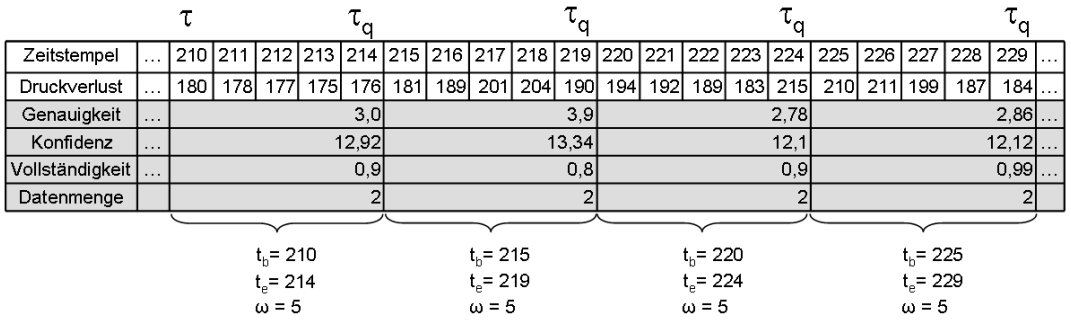


Abbildung 4.2.: Beispiel der fensterbasierten Datenqualitätsmodellierung

Fenstergröße bestimmt die Granularität der Datenqualität und die Verzögerung ihrer Auslieferung. Durch das verringerte Datenvolumen gegenüber dem naiven Ansatz können wichtige Ressourcen in der Datenübertragung gespart werden. Das Datenqualitätsfenster (engl. data quality window) bildet die Basis des Datenqualitätsmodells DQMx.

Jeder Attributstrom A_i des Datenstroms D wird in κ_i Fenster zerlegt. Jedes Datenqualitätsfenster $w_i(k)$ für $1 \leq k \leq \kappa_i$ wird mit Hilfe des Startpunkts t_b, des Endpunktes t_e und der Fenstergröße ω identifiziert. Um die fensterbasierten Datenqualitätsinformationen im Datenstrom zu übertragen, wird ein neuer Tupeltyp eingeführt. Das Datenqualitätstupel τ_q umfasst neben den traditionellen Tupelelementen (Zeitstempel, Attribut) einen Datenqualitätswert q_w für jede Dimension $q \in Q_i$. Somit enthält ein Datenqualitätsfenster ω gemessene (Sensor-)Datenwerte für jedes Attribut A_i sowie ϑ_i Datenqualitätsinformationen. Es besteht aus $\omega - 1$ traditionellen Datenstromtupeln τ und einem Datenqualitätstupel τ_q am Ende des Fensters.

Abbildung 4.2 zeigt den Datenstromausschnitt der Druckverlustmessung mit Datenqualitätsfenstern der Größe $\omega = 5$ und $\vartheta = 4$ Datenqualitätsdimensionen: Genauigkeit, Konfidenz, Vollständigkeit, Datenmenge. Das generische Datenqualitätsmodell ist jedoch nicht auf diese Datenqualitätsdimensionen beschränkt. Um das Modell flexibel für weitere Anwendungsgebiete zu gestalten, wurde beim Entwurf auf die einfache Erweiterbarkeit geachtet.

Des Weiteren ist die Fenstergröße nicht festgeschrieben, sondern kann für jedes Attribut gesondert definiert werden und ist auch während der Laufzeit der Datenstromübertragung adaptierbar. Kleine Datenqualitätsfenster liefern feingranulare DQ-Informationen, resultieren aber in größerem Datenzuwachs. Größere Fenster gewährleisten die benötigten Ressourceneinsparungen, jedoch auf Kosten der DQ-Informationsgüte, die nun über viele Datenstromelemente aggregiert wird. Um dem Nutzer die schwierige Aufgabe der Fenstergrößendefinition abzunehmen, werden in Abschnitt 4.3 zwei Ansätze zur automatischen Adaption vorgestellt.

Das um Datenqualitätsfenster erweiterte Datenstrommodell ist in Abbildung 4.3 in der Notation des OMG Standards des Common-Warehouse-Metamodells (CWM) [PM01] dargestellt. Der CWM-Erweiterungsmechanismus der Vererbung wird genutzt, um die Klassen DataStream und DataStreamAttribute zur Beschreibung des Datenstromschemas von den bestehenden Klassen Table und Column des relationalen Schemas abzuleiten. Das Paket RowSet referenziert Instanzdaten, die in einer Datenbanktabelle gespeichert sind. Entsprechend wurde die Klasse SubStream abgeleitet, um einen Teildatenstrom bestehend aus ω Datentupeln, aufgenommen über eine gegebene Zeitspanne zu definieren.

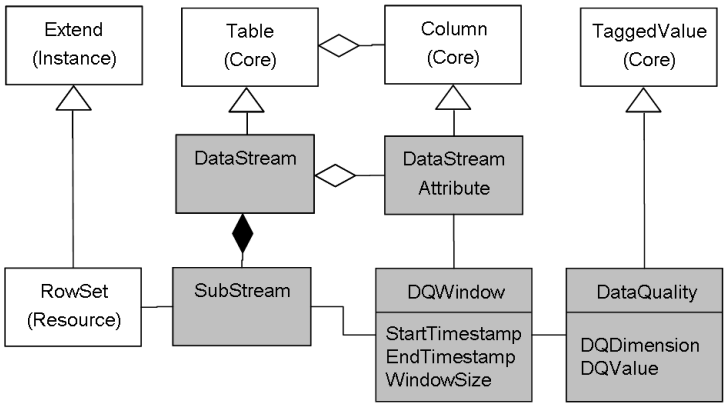


Abbildung 4.3.: Metamodell der fensterbasierten Datenqualitätsverwaltung

Die Klasse DQWindow wurde als Verbindungspunkt zwischen der schematischen Sicht und der Instanzdatensicht eingeführt. Die Datenqualitätsfenster im Datenstrom werden als Instanzen dieser Klasse mit den Attributen Anfangszeitpunkt (StartTimestamp), Endzeitpunkt (EndTimestamp), der Fenstergröße (WindowSize) sowie den benötigten Referenzen zu Schema- und Instanzsicht definiert. Für jedes Datenqualitätsfenster wird die Datenqualität (DataQuality) als TaggedValue der CWM-Erweiterung modelliert. Dabei entspricht jede Datenqualitätsinformation einer Instanziierung der Klasse (DataQuality) mit dem Attribut-Werte-Paar <DQDimension, DQValue>.

Mit Hilfe der fensterweisen Datenqualitätsübertragung und -verarbeitung kann der zusätzliche Kommunikations- und Speicheraufwand im Vergleich zur naiven DQ-Annotation um den Faktor der durchschnittlichen Fenstergröße $\bar{\omega} = m/\bar{\kappa}$ verringert werden, wobei κ die über die Attribute gemittelte Fensteranzahl bezeichnet. Die fensterbasierte Datenqualitätsübertragung erfüllt damit Anforderung 2.

4.1.3. Berechnung der Fensterdatenqualität

In diesem Abschnitt werden die mathematischen Grundlagen zur initialen Berechnung der Datenqualität eines Datenstromfensters erläutert. Ausgehend von der Datenqualitäts-

definition in Abschnitt 2.4.2 werden Methoden zur fensterbasierten Aggregation jeder Datenqualitätsdimension vorgeschlagen. Das Attribut A_0 repräsentiert den Zeitstempel des Datenstroms. Dieser wird nicht mit Datenqualitätsinformationen ausgestattet. Die folgenden Definitionen gelten für jedes weitere Attribut $A_i (1 \leq i \leq n)$ des Datenstroms, so dass für eine einfachere Schreibweise der Index i weggelassen wird.

Genauigkeit Die Genauigkeit beschreibt den absoluten, systematischen Fehler einer Messreihe, d.h. eines Datenstroms. Die Genauigkeit eines Datenstromfensters a_w wird durch die Zusammenfassung der Genauigkeiten, d.h. der absoluten, systematischen Fehler der enthaltenen Datenelemente $a(j)$ berechnet. Um die Gesetze der Gauß'schen Fehlerfortpflanzung in der Datenqualitätsverarbeitung anwenden zu können, muss die lineare Durchschnittsberechnung als Aggregationsfunktion gewählt werden. Außerdem erlaubt diese Aggregation die Abschätzung der Güte der Genauigkeitsinformationen, die die Grundlage für die automatische Adaption des Modellparameters der Fenstergröße in Abschnitt 4.3 bildet.

$$a_w(k) = \frac{1}{\omega_k} \sum_{j=t_b}^{t_e} a(j) \tag{4.1}$$

Konfidenz Die Konfidenz bemisst den absoluten, statistischen Messfehler mit Hilfe der Standardabweichung der Sensormesswerte. Um die initiale Konfidenz eines Datenqualitätsfensters zu bestimmen, muss die Standardabweichung der enthaltenen Sensordaten berechnet werden. In der vorliegenden Arbeit wurde die Konfidenzwahrscheinlichkeit von 99% gewählt, so dass die Standardabweichung mit 2,58 gewichtet werden muss (siehe Tabelle 2.1). Dieser Wert kann jedoch im generischen Qualitätsmodell entsprechend der Anwendung frei gewählt werden.

$$\epsilon_w(k) = 2,58 \cdot \sigma(w_k) \tag{4.2}$$

$$\sigma(w_k)^2 = \frac{1}{w} \sum_{j=t_b}^{t_e} x(j)^2 - \bar{x}^2 \tag{4.3}$$

Vollständigkeit Die Vollständigkeit beschreibt den Anteil gemessener Datenwerte $x \in X$ im Sensordatenstrom, um interpolierte Daten $\tilde{x} \in \tilde{X}$ zu kennzeichnen. Die Vollständigkeit eines Datenqualitätsfensters bezeichnet den Anteil der gemessenen Daten im Datenfenster. Sie gibt für jedes Datenelement des Fensters die Wahrscheinlichkeit an, dass dieses einem gemessenen Sensorwert entspricht. Müssen Konfidenz oder Vollständigkeit im Verlauf der Datenqualitätsverarbeitung aus den Eingangsqualitäten berechnet werden, so wird entsprechend der Berechnung der Fenstergenauigkeit der lineare Durchschnitt angewendet.

$$c_w(k) = \frac{|X(w_k)|}{\omega_k} \tag{4.4}$$

Datenmenge Die Datenmenge gibt an, wieviele gemessene Datenelemente x im Zuge der Datenstromverarbeitung zur Berechnung eines Datenelementes $x' = f(x)$ beigetragen haben. Während der Initialisierung des DQ-Fensters wurden noch keine Verarbeitungsschritte durchgeführt, so dass gilt $d_w(k) = 1$.

Aktualität Wie bereits ausgeführt, nimmt die Aktualität in der Datenqualitätsverwaltung eine Sonderrolle ein. Sie kann direkt aus den Zeitstempeln des Datenstroms bestimmt werden und muss somit nicht im Datenstrom mitgeführt werden. Die Aktualität wird nicht fensterbasiert berechnet, sondern kann ohne Mehraufwand für jedes einzelne Datenstromelement angegeben werden.

Die Aufteilung und Berechnung der Datenqualitätsfenster, kann direkt an der Datenquelle, dem Sensor, aber auch jedem anderen Punkt in der Datenstromverarbeitung stattfinden. Für eine effiziente Verarbeitung und kostengünstige Kommunikation sollte die Aggregation in Datenqualitätswerte jedoch so nah am Sensorknoten wie möglich erfolgen.

4.2. Persistente Qualitätsspeicherung

In vielen Anwendungen ist eine Untersuchung der Sensordatenhistorie notwendig, um zum Beispiel Entwicklungen des Datenverlaufs zu analysieren und Trends abzuleiten. Dazu müssen die Ergebnisse der Datenstromverarbeitung sowie deren Datenqualitätsinformationen in eine Datenbank eingepflegt werden. Dabei soll die Speicherplatz sparende Verwaltung in Datenqualitätsfenstern beibehalten werden, um einen effizienten Datenqualitätsimport vom Datenstrom in die Datenbank zu gewährleisten (siehe Anforderung 3).

Außerdem müssen Methoden zum Abfragen der Qualitätsinformationen zur Verfügung gestellt werden (Anforderung 5), um eine Schnittstelle zwischen Datenbank bzw. Datenstromsystem und der Benutzeroberfläche zur Datenqualitätsvisualisierung zu definieren. Dazu werden Erweiterungen der Datenbankanfragesprache SQL vorgestellt, die ohne Änderungen auch zur Anfrageformulierung in Datenstromsystemen mit Hilfe von CQL Verwendung finden können.

4.2.1. Datenqualität im relationalen Modell

Bisherige Ansätze zur Speicherung von Qualitätsinformationen in Datenbanken vervielfachen das Datenvolumen um die Anzahl der zu verwaltenden DQ-Dimensionen. Um eine effizientere Speicherung und Wartung von Datenqualitätsinformationen im

Datenbanksystem zu unterstützen, wird das traditionelle Metadatenmodell der relationalen Datenbank-Management-Systeme (RDBMS) in Anlehnung an das erweiterte Metadatenmodell des DSMS ergänzt. Die Datenqualitätsfenster des Datenstroms werden auf Partitionen der relationalen Datenbanktabelle (relationale Fenster) abgebildet.

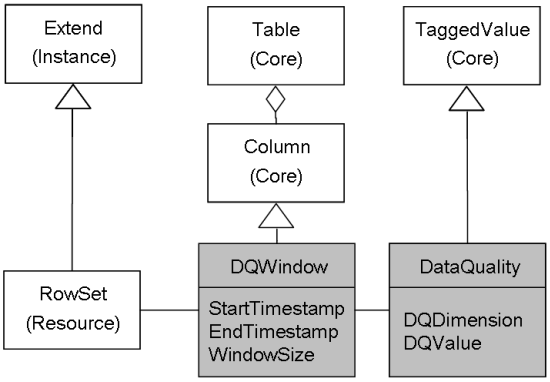


Abbildung 4.4.: Erweiterung des relationalen Metadatenmodells

Abbildung 4.4 zeigt das erweiterte relationale Metadatenmodell in der Notation des Common-Warehouse-Metamodells. Ein relationales Datenqualitätsfenster DQWindow ist als Ausschnitt der Instanzen RowSet einer Tabellenspalte Column definiert. Datenqualitätsinformationen DataQuality werden als TaggedValue modelliert (Vergleich Abbildung 4.3).

Um dieses Modell im RDBMS umzusetzen, wird der neue Tabellentyp der Datenqualitätstabelle eingeführt. Sie werden automatisch mit den Messwerttabellen angelegt, wobei für jedes Messwertattribut eine eigene DQ-Tabelle erzeugt wird. Tabelle 4.1 zeigt das Schema einer Datenqualitätstabelle. Für jede DQ-Dimension wird eine Spalte angelegt. Die abgelegten Datenqualitätsinformationen werden durch Start- und Endzeitstempel des DQ-Fensters identifiziert.

Spaltenname	**Typ**	**Kommentar**
Startzeitstempel	Timestamp	Start des DQ-Fensters
Endzeitstempel	Timestamp	Ende des DQ-Fensters
Genauigkeit	Double	Wert der Genauigkeit des gegebenen Fensters
Vollständigkeit	Double	Wert der Vollständigkeit des gegebenen Fensters
...	...	auf weitere DQ-Dimensionen erweiterbar

Tabelle 4.1.: Schema der Datenqualitätstabelle

Abbildung 4.5 zeigt die DQ-Tabelle (unten links) am Beispiel des Pkw-Recyclings aus Anwendungsszenario 2. Die Sensormessdaten der Öltemperatur werden mit Hilfe der

Dimensionen Genauigkeit und Vollständigkeit beschrieben, so dass die DQ-Tabelle PkwDaten_DQ mit den Spalten Genauigkeit und Vollständigkeit erzeugt wird.

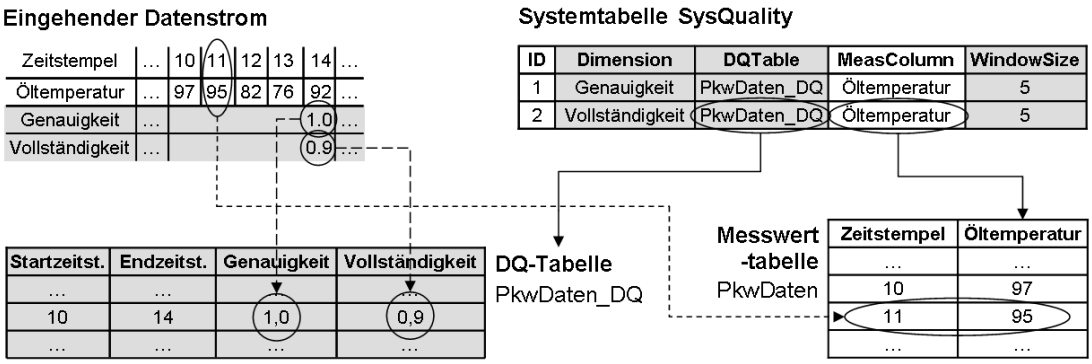


Abbildung 4.5.: Daten- und Qualitätsimport

Zur Gewährleistung eines effizienten Datenqualitätsmanagements ist es notwendig, die Systemtabellen des Datenbankkataloges zu erweitern. Die Katalogtabelle SYSQUALITY (siehe Tabelle 4.2) wird eingeführt. Sie verbindet die Datenqualitäts- und Messwerttabellen über interne Zeiger und enthält Angaben zu den vorhandenen DQ-Dimensionen. Datenqualitätsinformationen, die Messdaten der *Sensordatenspalte* beschreiben, werden in der DQ-Tabelle *DQTable* abgelegt. Die Spalte *Dimension* listet die gespeicherten DQ-Dimensionen anhand kurzer Kennungen auf: A - Genauigkeit, C - Vollständigkeit, E - Konfidenz, D - Datenmenge. Wie in Abschnitt 2.4.2 erläutert, kann die Aktualität während der Datenverarbeitung als Differenz des Messzeitstempels und der aktuellen Systemzeit berechnet werden. Sie muss somit nicht in der Datenbank abgelegt werden.

Spaltenname	Typ	Kommentar
ID	Char[36]	Eindeutiger Identifikator
Dimension	Char[4]	Datenqualitätsdimension A, E, C, D
DQ-Tabelle	Char[36]	Zeiger auf Tabelle mit Datenqualitätsinformationen
Sensordatenspalte	Char[36]	Zeiger zur Spalte, die die Messdaten enthält

Tabelle 4.2.: Schema der Katalogtabelle SYSQUALITY

Um weitere Datenqualitätsdimensionen speichern zu können, müssen lediglich entsprechende Kennungen in die Definition der Katalogtabelle eingefügt werden. Das Schema der Datenqualitätstabelle kann eine beliebige Zahl an Dimensionen aufnehmen.

Abbildung 4.5 zeigt den Datenqualitätsimport aus einem dynamischen Datenstrom in das statische Datenbanksystem. Der Datenstrom wird in Messwertdaten (weiß) und DQ-Informationen (grau) unterteilt. Erstere werden in die zugehörige Messwerttabelle

(PkwDaten) eingefügt. Nachfolgend werden die DQ-Fenster des Datenstroms auf die relationalen Fenster der Datenbank abgebildet. Dazu werden die DQ-Informationen mit Zeitstempeln für Fensterbeginn und -ende versehen und in der Datenqualitätstabelle (PkwDaten_DQ) gespeichert. Wird eine DQ-Dimension, für die in der Datenqualitätstabelle eine Spalte angelegt ist, nicht mit dem Datenstrom verbreitet, wird der fehlende DQ-Wert in der Datenbank durch einen Null-Wert repräsentiert.

Aufgrund der Übernahme des Fensterkonzeptes ist lediglich eine zusätzliche Einfügeoperation pro Datenqualitätsfenster des Datenstroms notwendig. Anforderung 3 des effizienten Imports von Datenqualitätsinformationen ist damit erfüllt.

4.2.2. Erweiterung der Anfragesprache

Um Datenqualitätsinformationen in eine Datenbank zu integrieren, sind Erweiterungen der Data Definition Language (DDL) und Data Manipulation Language (DML) erforderlich. Einersetits müssen SQL-Befehle für das Anlegen von Relationen für Messdaten mit zugeordneten Datenqualitätsdimensionen bereit gestellt werden. Andererseits sind Befehle zur Wartung (z.B. Einfügen, Löschen) der Daten notwendig.

Anlegen der Datenstrukturen

Um das automatische Anlegen der oben beschriebenen Datenqualitätstabellen zu ermöglichen, wurde der CREATE TABLE-Befehl um die optionale WITH DATAQUALITY- Befehlssequenz erweitert (siehe Abbildung 4.6). Die Sequenz ist für jede Spalte anzugeben, für die DQ-Informationen vorliegen. Die Datenqualitätsdimensionen werden im Feld *DQDimension* mit Hilfe ihrer Kennungen A,E,C und/oder D übergeben.

CreateTable

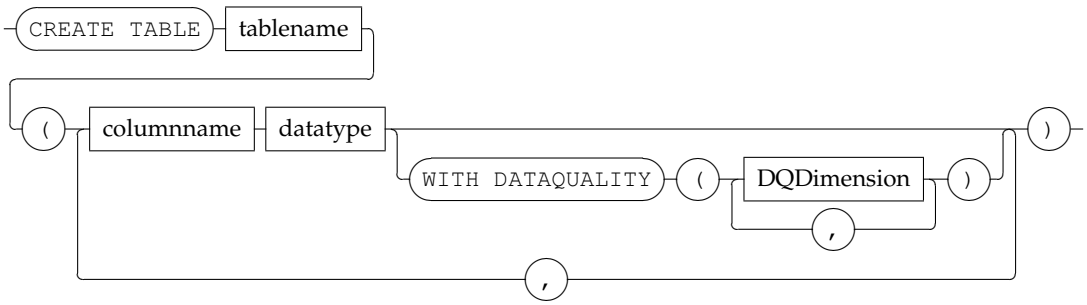


Abbildung 4.6.: Struktur des CREATE TABLE-Befehls

Folgender SQL-Befehl legt eine Messdaten-Tabelle mit drei Spalten an und erstellt zusätzlich zwei Datenqualitätstabellen für die Attribute `Oeltemperatur` und `Kilometerstand`, welche mit unterschiedlichen DQ-Werten gefüllt werden können.

```
CREATE TABLE PkwDaten (
   Zeitstempel timestamp not null,
   Oeltemperatur double WITH DATAQUALITY(A,C,D,E),
   Kilometerstand double WITH DATAQUALITY(A,C,E) )
```

Einfügen der Qualitätsinformationen

Um DQ-Datensätze in die geschaffenen Speicherstrukturen einzufügen, muss ein weiterer Befehl hinzugefügt werden: der INSERT DATAQUALITY INTO-Befehl (siehe Abbildung 4.7). Er ermöglicht die fensterweise Integration der DQ-Datensätze unter Angabe der Messwerttabelle (*tablename*) und des spezifischen Attributs (*columnname*).

InsertDataQuality

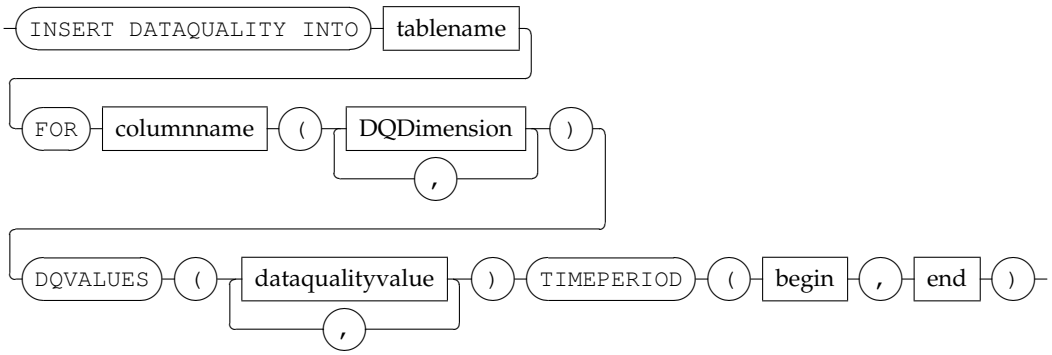


Abbildung 4.7.: Struktur des INSERT DATAQUALITY INTO-Befehls

Die zu speichernden Datenqualitätsdimensionen werden wieder durch die Kennungen spezifiziert und die zugehörigen Werte nach der Befehlssequenz DQVALUES als Parameterliste übergeben. Da die Datenqualitätsdimensionen fensterweise spezifiziert werden, müssen Beginn- und Endzeitstempel der Fenster in der TIMEPERIOD-Sequenz vermerkt werden.

Im folgenden Beispiel werden für die Spalte `Oeltemperatur` der Tabelle `PkwDaten` DQ-Informationen der Dimensionen Genauigkeit (A) und Vollständigkeit (C) für ein entsprechendes Zeitfenster eingefügt.

```
INSERT DATAQUALITY INTO PkwDaten FOR Oeltemperatur(A, C)
DQVALUES (1.2, 0.74)
TIMEPERIOD ('2008-01-29 13:00:00','2008-01-29 13:00:20')
```

Zum Löschen der Datenqualitätsinformationen und Speicherstrukturen wurden zudem die Befehle DROP TABLE und DELETE angepasst.

Abfrage der Qualitätsinformationen

Zum Auslesen der gespeicherten Datenqualitätsinformationen muss die Datenbankanfragesprache DQL (Data Query Language, Teil der SQL) bzw. die Datenstromanfragesprache CQL erweitert werden. Dazu werden Funktionen zur Abfrage jeder verfügbaren Datenqualitätsdimension entwickelt (siehe Tabelle 4.3). Sie können sowohl in der SELECT- als auch in der WHERE-Klausel aufgerufen werden. Außerdem ist die Anfrage der Qualität einzelner Attributspalten bzw. -ströme, algebraisch verknüpfter Attribute oder Attributaggregationen möglich. Um die Liste der auslesbaren Datenqualitätsdimensionen zu erweitern, müssen die entsprechenden Funktionen entworfen und implementiert werden.

DQ-Dimension	Systemfunktion
Genauigkeit	accuracy()
Konfidenz	confidence()
Vollständigkeit	completeness()
Datenmenge	dataVolume()
Aktualität	timeliness()

Tabelle 4.3.: Systemfunktionen zur Qualitätsabfrage

Im folgenden Beispiel wird neben den Sensordaten der numerische Messfehler als Summe des statistischen `confidence()` und systematischen Fehlers `accuracy()` des Attributs `Oeltemperatur` ausgegeben, falls die Vollständigkeit `completeness()` bei mindestens 90% liegt.

```
SELECT Zeitstempel, Oeltemperatur,
 confidence(Oeltemperatur)+accuracy(Oeltemperatur)
FROM PkwDaten
WHERE completeness(Oeltemperatur) > 0.9
```

Bei einer solchen Abfrage der Qualität eines Anfrageergebnisses muss die Datenqualitätsverarbeitung gestartet werden, die in Kapitel 5 beschrieben wird.

Umsetzung der Befehlserweiterungen im Datenbanksystem

Die Integration der Katalogtabelle SYSQUALITY erlaubt die effiziente Zuordnung von Mess- und Qualitätsdaten über interne Zeiger, erfordert jedoch eine Erweiterung der internen Datenbankstruktur. Soll die Datenqualitätsverwaltung ohne Änderung des zugrunde liegenden Metamodells implementiert werden, muss SYSQUALITY als traditionelle Nutzertabelle angelegt werden. Dabei muss auf eine restriktive Rechtevergabe geachtet werden, da ein versehentliches Löschen die Zuordnung von Mess- und Qualitätsdaten unmöglich macht. Außerdem muss auf den traditionellen Tabellenverbund

zurückgegriffen werden, was sowohl Einfügeoperationen als auch das Auslesen von Datenqualitätsinformationen erheblich verlangsamt.

Die Befehlserweiterungen zum Anlegen der Datenstrukturen und Einfügen von Qualitätsinformationen können sowohl außerhalb als auch innerhalb des Datenbanksystems realisiert werden. Im ersten Fall, muss ein Parser vorgeschalten werden, der z.B. den erweiterten CREATE TABLE-Befehl in mehrere traditionelle Befehlsaufrufe zum Anlegen der Messwerttabelle und der zugehörigen Datenqualitätstabellen sowie dem Einfügen der Verknüpfungen in SYSQUALITY übersetzt. Der zweite Fall der internen Umsetzung erlaubt wiederum den effizienten Zugriffs auf den Systemkatalog. Außerdem wird die Verarbeitungszeit durch die Erweiterung des systeminternen Parsers verkürzt. Die Befehlssequenzen werden direkt ausgeführt.

Auch die Funktionen zum Auslesen der Qualitätsinformationen können durch Erweiterungen im Parser realisiert werden. Sie werden dabei durch eigenständige Anfrageknoten ersetzt. Nachfolgend kommt es zu einer speziellen Verarbeitung dieser Knoten, so dass es leicht fällt die komplexe Datenqualitätsverarbeitung zu berücksichtigen. Das Einfügen neuer Anfrageknoten erfordert aber starke Eingriffe in das Gesamtsystem, die nicht immer erlaubt sind.

Eine zweite Möglichkeit ist die Realisierung als Funktionen des Datenbank- oder Datenstromsystems ähnlich der Aggregationsfunktionen $sum()$ oder $average()$. Ist die interne Erweiterung des genutzten Daten-Management-Systems durch neue Systemfunktionen nicht möglich, muss auf nutzerdefinierte Funktionen ausgewichen werden. Systemfunktionen bieten allerdings zumeist einen größeren Funktionsumfang und werden stärker in die Optimierung gestellter Datenbankanfragen einbezogen.

Datenqualitätsinformationen werden in der Regel im Kontext der beschriebenen Messdaten ausgelesen. Zuerst werden wie oben erläutert interne Referenzen der Systemkatalog genutzt, um Messwert- und Datenqualitätstabellen zuzuordnen. Dann müssen die gewünschten Werte ausgelesen werden. Um alle Messwerte eines Datenqualitätsfensters anhand des Start- und Endzeitstempels in der Datenqualitätstabelle zu identifizieren, wird ein Verbund der Messdatentabelle und der benötigten DQ-Tabellen auf Basis des Zeitstempels ausgeführt. Die Performanz dieses Verbundes kann durch eine Indexstruktur optimiert werden, die automatisch im erweiterten CREATE TABLE-Befehl angelegt wird.

Eine detaillierte Diskussion der Umsetzung der vorgestellten Datenqualitätsalgebra in relationalen Datenbanksystemen findet sich in [Sei08].

4.3. Adaption der Fenstergröße

Die Definition der Fenstergröße stellt eine schwierige Aufgabe bei der Arbeit mit dem vorgestellten Datenqualitätsmodell DQMx dar. Ein Kompromiss zwischen großen Fenstern, die niedrigen Ressourcenverbrauch garantieren, und kleinen Fenster mit fein-

granularen, präzisen DQ-Informationen muss gefunden werden. Außerdem muss die Fenstergröße während der Laufzeit an sich ändernde Datenstrom- sowie Datenqualitätseigenschaften angepasst werden.

Die Adaption der Fenstergröße, dargestellt in Abbildung 4.8, kann an jedem Punkt in der Operatorenkette der Datenstromverarbeitung erfolgen. Wenn eine Anpassung der Fenstergröße notwendig ist, wird die berechnete Fenstergröße an die Sensorknoten der Datenquellen übermittelt, um die folgenden Datenqualitätsfenster mit der neuen Größe zu initialisieren. Sollen die Datenqualitätsfenster vergrößert werden, können alle Fenster sofort mit der neuen Fenstergröße versehen werden. Die Fensterdatenqualitäten der bisher kleinen DQ-Fenster werden zu größeren Fenstern zusammengefasst. Ist eine Fensterverkleinerung angezeigt, erfolgt die Fenstergrößenanpassung verzögert, da die vorhandenen grobgranularen DQ-Informationen keine Rückschlüsse auf feingranulare DQ-Werte erlauben. Erst nach Benachrichtigung der Sensorknoten stehen DQ-Fenster mit verkleinerter Größe zur Verfügung.

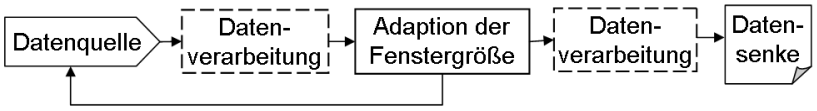


Abbildung 4.8.: Adaption der Größe der Datenqualitätsfenster

Im Folgenden werden zwei Verfahren zur dynamischen Adaption der Größe der Datenqualitätsfenster vorgestellt. Zuerst wird die Fenstergröße auf Basis der Güte der Datenqualitätsinformationen bestimmt. Danach wird die Fensteranpassung mit Hilfe der Interessantheit des Datenstroms vorgestellt.

4.3.1. Berechnung der Datenqualitätsgüte

Bei der Zusammenfassung der Datenqualitätsinformationen in DQ-Fenstern gehen Informationen verloren. Dadurch ist kein Zurückrechnen auf die ursprünglichen DQ-Informationen der Einzelelemente möglich, sondern jeder Datenwert wird einheitlich mittels der aggregierten Datenqualität beschrieben. Die Abweichung zwischen den ursprünglichen tupelweisen DQ-Informationen und der aggregierten Fensterqualität bestimmt die Güte der Qualitätsabschätzung.

Definition 4.1 Der Datenqualitätsfehler $\Delta q_w(k)$ eines Datenqualitätsfensters $w(k) = [t_b, t_e]$ bezeichnet die mittlere euklidische Distanz[1] zwischen DQ-Informationen der einzelnen Datenelemente $q(j)(t_b \leq j \leq t_e)$ und des aggregierten Qualitätswertes $q_w(k)$.

[1] Die euklidische Distanz ist ein typisches Abstandsmaß für metrisch skalierte Daten.

$$\Delta q_w(k) \quad = \quad \sqrt{\frac{1}{\omega}\sum_{j=t_b}^{t_e}(q(j)-q_w(k))^2} \tag{4.5}$$

Bildlich beschrieben, werden positive und negative Ausreißer der Qualitätsinformation durch die fensterweise Aggregation geglättet. Je stärker die Datenqualitäten streuen, umso größer ist die durchschnittliche Distanz zwischen Tupel- und Fensterqualität und umso größer ist der Datenqualitätsfehler. Besonders interessant sind Grenzfälle des DQ-Fehlers bei unterschiedlichen Fenstergrößen. Für große DQ-Fenster ($\omega \rightarrow \infty$) konvergiert Δq_w gegen einen Grenzwert, der durch die Wahrscheinlichkeitsverteilung der ursprünglichen Datenqualitätsinformationen $q(j)$ bestimmt wird. Andererseits führen kleine Fenster zu kleineren Datenqualitätsfehlern. Für den Grenzfall $\omega = 1$ verschwinden sie ganz.

Wird beispielsweise ein Datenqualitätsfenster mit vier Druckmesswerten $p_1 ... p_4$ und den Tupelgenauigkeiten $a_1 = 3,0bar; a_2 = 3,3bar; a_3 = 2,7bar; a_4 = 2,8bar$ betrachtet, so beträgt die aggregierte Fenstergenauigkeit $a_w = 2,95bar$ für die Fenstergröße $\omega = 4$. Der Genauigkeitsfehler ist $\Delta a_w = 0,2bar$. Untersucht man hingegen die Fenster der Größe $\omega = 2$, z.B. $[a_1; a_2][a_3; a_4]$, so wird die durchschnittliche Abweichung der Genauigkeit zu $\Delta a_w = 0,1bar$ reduziert. Die Reduktion erfolgt jedoch nicht nur beim obigen Beispiel; $\Delta a_{\omega=2} \leq \Delta a_{\omega=4}$ gilt für alle Fensterkombinationen der Größe $\omega = 2$.

Die Beobachtung, dass kleine Fenster kleine Datenqualitätsfehler bedingen, kann zu einem Theorem verallgemeinert werden, das die Basis für die automatische Adaption der Fenstergröße mit Hilfe des Datenqualitätsfehlers bietet.

Theorem 4.1 In einem gegebenen Fenster $\hat{w}$ ist der Datenqualitätsfehler $\Delta\hat{q}_w$ immer größer oder gleich dem Durchschnitt der Datenqualitätsfehler $\Delta q_w(k)$, die durch eine beliebige Aufteilung von $\hat{w}$ in kleinere ($\omega < \hat{\omega}$) nicht-überlappende Teilfenster $w(k)$, $1 \leq k \leq \kappa$ bestimmt werden.

Beweis Theorem 4.1 folgt aus dem Gesetz der vollständigen Varianz in Gleichung 4.6, das für beliebige Zufallsvariablen X und Y derselben Wahrscheinlichkeitsverteilung gilt, sofern X eine endliche Varianz aufweist.

$$\mathrm{Var}(X) \quad = \quad E[\mathrm{Var}(X|Y)] + \mathrm{Var}(E[X|Y]) \tag{4.6}$$

X sei die ursprüngliche Datenqualitätsinformationen $q(j)$ des Fensters w; Y bestimmt die Aufteilung von X in κ nicht-überlappende Teilfenster $w(k)$. Wird die Wahrscheinlichkeitsverteilung als diskrete Gleichverteilung angenommen, so reduzieren sich die Erwartungswerte $E[X|Y]$ im zweiten Term auf der rechten Seite von Gleichung 4.6 zu den durchschnittlichen fensterweisen DQ-Informationen $q_w(k)$ und die Varianzen $\mathrm{Var}(X|Y)$ im ersten Term auf der rechten Seite von Gleichung 4.6 zu Quadraten der

Datenqualitätsfehler $[\Delta q_w(k)]^2$. Der Beweis erfolgt durch Substitution in Gleichung 4.6 und $\mathrm{Var}(E[X|Y]) \geq 0$. □

Theorem 4.1 gilt nur für den durchschnittlichen Datenqualitätsfehler. Die Verkleinerung der Datenqualitätsfenster kann zu einer Vergrößerung einzelner DQ-Fehler führen, garantiert jedoch die Verkleinerung des Durchschnitts aller betrachteter Fenster im Datenstrom. Außerdem zeigt der obige Beweis, dass die Reduzierung des DQ-Fehlers stark von der Varianz der ursprünglichen Datenqualitätsinformationen abhängt. Die Verkleinerung der Datenqualitätsfenster wird einen größeren Einfluss haben, je stärker die Datenqualitätswerte streuen.

Signifikante Streuungen der Datenqualität sind nicht an den Sensorknoten, sondern in der nachfolgenden Verarbeitung der Messdaten sichtbar. Zur Definition der optimalen Fenstergröße muss der Datenqualitätsfehler deshalb nicht am Sensor, sondern in der Verarbeitungskette bestimmt werden.

Die Varianz der tupelbasierten DQ-Informationen $\mathrm{Var}(X|Y)$ ist ein guter Indikator des Datenqualitätsfehlers. Leider kann die Varianz nicht durch die Operatoren der Datenverarbeitungskette verfolgt werden. Zum Beispiel ist es nicht möglich, die Varianz eines Selektionsergebnisses auf Basis der eingehenden Varianz zu ermitteln. Ebensowenig können Qualitätsfluktuationen, die durch den Verbund mehrerer Datenströme entstehen, berechnet und kontrolliert werden.

Um dieses Problem zu lösen, wird ein neues Datenstromtupel eingeführt: das Qualitätskontrolltupel τ_c. Es umfasst die tupelbasierten DQ-Informationen und erlaubt so zusammen mit den fensterbasierten Datenqualitätswerten in τ_q die Kontrolle des Datenqualitätsfehlers an jedem Punkt der Datenstromverarbeitung.

4.3.2. Kontrolle der Datenqualitätsgüte

Die in diesem Abschnitt vorgestellte Datenqualitätskontrolle adaptiert die Größe der Datenqualitätsfenster, so dass der Datenqualitätsfehler den gewünschten Schwellwert nicht übersteigt, während ein minimales zusätzliches Datenvolumen benötigt wird.

Nach der Definition des DQ-Kontrolltupels, wird ein allgemeiner Überblick über die Qualitätskontrolle gegeben. Um verschiedene Datenqualitätsdimensionen zu vergleichen, werden die Fehler mit Methoden der Statistik normalisiert. Des Weiteren dient die Prozesskontrolle zur Stabilisierung der Fenstergrößenkonfiguration.

Qualitätskontrolltupel

Aufgrund des hohen Datenvolumens ist es nicht möglich DQ-Informationen für jedes Tupel bereit zu stellen (siehe Abschnitt 4.1.1). Qualitätskontrolltupel τ_c werden deshalb mit der konstanten Kontrollrate r_c zufällig im Datenstrom eingestreut (siehe Abbildung 4.9, dunkelgrau). Sie bilden nur einen geringen Anteil der Datenstromtupel, so dass ein Kom-

promiss zwischen zusätzlichem Datenvolumen und der DQ-Fehlerkontrolle gefunden werden kann.

		τ	τc		τq					τc	τq	τc			τq					τc = τq	
Zeitstempel	…	210	211	212	213	214	215	216	217	218	219	220	221	222	223	224	225	226	227	228	229	…
Druckverlust	…	180	178	177	175	176	181	189	201	204	190	194	192	189	183	215	210	211	199	187	184	…
Genauigkeit	…		3,5							3,8		2,3									1,9	…
Konfidenz	…		12,5							13,4		12,4									11,5	…
Vollständigkeit	…		0,9							0,8		0,9									1	…
Datenmenge	…		2							2		2									2	…
Genauigkeit	…					3,0					3,9					2,78					2,86	…
Konfidenz	…					12,92					13,34					12,1					12,12	…
Vollständigkeit	…					0,9					0,8					0,9					0,99	…
Datenmenge						2					2					2					2	

Abbildung 4.9.: Datenqualitäts- und Qualitätskontrolltupel

Definition 4.2 Das Qualitätskontrolltupel τ_c besteht aus Zeitstempel t, Messwert(en) x_i und je einem Datenqualitätswert q pro DQ-Dimension Q und Messwertattribut A_i, wobei $1 \leq i \leq n$.

Die Kontrolltupel werden mit Hilfe eines Bernoulli-Samplings mit der Samplingrate (Kontrollrate) r_c bestimmt. Entsprechend der Kontrollrate enthält ein Datenqualitätsfenster 0 bis ω Kontrolltupel, mit deren Hilfe der durchschnittliche Qualitätsfehler $\Delta q(w)$ für jede DQ-Dimension q geschätzt wird.

Definition 4.3 Der geschätzte Datenqualitätsfehler δq_w einer Dimension q im Fenster w ist als durchschnittliche euklidische Distanz zwischen der Fensterdatenqualität des Datenqualitätstupels τ_q und den tupelweisen DQ-Informationen der in w enthaltenen Qualitätskontrolltupel τ_c definiert.

$$\delta q_w = \frac{1}{|\tau_c|} \sum_{\forall \tau_c \in w} (q(\tau_c) - q(\tau_q))^2 \tag{4.7}$$

Qualitätskontrolloperator

Die Datenqualitätskontrolle, dargestellt in Abbildung 4.10, berechnet zuerst den Datenqualitätsfehler für jede DQ-Dimension und jedes DQ-Fenster. Um verschiedene DQ-Dimensionen zusammenzufassen, werden die Qualitätsfehler normalisiert und transformiert. Schließlich werden Methoden der statistischen Prozesskontrolle zur Ableitung der optimalen Fenstergröße angewendet.

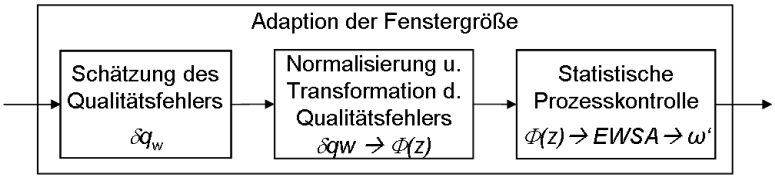


Abbildung 4.10.: Fenstergrößenadaption mit Hilfe des Datenqualitätsfehlers

Normalisierung des Qualitätsfehlers

Mit Hilfe der Qualitätskontrolltupel kann der Datenqualitätsfehler für jede Dimension geschätzt werden. Diese Schätzungen müssen in zwei Schritten normalisiert und transformiert werden, um Standardtechniken der Prozesskontrolle auf die Datenstromverarbeitung anwenden zu können.

Zuerst werden die Qualitätsfehlerschätzungen für die unterschiedlichen DQ-Dimensionen normalisiert. Die oft unbekannte Wahrscheinlichkeitsverteilung der Fehlerschätzungen kann durch die Normalverteilung angenähert werden. Mittelwert μ und Standardabweichung σ hängen dabei von der Qualitätsdimension ab. Zur Normalisierung der Schätzungen wird die Transformation $\delta q \rightarrow z = [\delta q - \mu(\delta q)]/\sigma(\delta q)$ angewendet. Die normalisierten DQ-Fehlerschätzungen z folgen der Standardnormalverteilung. In der prototypischen Umsetzung werden μ und σ iterativ für jede DQ-Dimension eines eintreffenden DQ-Fensters aktualisiert.

Der zweite Schritt bildet z auf die kumulative Normalverteilung Φ ab: $z \rightarrow \hat{z} \equiv \Phi(z)$. Entsprechend der Definition der kumulativen Verteilung (Integral der Wahrscheinlichkeitsdichte) sind die transformierten Fehlerschätzungen nun gleich verteilt im Intervall $[0;1]$. Die Transformation zur kumulativen Normalverteilung kann effizient implementiert werden [AS64], so dass eine effektive Realisierung des vorgeschlagenen Normalisierungs- und Transformationsprozesses möglich ist.

Wird zum Beispiel der geschätzte Genauigkeitsfehler $\delta a = 0,5$ mit dem Mittelwert $\mu(\delta a) = 0,43$ und der Standardabweichung $\sigma(\delta a) = 0,09$ betrachtet, so wird der normalisierte standardisierte Genauigkeitsfehler zu $\hat{z} = \Phi((0,5 - 0,43)/0,09) = 0,78$ berechnet.

Berechnung und Kontrolle des Fehlertrends

Um den Datenqualitätsfehler zu kontrollieren und eventuell zu korrigieren, werden Prozesskontrolltechniken auf die normalisierten und transformierten Fehlerschätzungen angewendet. Der exponentiell gewichtete, geglättete Durchschnitt $EWSA$ (engl. exponentially weighted smoothed average) der geschätzten Fehler bildet einen statistisch gültigen

Fehlertrend. Er wird jeweils mit Hilfe der Fehlerschätzungen aller DQ-Dimensionen eines Fensters wie folgt aktualisiert.

$$EWSA_{i+1} = \beta \cdot \hat{z}_{i+1} + (1-\beta) \cdot EWSA_i \tag{4.8}$$

Der Parameter β beschreibt die Empfindlichkeit des Trends für Änderungen der DQ-Fehler. Typische Werte der Prozesskontrolle sind $\beta \leq 0,05$. Der Fehlertrend $EWSA$ ist normal verteilt um den Mittelwert $\mu = 0,5$ mit der Standardabweichung $\sigma(EWSA) = 2 \cdot \sqrt{1/12 \cdot \beta/(2-\beta)}$. Gleichung 4.8

Zur Kontrolle des Datenqualitätsfehler wird die Fenstergröße angepasst, wenn der DQ-Fehlertrend das Kontrollintervall $[lowerErrorBound; upperErrorBound]$ und den Schwellwert $s_{\delta q}$ des akzeptierten Datenqualitätsfehlers verlässt.

$$upperErrorBound = s_{\delta q} \tag{4.9}$$
$$lowerErrorBound = s_{\delta q} - \rho \cdot \sigma(EWSA) \tag{4.10}$$

wobei ρ das $(1 - p/2)$-Quantil der Wahrscheinlichkeit p beschreibt, mit der $EWSA$ im Intervall verbleibt.

Ist der Qualitätsfehler zu groß, wird die obere Intervallgrenze $upperErrorBound$ verletzt, und das Qualitätsfenster muss verkleinert werden. Ist der Fehler sehr klein, wird die untere Intervallbeschränkung $lowerErrorBound$ unterschritten und die Fenster können vergrößert werden, um Datenvolumen einzusparen.

Wurden zum Beispiel $p = 0,95$, $\beta = 0.04$ und der (normalisierte und transformierte) akzeptierte Schwellwert $s_{\delta q} = 0,47$ gewählt, liegen der aktuelle Fehlertrend bei $EWSA_i = 0,4$ und der geschätzte Genauigkeitsfehler bei $\delta a = 0,5$ (s.o.), so beträgt der aktualisierte Fehlertrend $EWSA_{i+1} = 0.42$ und liegt damit im Kontrollintervall $[0,47; 0,392]$. Die Fenstergröße bleibt konstant, obwohl die Einzelbeobachtung des normalisierten, transformierten Genauigkeitsfehlers $\hat{z} = 0,78$ das Kontrollintervall überschreitet. Der geglättete Trend $EWSA$ hält die Fenstergröße statistisch stabil und verhindert so eine Schwingung mit monoton wachsender Amplitude.

Kleine Werte des DQ-Fehlerschwellwertes $s_{\delta q}$ erzwingen kleinere Schwankungen zwischen Tupel- und Fensterdatenqualität. Somit führt ein niedriger Akzeptanzschwellwert zu kleinen Datenqualitätsfenstern und umgekehrt.

Anpassung der Fenstergröße

Sobald der Fehlertrend das Kontrollintervall verlässt, muss die Fenstergröße angepasst werden. Die Fenstervergrößerung bzw. -verkleinerung muss so erfolgen, dass danach das Intervall wieder eingehalten werden kann. Zuerst wird die benötigte Verkleinerung

bzw. Vergrößerung des DQ-Fehlers Δq_{req} berechnet. Dann kann die neue Fenstergröße ω' abgeleitet werden.

Die Inversion des Fehlertrends $EWSA$ definiert den maximalen Fehler für eine Fensterverkleinerung, wenn die obere Intervallgrenze $upperErrorBound$ überschritten wurde. Der minimale Fehler für eine Fenstervergrößerung wird nach demselben Schema berechnet.

$$z_{req} = \Phi^{-1}\left(\frac{upperErrorBound - (1-\beta)\cdot EWSA_i}{\beta}\right) \tag{4.11}$$

Experimente zeigen, dass der Datenqualitätsfehler δq logarithmisch mit der Fenstergröße ω ansteigt, unabhängig von Beispielanwendung oder Datenstromeigenschaften. Es gilt $\delta q = ln(\omega) \pm \lambda$. Die Rücktransformation von z_{req} zu δq_{rec} ergibt die aktualisierte Fenstergröße ω' für den benötigten DQ-Fehler δq_{req}, den tatsächlichen Fehler δq und die bisherige Fenstergröße $\omega : \omega' = \omega \cdot e^{\delta q_{rec} - \delta q}$.

Zusammenfassung

Die Größe der Datenqualitätsfenster wird dynamisch während der Datenstromverarbeitung angepasst, um den Datenqualitätsfehler unter einem vordefinierten akzeptierten Fehlerschwellwert zu halten. Sobald der Qualitätskontrolloperator eine Über- bzw. Unterschreitung des statistisch stabilen Kontrollintervalls feststellt, wird die Fenstergröße in dem Maße verkleinert bzw. vergrößert, das den Fehlertrend in einen stabilen Zustand zurückführt.

4.3.3. Interessantheit des Datenstroms

Soll die Größe der Datenqualitätsfenster ohne zusätzliches Datenvolumen der Qualitätskontrolltupel überwacht werden, bietet die *Interessantheit* des Datenstroms einen guten Ansatzpunkt. Kritische Datenstrombereiche (starke Messwertanstiege, Fluktuationen, etc.) sind für die Datenauswertung von besonderem Interesse und werden mit kleinen Datenqualitätsfenstern überwacht. Konstante Datenwerte hingegen werden in großen Fenstern beschrieben, um das Datenvolumen zu reduzieren. Ebenso können die Datenqualitätsinformationen selbst zur Bestimmung der Fenstergröße herangezogen werden. Zum Beispiel wird die Fenstergröße verkleinert, wenn geringe Datenqualität beobachtet wird, die bei der Datenauswertung besonders beachtet werden muss.

Abbildung 4.11 zeigt das Konzept der Fenstergrößenanpassung auf Basis der Interessantheit des Datenstroms. Zuerst wird die Interessantheit des Datenstroms bestimmt und nachfolgend normalisiert. Ausgehend von der Distanz zum vordefinierten Schwellwert der Interessantheit wird die Fenstergröße aktualisiert. Das allgemeine Vorgehen ist

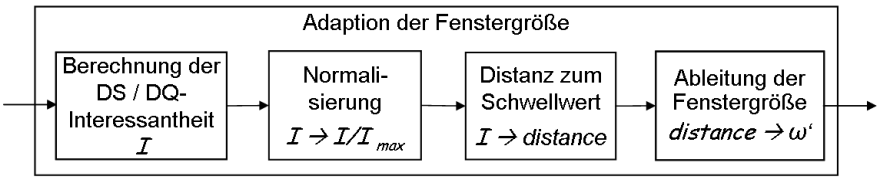


Abbildung 4.11.: Fenstergrößenadaption mit Hilfe der Interessantheit

in Algorithmus 4.1 dargestellt. Er bildet das Rahmenwerk für eine Vielzahl von Anwendungen, die durch spezifische Definitionen der Funktionen *getInterestingness()* und *updateSize()* bestimmt werden.

Algorithmus 4.1 : Fenstergrößenadaption auf Basis der Interessantheit

Input : D Datenstrom, w Fenstergröße, s_I Schwellwert der Interessantheit
Output : w' aktualisierte Fenstergröße

```
1 while D.hasNextWindow() do
2 |   I = getInterestingness(D);
3 |   distance = |I − s_I|;
4 |   w' = updateSize(distance, w);
5 end
```

Berechnung der Interessantheit

Die Berechnung der Interessantheit hängt stark vom Anwendungskontext ab. Die Funktion *getInterestingness()* kann mit Hilfe verschiedener Datenstromoperatoren definiert werden. Im Folgenden werden vier Beispielausprägungen erläutert.

currentValue() Um außergewöhnliche Messwerte, z.B. sehr hohe Temperaturen, zu erkennen wird kein zusätzlicher Operator benötigt.

slidingSlope() Extreme Messwertänderungen, wie das plötzliche starke Ansteigen eines Sensormesswertes, werden mit Hilfe der Anstiegsberechnung (siehe Abschnitt 5.4.5) überwacht. Zum Beispiel muss der Öldruck in einem Hydraulikzylinder zur frühzeitigen Erkennung eines Druckabfall durch Dichtungsverschleiß beobachtet werden. Überschreitet der Anstieg den kritischen Schwellwert, müssen die Datenqualitätsfenster verkleinert werden, damit eine genaue Datenanalyse erfolgen kann.

fft() Starke Messwertschwankungen sind ein weiterer Indikator zur Erkennung wichtiger Datenstrombereiche. Um Unstetigkeiten und ungewöhnliche Schwankungen festzustellen wird die Spektralanalyse mit Hilfe der Fast Fourier Transformation (FFT, siehe Abschnitt 5.3.3) vorgenommen. So können zu hohe Frequenzamplituden erkannt und mit kleinen Datenqualitätsfenstern bewertet werden.

fftSlope() Schließlich können Verschiebungen der Signalfrequenzen Aufschluss über interessante Strompartitionen geben. Die FFT wird mit der Anstiegsberechnung kombiniert, um signifikante Unregelmäßigkeiten der Frequenzbänder aufzudecken. So kann eine komplexe Maschinenkennlinie mit Hilfe einer Vielzahl an Sensoren aufgezeichnet werden. Die stetig wiederholten Produktionsprozesse spiegeln sich in charakteristischen Perioden der Kennlinie wider. Eine Verschiebung dieser Perioden ist ein eindeutiges Signal für mögliche Maschinenausfälle und muss zu einer Verkleinerung der Datenqualitätsfenster führen, um die detaillierte Auswertung der Messdaten zu ermöglichen.

Zur Vereinfachung der Definition der Interessantheit I und des Schwellwertes s_I werden beide in den Wertebereich $[0;1]$ transformiert. Sie werden in Relation zur maximalen Interessantheit I_{max} gesetzt, wobei I_{max} für jeden Interessantheitsindikator getrennt berechnet werden muss.

$$I' = \frac{I}{I_{max}} \qquad s'_I = \frac{s'_I}{I_{max}} \tag{4.12}$$

Die maximale Interessantheit außergewöhnlicher Messwerte ist durch den maximalen Wertebereich der verwendeten Sensoren gegeben $I_{max} = x_{max}$. Der maximale Anstieg kann mit Hilfe dieser Wertebereichsgrenze und der kleinsten Zeitschrittweite des Systems berechnet werden: $I_{max} = x_{max}/\Delta t_{min}$. Die maximale Frequenz für den Indikator *fft()* ist durch das Nyquist-Shannon-Abtasttheorem $f_{abtast} > 2f_{max}$ definiert [KJ05]. Die Abtastfrequenz f_{abtast} des Sensors entspricht der aktuellen Datenrate r des Datenstroms. Das Signalfrequenzband ist damit durch $I_{max} = f_{max} < r/2$ nach oben begrenzt. Die Berechnung der maximalen Interessantheit des Indikators fftslope() ergibt sich aus maximaler Frequenz und minimaler Zeitschrittweite: $I_{max} = f_{max}/\Delta t_{min}$.

Nach der Normalisierung von Interessantheit und Schwellwert kann ihre Distanz *distance* im Intervall $[-1,1]$ bestimmt werden (siehe Algorithmus 4.1, Zeile 3) und die Aktualisierung der Fenstergröße steuern.

Aktualisierung der Fenstergröße

Bei der Aktualisierung der Fenstergröße mit Hilfe der Funktion *updateSize()* können zwei grundlegende Verfahren unterschieden werden. Zum Einen kann die neue Fenstergröße unabhängig von der aktuellen gewählt werden. Zum Anderen kann die zeitliche Entwicklung der Fenstergröße und Schwellwertüberschreitung in Betracht gezogen werden.

Die zeitlich unabhängige Definition der Fenstergröße nutzt die maximal und minimal erlaubte Fenstergröße, die die Verzögerung der DQ-Auslieferung bzw. das zusätzliche Datenvolumen beschränkt. Alle Datenstrompartitionen, deren Interessantheit den Schwellwert übertrifft ($distance \geq 0$), müssen mit der minimalen Fenstergröße ω_{min} überwacht

werden. Das minimale Interesse, d.h. die minimale Distanz $distance = -1$, wird der maximalen Fenstergröße ω_{max} zugeordnet. Die übrigen Fenstergrößen $-1 < distance < 0$ werden wie folgt interpoliert.

$$\omega' = (\omega_{min} - \omega_{max}) \cdot distance + \omega_{min} \tag{4.13}$$

Die Validierung in Abschnitt 7.2 schlägt die Grenzen der Fenstergröße $\omega_{min} = 10$ und $\omega_{max} = 100$ vor, so dass gilt $\omega' = -90 \cdot distance + 10$.

Die zeitliche Entwicklung der Fenstergröße geht von der Annahme aus, dass die untersuchte Datenstrompartition umso interessanter ist, je länger der Interessantheitsindikator den Schwellwert übersteigt. Die Verkleinerung bzw. Vergrößerung der Datenqualitätsfenster wird fortgesetzt, so lange der Schwellwert über- bzw. unterschritten wird und die minimale bzw. maximale Fenstergröße nicht erreicht ist. Die Distanz gibt hier die Schrittweite der Änderung der Fenstergröße vor. Je größer die Distanz, umso interessanter bzw. uninteressanter ist der aktuelle Stromabschnitt, und umso stärker wird die Fenstergröße vergrößert bzw. verkleinert.

$$w' = w \cdot (1 - distance) \tag{4.14}$$

Datenstromfenster werden nicht nur zur Verwaltung von Datenqualitätsinformationen genutzt. Viele Operatoren der Datenstromverarbeitung arbeiten ebenfalls auf Fenstern. So erfolgen zum Beispiel die Aggregation und die Verbundberechnung in Datenstromsystemen zumeist in gleitenden Datenstromfenstern. Das in diesem Abschnitt beschriebene Verfahren zur Definition der Größe von Datenstromfenstern kann ohne Einschränkungen auf diese Anwendungsgebiete übertragen werden. Zum Beispiel können uninteressante Bereiche eines beliebigen Datenstroms zu größeren Fenstern zusammengefasst werden, während kritische Strompartitionen mit Hilfe von feingranularen Aggregationsergebnissen beschrieben werden.

4.4. Zusammenfassung

Die fensterbasierte Datenqualitätsübertragung des Modells DQMx löst den Konflikt zwischen beschränkten Hardware-Ressourcen und dem Wunsch nach feingranularen Datenqualitätsinformationen. Die Datenqualität wird in Datenstromfenstern zusammengefasst, so dass das zusätzliche Datenvolumen auf ein vertretbares Maß reduziert werden kann. Zur optimalen Ausnutzung des DQMx wurden zwei Methoden zur dynamischen Adaption der Fenstergröße entwickelt. Die fensterbasierte Datenqualitätsmodellierung erfüllt damit Anforderung 2 aus Abschnitt 2.5.

Für die Datenqualitätsdimensionen Genauigkeit, Konfidenz, Vollständigkeit und Datenmenge wurden Methoden zur Initialisierung der Datenqualitätsfenster angegeben.

Die Aktualität kann direkt aus der Differenz der Systemzeit und des Zeitpunkts der Messung berechnet werden. Das vorgestellte Modell erfüllt Anforderung 1.

Die vorgestellten Erweiterungen des relationalen Metadatenmodells sowie der SQL-Befehle erlauben die persistente Speicherung von DQ-Informationen in einer Datenbank, um Qualitätsbewertungen bei nachfolgenden Data-Mining-Analysen zu unterstützen. Das Konzept der Datenqualitätsfenster wird mit Hilfe von relationalen Fenstern im Datenbank-Management-System abgebildet, so dass ein effizienter Daten- und Qualitätsimport aus eingehenden Datenströmen möglich ist (siehe Anforderung 3).

Abbildung 4.12 fasst die Anforderungen zusammen, die durch das Datenqualitätsmodell DQMx in diesem Kapitel erfüllt wurden. Des Weiteren wurde für jede Datenqualitätsdimension eine Systemfunktion definiert, um die Qualitätsinformationen in einer Datenbank- oder Datenstromanfrage auszulesen. Diese Systemfunktionen veranlassen die Datenqualitätsverarbeitung entsprechend der auszuführenden Anfrageoperatoren, die ausführlich im folgenden Kapitel beschrieben wird.

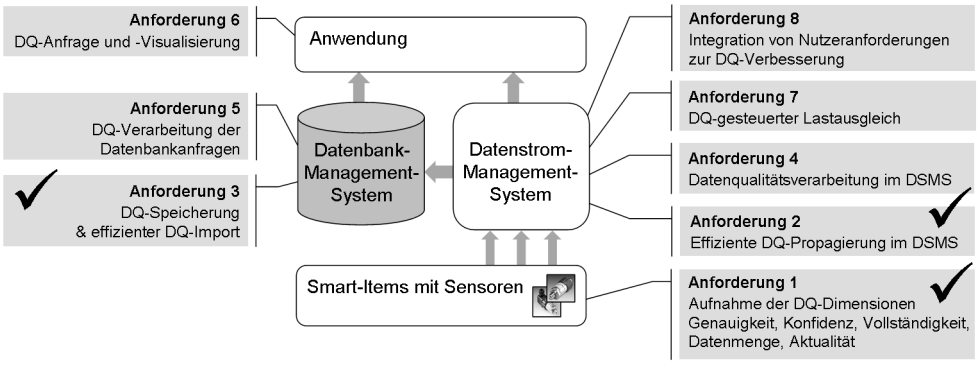


Abbildung 4.12.: Erfüllte Anforderungen 1, 2 und 3

5 Verarbeitung von Datenqualitätsinformationen

Sowohl in Datenstrom- als auch in Datenbank-Management-Systemen unterliegen die gesammelten Rohdaten zahlreichen Verarbeitungsschritten um Wissen zu gewinnen. Im Datenstromsystem einer Smart-Item-Umgebung spielt außerdem die Reduktion des Datenvolumens eine wichtige Rolle. Zur Bestimmung der Datenqualität der Datenverarbeitungsergebnisse müssen die Einflüsse der unterschiedlichen Datenoperatoren auf die verschiedenen Datenqualitätsdimensionen untersucht werden.

In diesem Kapitel wird zuerst das Konzept der Datenqualitätsalgebra für Sensordaten vorgestellt. Danach werden verschiedene Klassen von Datenoperatoren analysiert. Abschnitt 5.2 befasst sich mit Operatoren der mathematischen Algebra. Operatoren der Signalverarbeitung werden in Abschnitt 5.3 untersucht. Abschnitt 5.4 erläutert die Qualitätsverarbeitung bei relationalen Datenbankoperatoren der SQL sowie Datenstromoperatoren der CQL. Die entwickelte Datenqualitätsalgebra basiert auf dem Datenqualitätsmodell DQMx und unterstützt die fensterbasierte Propagierung und Speicherung von Datenqualitätsinformationen.

5.1. Definition der Datenqualitätsalgebra

In einem Smart-Item-System dienen Sensoren als Datenquellen. Sie liefern eine Menge von Rohdaten X (siehe Abbildung 5.1). Die Sensordaten werden mit Hilfe einer Funktion $F(X)$ verarbeitet, um ein Ergebnis Y zu berechnen. Die Funktion F kann dabei u.a. durch

eine CQL- bzw. SQL-Anfrage, ein Excel-Makro oder eine mathematische Formel gegeben sein.

Abbildung 5.1.: Konzept der Datenqualitätsverarbeitung

Die Funktion F wird als Operatorenmenge Θ aufgefasst. Θ stellt eine Teilmenge der Menge aller definierten Operatoren O dar, wobei doppelte Einträge aus O in Θ erlaubt sind. Jeder Operator wird hinsichtlich der Daten- sowie der Datenqualitätsverarbeitung alleinstehend und unabhängig von vorherigen oder nachfolgenden Operationen betrachtet. Neben den Messdaten liefern die Sensoren Informationen über deren Datenqualität DQ_X. Zur Berechnung der Qualität DQ_Y eines Anfrageergebnisses Y muss eine Funktion F^{DQ} gefunden werden, die die Eingangsqualitäten verarbeitet und zusammenfasst.

Ziel ist es, für alle Operatoren $o \in O$ einen entsprechenden Datenqualitätsoperator für die Datenqualitätsdimensionen Genauigkeit, Konfidenz, Vollständigkeit und Datenmenge zu finden. Die erweiterten Operatoren o' können dann zu jeder beliebigen Funktion $F' = F \otimes F^{DQ}$ zusammengesetzt werden, die sowohl die Daten- als auch die Datenqualitätsverarbeitung umfasst. Diese Operatoren bilden die gesuchte Datenqualitätsalgebra.

Definition 5.1 Die Datenqualitätsalgebra Ω ist definiert durch die Menge der erweiterten Operatoren $o' \in O'$ bestehend aus $o \in O$ und $o^{DQ} \in O^{DQ}$, wobei

- jeder Operator o die Messdaten eines oder mehrerer Datenattribute verarbeitet: $y = o(x_i)$ für $1 \leq i \leq n$
- und jeder Datenqualitätsoperator o^{DQ} die Ergebnisqualität aus den Datenqualitätsinformationen und den zugehörigen Messdaten berechnet: $DQ_y = o^{DQ}(DQ_{x_i}, x_i)$ für $1 \leq i \leq n$.

Welche Operatoren zur Verarbeitung von Sensordaten in Frage kommen, wurde bereits in Abschnitt 2.3 eingehend untersucht. Der Aufbau der Abschnitte 5.2 bis 5.4 richtet sich nach der dort vorgenommenen Klassifizierung der Operatoren.

Die Operatoren der Datenverarbeitung sind durch die Wege der Datenübertragung miteinander verbunden. Eine Datenanfrage kann damit als gerichteter Graph bestehend aus Knoten und Kanten aufgefasst werden, wie er in Abbildung 5.2 dargestellt ist.

Die Sensoren dienen als Datenquellen und stellen besondere Knoten ohne Vorgänger dar. Die Anwendungen und/oder Zieldatenbanken fungieren als Datensenken ohne Nachfolgerknoten. Ein Operatorknoten hat immer einen Datenausgang und kann mehrere

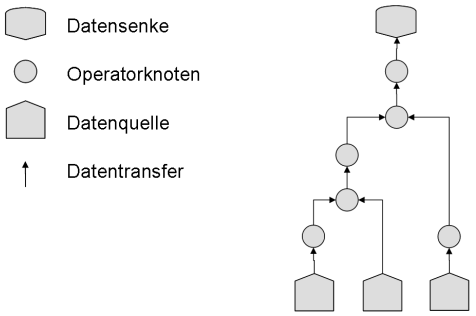


Abbildung 5.2.: Modellgraph der Datenverarbeitung

Dateneingänge besitzen. Jeder Datenausgang ist mit einem Eingangspunkt des nachfolgenden Operators über eine Kante zur Datenübertragung verbunden, die der logischen Modellierung des Datentransfers dient. Die Datenübertragung wird als unverzögert und verlustfrei angenommen. Die vorliegende Arbeit konzentriert sich auf Qualitätseinflüsse der Datenverarbeitung an den Operatorknoten. Datenübertragungsfehler oder Paketverluste können jedoch als zusätzliche Graphenknoten modelliert werden.

5.2. Datenqualität in der numerischen Algebra

In diesem Abschnitt wird untersucht, wie sich Messfehler und andere Datenqualitätsprobleme auf die Verarbeitungsergebnisse der numerischer Mathematik auswirken. Zuerst werden arithmetische Operatoren, wie Addition oder Quadratwurzel analysiert. Danach wird der Schwellwertvergleich untersucht, der die Datensätze eines Sensordatenstroms auf die Booleschen Ergebniswerte *wahr* und $falsch$ abbildet. Anschließend wird die Datenqualitätsverarbeitung bei Booleschen Operatoren erläutert.

5.2.1. Arithmetische Operatoren

Arithmetische Operatoren, wie zum Beispiel Addition und Division, fassen Datenstromattribute zu einem Ergebnisattribut zusammen. Unäre Operatoren der Arithmetik verändern die Datensätze eines einzelnen Datenstromattributes. Die Sensorwerte können zum Beispiel mit einer Konstanten gewichtet oder quadriert werden. Es wird vorausgesetzt, dass alle benötigten Datenstromattribute die gleiche Datenrate und keine Nullwerte besitzen. Ersteres wird durch den Verbund der Datenströme gewährleistet (siehe Abschnitt 5.4.4). Zweiteres wird durch die initiale Interpolation aller fehlenden Messwerte garantiert.

Werden mehrere Datenstromattribute miteinander verknüpft, müssen auch die beschreibenden DQ-Dimensionen verrechnet werden. Wie in Abschnitt 4.1.2 beschrieben, wird

die Datenqualität in Datenstromfenstern propagiert. Für jedes Attribut sind unterschiedliche Fenstergrößen erlaubt, so dass die Fenster der eingehenden Attribute nicht zwingend kongruent sind. Abbildung 5.3 zeigt zwei Attributströme mit den Fenstergrößen $\omega_1 = 5$ und $\omega_2 = 4$. Um die ressourcensparende Wirkung der DQ-Fenster zu erhalten, wird die größere Spanne als Fenstergröße des Ergebnisattributes gewählt: $\omega' = max(\omega_1, \omega_2) = 5$.

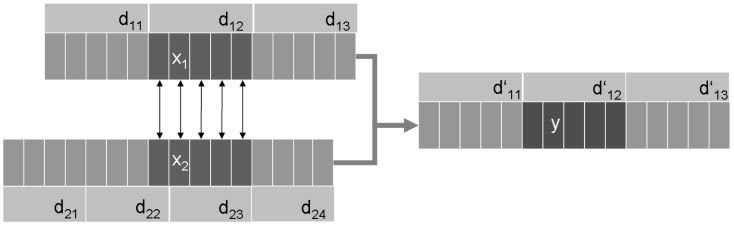


Abbildung 5.3.: Zusammenfassung des Datenvolumens

Datenmenge

Die Datenmenge gibt die Anzahl der Rohdatenwerte an, die zur Berechnung eines Ergebniswertes herangezogen wurden. Die Datenmenge eines Ergebnisfensters fasst die Datenmengen der eingehenden Fenster zusammen. Zur Berechnung der neuen Fensterqualität muss somit zuerst die Datenmenge eines einzelnen Datensatzes berechnet werden, um nachfolgend den Durchschnitt zu bilden. Die Datensätze erben die Eingangsqualitäten $q_w(k)$ des jeweiligen Datenqualitätsfenster $w(k)$, wobei $1 \leq k \leq \kappa$.

Die Datenmenge $d(y)$ in Abbildung 5.3 wird als Summe der Datenmengen $d(x_1)$ und $d(x_2)$: $d(y) = d_{12} + d_{23}$ bestimmt. Die durchschnittliche Fensterdatenmenge d'_{12} ergibt sich zu $d'_{12} = \frac{1}{5}(5 \cdot d_{12} + 1 \cdot d_{22} + 4 \cdot d_{23})$. Die Datenmengen d_{12} bzw. d_{23} beschreiben fünf bzw. vier Datentupel im Eingangsstrom des Ergebnisfensters d'_{12}, so dass sie mit 5 bzw. 4 gewichtet werden müssen.

Theorem 5.1 Die Ergebnisdatenmenge d_w beliebiger arithmetischer Operatoren $f(x_i|1 \leq i \leq n)$ ist stets die durchschnittliche Summe der Eingangsdatenmengen $d_w(i,p)$ gewichtet mit dem Anteil $part(i,p)$ der Eingangsdatenstromfenster.

$$d'_w(k) \quad = \quad \frac{1}{\omega_k} \sum_{i=1}^{n} \sum_{p=1}^{\#parts} part(i,p) \cdot d_w(i,p) \tag{5.1}$$

Die Summe wird über $\#parts$ beteiligte Fensterpartitionen und alle involvierten Attribute A_i für $1 \leq i \leq n$ gebildet. Das Beispiel in Abbildung 5.3 zeigt $\#parts = 2$ Partitionen der Fenster d_{22} und d_{23} des zweiten Attributstroms. Die Fensterdatenmenge beschreibt die

durchschnittliche Datenmenge der enthaltenen Datentupel, so dass anschließend durch die Fenstergröße $\omega_k = 5$ geteilt wird.

Dies gilt auch für unäre Operatoren. Hier ist $n = \#parts = 1$ und $part(1,1) = \omega_k$, so dass gilt $d'_w(k) = d_w(k)$. Die Datenmenge ist konstant.

Vollständigkeit

Numerische Operatoren entsprechen der konjunktiven Operatorklasse aus [BNQ06] (siehe Abschnitt 3.3.2). Ein Ergebniswert kann nur berechnet werden, wenn keines der eingehenden Datentupel Null-Werte aufweist. Zur Vermeidung kaskadierender Null-Werte in der Datenverarbeitung werden alle fehlenden Messwerte initial interpoliert. Die Fenstervollständigkeit drückt die Wahrscheinlichkeit aus, dass jedes Datentupel dieses Fensters interpoliert wurde. Um diese Interpretation der Vollständigkeit zu berücksichtigen, wird nicht die maximale, sondern die durchschnittliche Vollständigkeit als Fenstervollständigkeit bestimmt. Die Berechnung der Fenstervollständigkeit ähnelt der Berechnung der Datenmenge. Allerdings muss zusätzlich durch den Faktor n der Attributanzahl dividiert werden.

Theorem 5.2 Die Fenstervollständigkeit c_w für beliebige arithmetische Operatoren $f(x_i|1 \leq i \leq n)$ wird als durchschnittlicher Mittelwert der Eingangsvollständigkeiten $c_w(i,p)$, gewichtet mit dem Anteil $part(i,p)$ der Eingangsdatenstromfenster, berechnet.

$$c'_w(k) \quad = \quad \frac{1}{n \cdot \omega_k} \sum_{i=1}^{n} \sum_{p=1}^{\#parts} part(i,p) \cdot c_w(i,p) \tag{5.2}$$

Auch hier kann gezeigt werden, dass die Vollständigkeit für unäre Operatoren konstant ist. Wird $n = \#parts = 1$ und $part(1,1) = \omega_k$ gesetzt, so gilt $c'_w(k) = c_w(k)$.

Genauigkeit und Konfidenz

Ergebnisgenauigkeit und -konfidenz werden bei arithmetischen Operationen mit Hilfe der Gauß'schen Fehlerfortpflanzung berechnet. Die Messfehler der Eingangsdatenströme $A_i(1 \leq i \leq n)$ werden mit der jeweiligen partiellen Ableitung $\frac{\partial Y}{\partial A_i}$ des Berechnungsergebnisses Y gewichtet. Systematische Fehler der Genauigkeit werden linear, statistische Fehler der Konfidenz aufgrund ihrer zufälligen Streuung quadratisch addiert.

Zuerst werden die Genauigkeiten $a(y)$ und Konfidenzen $\epsilon(y)$ der einzelnen Ergebnisdatensätze berechnet. Die Eingangsgenauigkeiten $a(x_i)$ und -konfidenzen $\epsilon(x_i)$ der Datensätze x_i werden den entsprechenden fensterbasierten DQ-Informationen entnommen.

$$a(y) = \sum_{i=1}^{n} \left| \frac{\partial Y}{\partial A_i} \right| a(x_i) \tag{5.3}$$

$$\epsilon(y) = \sqrt{\sum_{i=1}^{n} \left(\frac{\partial Y}{\partial A_i} \right)^2 \epsilon(x_i)^2} \tag{5.4}$$

Die fensterbasierte Genauigkeit $a_w(k)$ und Konfidenz $\epsilon_w(k)$ wird nachfolgend als Durchschnitt der im Fenster enthaltenen Tupelgenauigkeiten bzw. -konfidenzen definiert.

Theorem 5.3 Die Genauigkeit a_w bzw. Konfidenz ϵ_w des Ergebnisfensters $w = [t_b; t_e]$ beliebiger arithmetischer Operatoren $f(x_i | 1 \leq i \leq n)$ wird als Durchschnitt der Datensatzgenauigkeiten bzw. -konfidenzen berechnet, die mit Hilfe der Gauß'schen Fortpflanzung systematischer bzw. statistischer Messfehler bestimmt wurden.

$$q_w(k) = \frac{1}{\omega_k} \sum_{j=t_b}^{t_e} q(y_j) \;\; q \in \{a, \epsilon\} \tag{5.5}$$

Sind die Fenster in den Datenstromattributen kongruent, vereinfacht sich die Berechnung. Die Fensterqualität kann direkt eingesetzt werden.

$$a'_w(k) = \sum_{i=1}^{n} \left| \frac{\partial Y}{\partial A_i} \right| a_w(k,i) \qquad \epsilon'_w(k) = \sqrt{\sum_{i=1}^{n} \left(\frac{\partial Y}{\partial A_i} \right)^2 \epsilon_w(k,i)^2} \tag{5.6}$$

Erweiterung auf korrelierte Messdaten

Die Standardform der Gauß'schen Fehlerfortpflanzung setzt unkorrelierte Messdatenströme voraus. In realistischen Anwendungen kann es jedoch zu Abhängigkeiten zwischen verschiedenen Sensormessreihen kommen. So können zum Beispiel hohe Temperaturen zu höheren Druckmessungen führen. Eine hohe Partikelverschmutzung von Hydrauliköl hat eine höhere Viskosität zur Folge.

Die erweiterte Form der Gauß'schen Fehlerfortpflanzung bedient sich einer modifizierten Form der Kovarianzmatrix COV um den statistischen sowie systematischen Fehler des Verarbeitungsergebnisses korrelierter Attribute $A_i (1 \leq i \leq n)$ zu berechnen. Die Eingangsfehler werden in der Matrixdiagonale eingetragen. Die Ergebniskonfidenz erfordert die quadratischen Eingangskonfidenzen. Da sich der systematische Messfehler linear fortpflanzt, sind die Eingangsgenauigkeiten einfach anzutragen.

$$COV(A_1, A_2, ..., A_n) = \begin{pmatrix} \epsilon(A_1)^2 & cov(A_1, A_2) & \dots & cov(A_1, A_n) \\ cov(A_2, A_1) & \epsilon(A_2)^2 & \dots & cov(A_2, A_n) \\ \vdots & \ddots & & \vdots \\ cov(A_n, A_1) & \dots & & \epsilon(A_n)^2 \end{pmatrix} \quad (5.7)$$

$$cov(A_i, A_j) = E[(A_i - E[A_i]) \cdot (A_j - E[A_j])] \quad i \neq j \quad (5.8)$$

Die Berechnung des statistischen Fehlers $\epsilon_v(y)$ (siehe Gleichung 5.4) erweitert sich wie folgt, wobei ∇f den Vektor der partiellen Ableitungen $\frac{\partial Y}{\partial A_i}$ darstellt.

$$\epsilon(y)^2 = \nabla f^T \cdot COV(A_1, ..., A_n) \cdot \nabla f \quad (5.9)$$

Zur Berechnung des systematischen Fehlers $a_v(y)$ setzt sich dieser Vektor aus den Quadratwurzeln der Ableitungen zusammen. Am anschaulichsten lässt sich die Anwendung der erweiterten Gauß'schen Fehlerfortpflanzung an einem Beispiel erklären. Sei $Y = A \cdot B$, so ergeben sich die partiellen Ableitungen im Vektor ∇f bzw. im transponierten Vektor ∇f^T wie folgt.

$$\frac{\partial Y}{\partial A} = B \qquad \frac{\partial Y}{\partial B} = A \quad (5.10)$$

$$\nabla f_\epsilon = \begin{pmatrix} B \\ A \end{pmatrix} \qquad \nabla f_\epsilon^T = (B\ A) \quad (5.11)$$

Um die Konfidenz bzw. Genauigkeit des Ergebnisses Y zu berechnen, werden die Eingangskonfidenzen $\epsilon(A), \epsilon(B)$ bzw. -genauigkeiten $a(A), a(B)$ in die Kovarianzmatrix eingesetzt. Somit gilt

$$\epsilon(y)^2 = (B\ A) \cdot \begin{pmatrix} \epsilon(A)^2 & cov(A, B) \\ cov(B, A) & \epsilon(B)^2 \end{pmatrix} \cdot \begin{pmatrix} B \\ A \end{pmatrix} \quad (5.12)$$

$$= B^2 \cdot \epsilon(A)^2 + A^2 \cdot \epsilon(B)^2 + 2AB \cdot cov(A, B) \quad (5.13)$$

sowie

$$a(y) = (\sqrt{B}\ \sqrt{A}) \cdot \begin{pmatrix} a(A) & cov(A, B) \\ cov(B, A) & a(B) \end{pmatrix} \cdot \begin{pmatrix} \sqrt{B} \\ \sqrt{A} \end{pmatrix} \quad (5.14)$$

$$= B \cdot a(A) + A \cdot a(B) + 2\sqrt{AB} \cdot cov(A, B)\ . \quad (5.15)$$

Diese Formeln lassen sich auf die einfache Gauß'sche Fehlerfortpflanzung zurückführen, indem unkorrelierte Ströme mit $cov(A, B) = 0$ angenommen werden. In diesem Fall ergeben Gleichung 5.3 und 5.4 die gleichen Funktionen zur Berechnung von Tupelgenauigkeit und -konfidenz.

5.2.2. Schwellwertvergleich

In vielen Anwendungen werden Sensordatenströme mit Schwellwerten verglichen. Der Schwellwert kann ein fest vorgegebener Zahlenwert sein oder mit Hilfe eines anderen Datenstroms berechnet werden. Deshalb müssen beim Schwellwertvergleich sowohl die Datenqualität des zu prüfenden Datenstroms als auch des Schwellwertes berücksichtigt werden.

Der Schwellwertvergleich wertet die binären Relationen $<, \leq, =, \geq$ oder $>$ aus und bildet einen Strom reelwertiger Datensätze $x \in \mathbb{R}$ auf die Booleschen Werte *wahr* oder *falsch*, bzw. 0 oder 1 ab: $\mathbb{R} \rightarrow \{0;1\}$.

Erfolgt der Vergleich mit einer Konstanten, wird weder die Datenmenge noch die Vollständigkeit bei dieser Abbildung geändert. Die Datenqualitätsfenster des Datenstroms bleiben konstant. Beim Vergleich mit einem anderen Sensordatenstrom müssen die Datenmengen und Vollständigkeiten zusammengefasst werden, wobei die größere Fensterspanne die neuen Datenqualitätsfenster vorgibt. Der Schwellwertvergleich verhält sich wie ein beliebiger arithmetischer Operator. Die Theoreme 5.1 und 5.2 können ohne Änderungen angewendet werden.

Die systematischen und statistischen Fehler der kontrollierten Datenstromwerte x und des Schwellwertes b, festgehalten in Genauigkeit $a_w(x)$ bzw. $a_w(b)$ und Konfidenz $\epsilon_w(x)$ bzw. $\epsilon_w(b)$, bilden den unsicheren Bereich $\delta = a_w(x) + a_w(b) + \epsilon_w(x) + \epsilon_w(x)$ um die Schwellwertfunktion (siehe Abbildung 5.4).

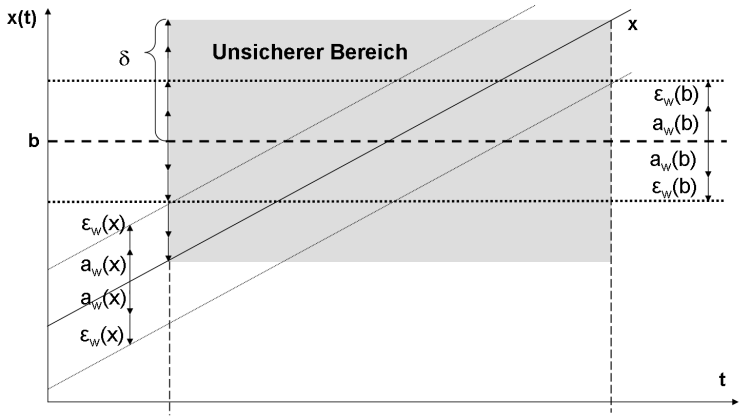


Abbildung 5.4.: Unsicherer Bereich beim Schwellwertvergleich

Im unsicheren Bereich ist es nicht möglich, klar zu entscheiden, ob der Schwellwert über- bzw. unterschritten wird. Um unsichere Entscheidungen im Ergebnis des Schwellwertvergleichs abzubilden, wird die Operatorfunktion erweitert. Entscheidungen im unsicheren Bereich $[b - \delta; b + \delta]$ wird entsprechend dem Grad der Unsicherheit ein re-

eller Wert zugeordnet, der als statistisch zufälliger Fehler in der DQ-Dimension der Konfidenz festgehalten wird.

$$\mathbb{R} \quad \rightarrow \quad \{0; \mathbb{R}; 1\} \tag{5.16}$$

Alle Entscheidungen außerhalb des unsicheren Bereiches können fehlerfrei getroffen werden. Es liegen weder systematische noch statistische Fehler vor, so dass gilt $a_w(k) = 0$ und $\epsilon_w(k) = 0$ für $1 \leq k \leq \kappa$.

Lägen gleich verteilte Messfehler vor, würde die Konfidenz eines Datensatzes im unsicheren Bereich auf 0,5 gesetzt werden (siehe Abbildung 5.4 a). Wie in Abschnitt 2.2 erläutert, werden numerische Messfehler jedoch als normal verteilt angenommen. Deshalb wird zur Abschätzung der Ergebniskonfidenz eine Gauß'sche Glockenkurve in den unsicheren Bereich gelegt (Abbildung 5.4 b). Die Konfidenz jedes einzelnen Datensatzes wird in Abhängigkeit von der Distanz $|x - b|$ zwischen Sensormesswert und Schwellwert berechnet. Anschließend wird die Fensterkonfidenz mit Hilfe der Durchschnittsbildung zusammengefasst.

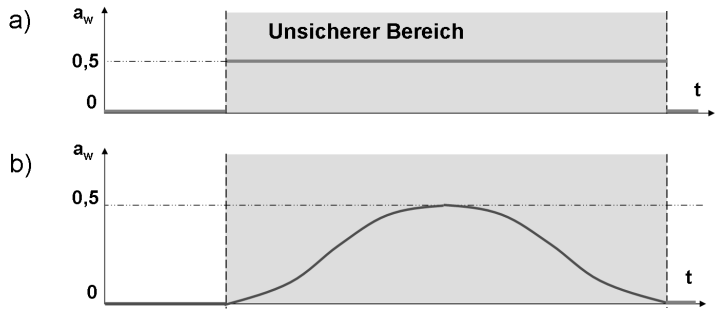


Abbildung 5.5.: Konfidenz des Schwellwertvergleichs

Theorem 5.4 Die Konfidenz $\epsilon_w(k)$ eines Schwellwertvergleichs wird durch die durchschnittliche Tupelkonfidenz $\epsilon(j)$ bestimmt, die mit Hilfe der Normalverteilung ϕ der Fehler im unsicheren Bereich $[b - \delta; b + \delta]$ berechnet wird. Die Verteilungsparameter μ und σ geben den Mittelwert und die Standardabweichung der beobachteten Distanz $dist = |x - b|$ innerhalb des untersuchten Datenqualitätsfensters an.

$$\epsilon_w(k) \quad = \quad \frac{1}{\omega_k} \sum_{j=t_b}^{t_e} \epsilon(j) \tag{5.17}$$

$$\epsilon(j) \quad = \quad \begin{cases} \Phi(|x(j) - b(j)|) & x(j) \in [b(j) - \delta(j); b(j) + \delta(j)] \\ 0 & else \end{cases} \tag{5.18}$$

Da die Unsicherheit vollständig in der DQ-Dimension Konfidenz festgehalten wird, wird die Genauigkeit des Schwellwertergebnisses auf $a_w(k) = 0$ gesetzt.

5.2.3. Boolesche Operatoren

Zur Verknüpfung verschiedener Sensordatenströme mit Booleschen Werten müssen Boolesche Operatoren hinsichtlich ihrer Qualitätsauswirkungen untersucht werden. Außerdem muss der neue Zustand der Unsicherheit einer Entscheidung aufgrund unpräziser Messdaten, z.B. $x = 0,5$, in die Boolesche Algebra integriert werden. Dazu werden die Operatoren der Aussagenlogik auf numerische Operatoren abgebildet. Die Aussage *wahr* erhält den Wert 0, *falsch* wird zu 1. Damit geben die numerischen Werte die Wahrscheinlichkeit des Zustandes $x = falsch$ an.

Die Negation eines Booleschen Wertes berechnet die Wahrscheinlichkeit, dass der Zustand $x = wahr$ angenommen wird. Da gilt $p(\neg x) = 1 - p(x)$, wird die Negation durch die Subtraktion von 1 abgebildet. Die Wahrheitswerte sind in Tabelle 5.1a aufgelistet.

a)

x_1	$\neg x_1$
1	0
0	1
0,5	0,5

b)

x_1	x_2	$x_1 \vee x_2$
1	1	1
1	0	0
1	0,5	0,5
0	1	0
0	0	0
0	0,5	0
0,5	1	0,5
0,5	0	0
0,5	0,5	0,25

Tabelle 5.1.: Wahrheitstabellen der Booleschen Negation und Disjunktion

Die Disjunktion wird durch die Multiplikation der Eingangswerte definiert. Tabelle 5.1b zeigt die möglichen Verknüpfungen. Ist der erste Attributwert unsicher, aber der zweite als wahr gegeben ($x_2 = 0$), so ist das Ergebnis wahr, da $wahr \vee x$ immer wahr ist. Entsprechend gilt $0 \cdot x = 0$. Ist das zweite Attribut falsch ($x_2 = 1$), so kann keine genaue Aussage gemacht werden. Das Ergebnis bleibt unsicher, da $falsch \vee wahr$ den Wert *wahr* zurückliefert, aber $falsch \vee falsch$ zu $falsch$ resultiert. Sind beide Attribute unsicher, so wird ein *falsches* Ergebnis bei der Disjunktion mit der Wahrscheinlichkeit $x_1 \vee x_2 = 0,25$ zurückgegeben.

Alle weiteren Operatoren lassen sich mit Hilfe der Kombination der Negation und der Disjunktion darstellen. Zum Beispiel ergibt sich für die Konjunktion die folgende numerische Abbildung.

$$x_1 \wedge x_2 = \neg(\neg x_1 \vee \neg x_2) \tag{5.19}$$
$$x_1 \wedge x_2 \Rightarrow 1 - (1 - x_1) \cdot (1 - x_2) \tag{5.20}$$

Nachdem die Boolesche Algebra auf numerische Operatoren abgebildet wurde, können die Theoreme 5.1 bis 5.4 zur Datenqualitätsverarbeitung aller Operatoren der Aussagenlogik herangezogen werden.

5.3. Datenqualität in der Signalverarbeitung

In diesem Abschnitt werden Operatoren der analogen Signalverarbeitung untersucht. Sie können ebenfalls auf digitalisierte Messdaten von Smart-Item-Sensoren angewendet werden. Die wichtigste Rolle spielen das Sampling, mit dessen Hilfe eine Stichprobe der Messdaten erhoben wird, um das Datenvolumen zu reduzieren, sowie die Interpolation, die Lücken im Datenstrom auffüllt.

5.3.1. Sampling

Mit Hilfe des Samplings wird eine definierte Anzahl zufällig gewählter Tupel aus dem Datenstrom entfernt. Samplingmethoden werden hauptsächlich zur Verringerung des Gesamtdatenvolumens zum Ausgleich hoher Lastspitzen in Datenstromsystemen, dem sogenannten Load-Shedding, genutzt. Außerdem können mit Hilfe von Stichprobenverfahren die Datenraten unterschiedlicher Datenströme für den Verbund angeglichen werden.

Die folgenden Überlegungen konzentrieren sich auf gleichförmige Stichprobenverfahren. Entsprechend dem Nyquist-Shannon-Abtasttheorem [KJ05] muss die Samplingfrequenz mehr als doppelt so groß wie die maximale Signalfrequenz sein, um Aliasing-Effekten vorzubeugen, die eine korrekte Rekonstruktion des Signals nach dem Sampling verhindern. Um trotzdem niedrige Samplingraten zu erlauben, kann das Sensorsignal durch einen Frequenzfilter geglättet werden (siehe Abschnitt 5.3.4).

Datenmenge, Vollständigkeit und Genauigkeit

Die Berechnung der Ergebnisqualität des Samplings erfolgt separat für jedes Datenstromfenster des Ergebnisdatenstroms. Abbildung 5.6 zeigt beispielhaft den Aufbau eines Ergebnisfensters bei einer Samplingrate von $r_{sa} = 1/4$ und der Fenstergröße $\omega = 5$. Die Grundgesamtheit der Stichprobe wird durch $1/r_{sa} \cdot \omega$ Tupel $x_{input} = x \in w'$ gebildet, im Beispiel sind dies 20 Tupel.

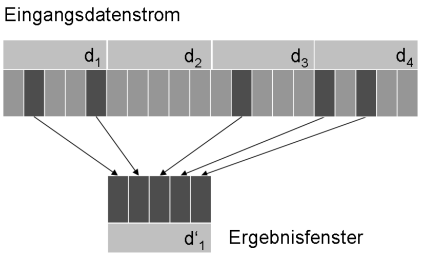


Abbildung 5.6.: Aufbau eines Ergebnisfensters beim Sampling

Um Datenmenge, Vollständigkeit und Genauigkeit des Ergebnisdatenstromfensters zu berechnen, werden die Qualitätsinformationen der in die Stichprobe aufgenommenen Tupel gemittelt.

Theorem 5.5 Die Datenmenge, Vollständigkeit und Genauigkeit eines Datenstromfensters wird beim Sampling als Durchschnitt der in die Stichprobe eingehenden Tupeldatenmengen, -vollstänigkeiten bzw. -genauigkeiten berechnet.

$$q_w(k) \quad = \quad \frac{1}{\omega_k} \sum q(x_{input}) \;\; q \in \{a, c, d\} \tag{5.21}$$

Daraus folgt für das obige Beispiel die Datenmenge $d'_1 = 1/5(2d_1 + d_3 + 2d_4)$, die Vollständigkeit $c'_1 = 1/5(2c_1 + c_3 + 2c_4)$ und die Genauigkeit $a'_1 = 1/5(2a_1 + a_3 + 2a_4)$.

Konfidenz

Der Informationsverlust aufgrund des Löschens von Datentupeln führt zu einem numerischen Fehler. Zum Beispiel besitzt die Tupelmenge $\{1, 1, 1, 1, 1, 1, 1, 1, 1, 100\}$ einen Durchschnitt von $avg = 10,9$. Wird das Tupel 100 in eine Stichprobe mit der Samplingrate $r_{sa} = 0,5$ aufgenommen, so wird der Durchschnitt nachfolgend zu $avg = 20,3$ berechnet. Ist dieses Tupel nicht in der Stichprobe enthalten, ergibt sich der verfälschte Durchschnitt zu $avg = 1$. Beide Ergebnisse weichen stark vom realen Ergebniswert ab. Durch den Zufallscharakter des Samplings wird ein statistischer Fehler eingeführt, der um den wahren Wert schwankt. Er muss für die nachfolgende Datenverarbeitung in der DQ-Dimension der Konfidenz festgehalten werden.

In [Haa97] leitet Haas die Berechnung des statistischen Samplingfehlers von der Definition des Konfidenzintervalls einer normal verteilten Datenreihe ab. Die Ausdehnung des Konfidenzintervalls kon wird in Gleichung 5.22 durch die Varianz σ^2 der Datenreihe und die Menge der Daten n bestimmt. Die Konfidenzwahrscheinlichkeit p beschreibt die Wahrscheinlichkeit, mit der der reale Wert im Konfidenzintervall $[\bar{x} - kon; \bar{x} + kof]$ um den Mittelwert der Datenreihe $\bar{x}$ liegt.

$$kon = z(1-\frac{p}{2})\frac{\sigma}{\sqrt{n}} \tag{5.22}$$

Der Ausdruck $z(1-\frac{p}{2})$ definiert das $1-p/2$-Quantil der zugrunde liegenden Verteilungsfunktion. Unter der Annahme der Normalverteilung kann $z(1-\frac{p}{2})$ durch die inverse kumulierte Dichteverteilungsfunktion der Normalverteilung $\phi = \Phi^{-1}(1-\frac{p}{2})$ angenähert werden. Die Quantile typischer Konfidenzwahrscheinlichkeiten sind in Tabelle 2.1 auf Seite 14 aufgelistet. Um den Konfidenzintervall einer Stichprobe kon_{sample} zu berechnen, wird die Stichprobenvarianz $s^2 = \frac{1}{1-n}\sum(x_i - \bar{x})^2$ als Abschätzung der Varianz σ^2 eingesetzt. Der Parameter n beschreibt nun die Größe der Grunddatenreihe, aus der die Stichprobe gezogen wird. Außerdem wird das Konfidenzintervall mit der Wurzel der Rate der gelöschten Tupel $1 - r_{sa}$ gewichtet (dritter Term in Gleichung 5.23).

$$kon_{sample} = \phi \cdot \frac{s}{\sqrt{n}} \cdot \sqrt{1 - r_{sa}} \tag{5.23}$$

Um den eingefügten Samplingfehler ϵ_w^+ zu berechnen, wird Gleichung 5.23 angepasst. Die Berechnung des Konfidenzintervalls erfolgt separat für jedes Datenstromfenster des Ergebnisdatenstroms. Die Größe der Stichprobe entspricht der DQ-Fenstergröße ω (siehe Abbildung 5.6). Die Größe der Grunddatenreihe ist somit $n = 1/r_{sa} \cdot \omega$. Die Stichprobenvarianz s^2 wird anhand der Datentupel des erzeugten Ergebnisfensters w' berechnet.

$$\epsilon_{w'}^+ = \phi \cdot \frac{s(w')}{\sqrt{\omega \cdot \frac{1}{r_{sa}}}} \cdot \sqrt{1 - r_{sa}} \tag{5.24}$$

Der geschätzte statistische Fehler ist umso größer, je stärker die Messdaten im Datenstromfenster streuen und je höher die Konfidenzwahrscheinlichkeit gewählt wird. Entsprechend dem hohen Informationsverlust ist der Fehler größer, je kleiner die Samplingrate und damit die gezogene Stichprobe ist. Entspricht die oben gegebene Beispieltupelfolge einem Datenqualitätsfenster, so beträgt das Konfidenzintervall $e_w^+ = 19,4$ bei einer Konfidenzwahrscheinlichkeit von $p = 95\%$, d.h. $\phi = 2,58$.

Entsprechend der Gauß'schen Fehlerfortpflanzung setzt sich die Gesamtkonfidenz des Ergebnisfensters aus den Eingangskonfidenzen der Datensätze der erzeugten Stichprobe sowie dem eingefügten statistischen Samplingfehler zusammen.

Theorem 5.6 Die Konfidenz eines Ergebnisfensters w' des Samplingoperators setzt sich aus dem quadratischen Durchschnitt der Konfidenzen ϵ_w der in die Stichprobe eingehenden Tupel x_{input} und des neu hinzugefügten Samplingfehlers $\epsilon_{w'}^+$ zusammen.

$$\epsilon_w(k) = \sqrt{\frac{1}{\omega_k}\sum \epsilon_w(x_{input})^2 + {\epsilon_{w'}^+}^2} \tag{5.25}$$

5.3.2. Interpolation

Die Interpolation generiert Datensätze basierend auf bestehenden Sensormessdaten, um Lücken durch Sensorausfälle zu füllen oder die Datenrate zweier Datenströme für den Verbund anzupassen. Durch die Datengenerierung mit der Interpolationsrate r_{in} werden $(r_{in} - 1) \cdot \omega$ Tupel im Datenqualitätsfenster eingefügt. Um die Fenstergröße und damit die Granularität der DQ-Informationen konstant zu halten, müssen neue Fenster gebildet werden. Jedes Mutterfenster wird in r_{in} Kindfenster aufgeteilt. Abbildung 5.7 zeigt die Interpolation der Rate $r_{in} = 2$.

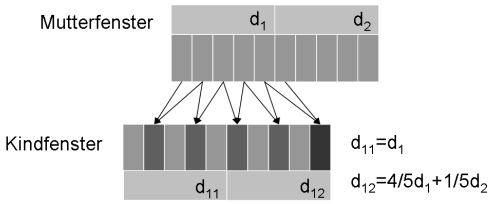


Abbildung 5.7.: Interpolation von Datenstromtupeln

Datenmenge, Genauigkeit und Konfidenz

Während die Datensätze mit Hilfe existierender Messwerte generiert werden, müssen die DQ-Informationen auf Basis vorhandener Datenqualitäten mit der gleichen Interpolationsstrategie eingefügt werden. Dabei erben die ersten $r_{in} - 1$ Kindfenster die Fensterdatenmenge, -genauigkeit und -konfidenz vom Mutterfenster, das beide Interpolationspartner enthält (siehe Abbildung 5.8), so dass gilt: $q_w(erste\ Kinder) = q_w(Mutter)$.

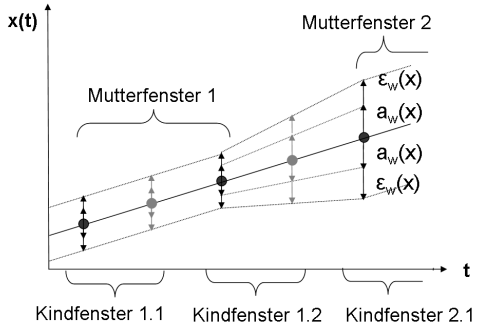


Abbildung 5.8.: Aufteilung der Kindfenster

Der letzte Daten- und Qualitätswert des letzten Kindfensters wird zwischen zwei DQ-Mutterfenstern k und $k+1$ interpoliert (siehe Abbildung 5.7). In Abbildung 5.8 wird das zweite Tupel des Kindfensters 1.2 zwischen den Mutterfenstern 1 und 2 generiert. Die Fensterqualität des Kindfensters fasst in Gleichung 5.26 die tupelbasierten Datenqualitätswerte der ersten $\omega - 1$ Datentupel $(q = q_w(k))$ und den Datenqualitätswert des letzten Datentupels $(q = gen(q_w(k), q_w(k+1))$ zusammen. Die Funktion gen definiert die ausgeführte Interpolationsfunktion.

$$q_w(letztesKind) = \frac{1}{\omega_k}[\underbrace{(\omega_k - 1) \cdot (q_w(k))}_{erste\ Datentupel} + \underbrace{gen(q_w(k), q_w(k+1))}_{letztes\ Datentupel}] \quad (5.26)$$

Bei der linearen Interpolation mit $gen = (x_1 + x_2)/2$ und $r_{in} = 2$ berechnen sich Ergebnisqualitäten zum Beispiel wie folgt.

$$q_w(erste\ Kinder) = q_w(k) \quad (5.27)$$

$$q_w(letztes\ Kind) = \frac{\omega_k - 1}{\omega_k} \cdot q_w(k) + \frac{1}{\omega_k} \cdot q_w(k+1) \quad (5.28)$$

Im Beispiel in Abbildung 5.7 wird die Datenmenge des ersten entstandenen Kindfensters $d_w(erste\ Kinder) = d_{11}$ mit Hilfe der Datenmenge des Mutterfensters $d_w(k) = d_1$ bestimmt. Die Datenmenge des zweiten Kindes $d_w(letztes\ Kind) = d_{12}$ ergibt sich aus Gleichung 5.26 bzw. 5.28 zu $d_{12} = \frac{4}{5} \cdot d_1 + \frac{1}{5} \cdot d_2$.

Theorem 5.7 Fensterdatenmenge, -genauigkeit und -konfidenz werden bei der Interpolation als Durchschnitt der eingehenden Tupeldatenmengen, -genauigkeiten bzw. -konfidenzen berechnet. Bei den ersten $r_{in} - 1$ Kindfenster entsprechen sie den Fensterqualitätswerten des Mutterfensters k. Das letzte Kindfenster fasst die Tupelqualitäten von $\omega_k - 1$ Kindtupeln des Mutterfensters k und eines zwischen den Mutterfenstern k und $k+1$ interpolierten Datentupels zusammen.

Vollständigkeit

Durch Interpolation verringert sich die Vollständigkeit, die den Anteil originaler Messdaten im Datenstrom repräsentiert.

Im ersten Schritt muss die Vollständigkeit der Kindfenster von der Muttervollständigkeit entsprechend der Strategie für Datenmenge, Genauigkeit und Konfidenz abgeleitet werden. Diese wird dann mit dem Reziproken der Interpolationsrate $1/r_{in}$ gewichtet, um die hinzugefügten Datensätze abzubilden.

Theorem 5.8 Die Fenstervollständigkeit wird bei der Interpolation als Durchschnitt der eingehenden Tupelvollständigkeiten berechnet und nachfolgend durch die Interpolationsrate dividiert.

5.3.3. Frequenzanalyse

Das Frequenzspektrum beschreibt die Gesamtheit der Frequenzen eines analogen Signals. Die Frequenzanalyse wird zur Transformation eines Sensorsignals vom Zeit- in den Frequenzbereich genutzt, so dass die enthaltenen Frequenzbänder sowie deren Amplituden und Phasen sichtbar werden. Dazu können unter anderem die Fourier-Analyse, die Laplace-Transformation oder eine Wavelet-Transformation verwendet werden.

Abbildung 5.9 zeigt die Frequenzanalyse am Beispiel eines Signals bestehend aus zwei überlagerten Sinuswellen $x(t) = sin(1 \cdot 2\pi t) + 0,8 \cdot sin(2 \cdot 2\pi t) + 3$. Im transformierten Signal im Frequenzbereich sind deutlich die Amplitudenspitzen der Frequenzen $f_1 = 1Hz$ und $f_2 = 2Hz$ zu erkennen.

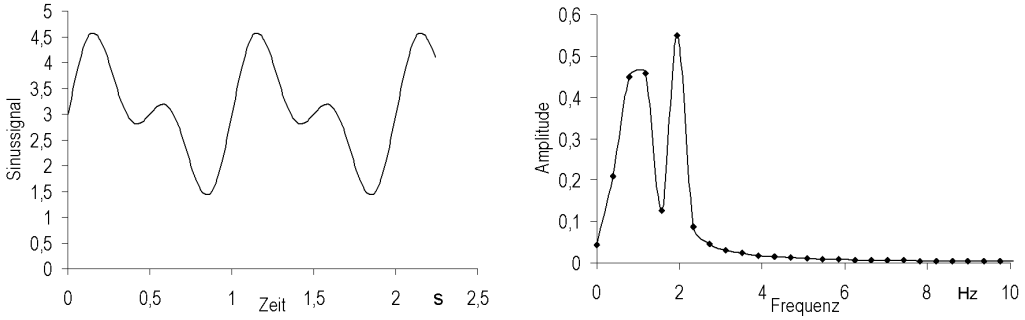


Abbildung 5.9.: Frequenzanalyse eines Sinussignals

Die Transformation der Datenqualitätsinformationen vom Zeit- in den Frequenzbereich wird am Beispiel der Fourier-Transformation [KJ05] beschrieben, wobei die abgeleiteten Methoden der DQ-Verarbeitung auch für alle anderen Transformationsfunktionen gelten. Gleichung 5.29 beschreibt die Fourier-Transformation der n-ten Periode eines zeitdiskreten, periodischen Signals von der Zeit- in die Frequenzdomäne.

$$X(n) \quad = \quad \frac{1}{N} \sum_{t_k=0}^{N-1} x(t_k) e^{-2i\pi \frac{nt_k}{N}} \tag{5.29}$$

Bei der Fourier-Transformation eines Sensordatenstroms entspricht N der Länge l der untersuchten Datengruppe, die Zeitvariable t_k wird durch den Index $j(0 \leq j < l)$ über alle Datentupel dieser Gruppe ersetzt. Die schnelle Fourier-Transformation (engl. Fast Fourier Transformation, FFT) erlaubt die Reduzierung des numerischen Berechnungsaufwandes von $O(N(N-1))$ zu $O(NldN)$ für den Spezialfall $N = 2^p (p \in \mathbb{Z})$ und liefert damit einen Algorithmus, der für die schnelle Verarbeitung von Datenströmen geeignet ist.

$$X_{sensor}(n) = \frac{1}{l}\sum_{j=0}^{l-1} x(j)e^{-2i\pi\frac{nj}{l}} \tag{5.30}$$

Datenmenge und Vollständigkeit

Die Frequenzanalyse liefert eine neue Repräsentation der Daten, deren Umfang die Granularität der resultierenden Frequenzen bestimmt. Typischerweise wird das Datenvolumen verkleinert, da das Frequenzspektrum eine kompaktere Repräsentation der Messdaten, eine sogenannte Synopse, darstellt. Zum Beispiel liefert die Fourier-Analyse von l Eingangsdatentupeln Aussagen über maximal $l/2+1$ Frequenzen. Die Synopsengröße wird η bezeichnet. Eine Synopse kann ein oder mehrere Datenqualitätsfenster umfassen $\eta \geq \omega$, aber auch Teil eines größeren Qualitätsfensters sein $\eta < \omega$. Um den Datenfluss der Datenstromverarbeitung nicht zu blockieren, wird die Frequenzanalyse ähnlich einer Aggregation in Datenstromfenstern ausgeführt (siehe Abschnitt 5.4.5).

Zur Berechnung der Ergebnisdatenqualität wird zuerst die Qualität eines einzelnen Datentupels der Ergebnissynopse untersucht. Entsprechend dem Verhältnis der Synopsen- und DQ-Fenstergröße, werden anschließend neue Datenqualitätsfenster im Frequenzbereich aufgebaut.

Die Vollständigkeit der Ergebnistupel einer erzeugten Synopse c_{syn} fassen die Vollständigkeiten der Eingangstupel dieser Synopse zusammen. Da alle Eingangsmesswerte gleichwertig zur Synopsendefinition beitragen, wird die Vollständigkeit über der gesamten Menge der Eingangsdaten gemittelt.

$$c_{syn} = \frac{1}{l}\sum_{j=1}^{l} c_v(j) \tag{5.31}$$

Gilt $\eta < \omega$, so werden die Vollständigkeiten aller Synopsen, die im zu erstellenden Datenqualitätsfenster enthalten sind, ebenfalls gemittelt.

Theorem 5.9 Die Vollständigkeit $c'_w(k)$ eines Datenqualitätsfensters des Frequenzspektrums wird als Durchschnitt der eingehenden Vollständigkeitsinformationen berechnet.

$$c'_w(k) = \begin{cases} \frac{1}{\omega}\sum_{i=1}^{\omega/\eta} c_{syn}(i) & \eta < \omega \\ c_{syn} & else \end{cases} \tag{5.32}$$

Wie bei der Vollständigkeit trägt die Datenmenge jedes Eingangstupels gleichförmig zur Datenmenge d_{syn} der entstehenden Synopsentupel bei. Aufgrund des additiven Charakters dieser DQ-Dimension wird die gesamte Eingangsdatenmenge zusammengefasst und nachfolgend auf die Datenqualitätsfenster der Synopse aufgeteilt.

$$d_{syn} = \sum_{j=1}^{l} d_v(j) \tag{5.33}$$

Theorem 5.10 Die Datenmenge $d'_w(k)$ eines Datenqualitätsfensters des Frequenzspektrums wird als Summe der eingehenden Datenmengen berechnet.

$$d'_w(k) = \begin{cases} \frac{1}{\omega} \sum_{i=1}^{\omega/\eta} d_{syn}(i) & \eta < \omega \\ \frac{\omega}{\eta} d_{syn} & else \end{cases} \tag{5.34}$$

Genauigkeit und Konfidenz

Die DQ-Dimensionen Genauigkeit und Konfidenz beschreiben die absoluten Messfehler der Sensormesswerte. Das Signal eines Sensordatenstroms kann damit als Addition der wahren Messwerte $\hat{x}(t)$ und des systematischen und statistischen Messfehlers aufgefasst werden: $x(j) = \hat{x}(j) + a(t) + \epsilon(t)$. Gleichung 5.29 wird in die Transformation des wahren Sensorsignals (erster Term der Gleichung 5.36), des systematischen Messfehlers (zweiter Term) und des statistischen Messfehlers (dritter Term) aufgeteilt.

$$X(n) = \frac{1}{l} \sum_{j=0}^{l-1} \hat{x}(j) + a_w(j) + \epsilon_w(j) e^{-2i\pi \frac{nj}{l}} \tag{5.35}$$

$$X(n) = \underbrace{\frac{1}{l} \sum_{j=0}^{l-1} \hat{x}(j) e^{-2i\pi \frac{nj}{l}}}_{\hat{X}(n)} + \underbrace{\frac{1}{l} \sum_{j=0}^{l-1} a_w(j) e^{-2i\pi \frac{nj}{l}}}_{a_{syn}=0} + \underbrace{\frac{1}{l} \sum_{j=0}^{l-1} \epsilon_w(j) e^{-2i\pi \frac{nj}{l}}}_{\epsilon_{syn}=0} \tag{5.36}$$

Wird die Frequenzanalyse separat für jedes Datenqualitätsfenster ($l = \omega$) oder in noch kleineren Strompartitionen ($l < \omega$) ausgeführt, sind Genauigkeit und Konfidenz konstant. Die Fourier-Transformation einer Konstanten beträgt $X(n) = 0$. Die konstanten Messfehler haben keine Auswirkung auf das Frequenzspektrum des Signals, so dass gilt $\epsilon_{syn} = 0$ und $a_{syn} = 0$ (siehe Gleichung 5.36). Systematischer und statistischer Messfehler des Frequenzspektrums betragen $a_w = \epsilon_w = 0$.

Eine weitere Möglichkeit ist die Analyse (gleitender) Datenstromfenster, die mehrere Datenqualitätsfenster überdecken ($l > \omega$). Hier kann der Signalanteil des statistischen und systematischen Messfehlers mit Hilfe je einer Rechteckimpulsfolge dargestellt werden. Jeder Rechteckimpuls beschreibt ω Datenstromtupel. Die Höhe der Impulse wird durch den Wert des Messfehlers ϵ_w bzw. a_w bestimmt. Gleichung 5.37 zeigt die Fourier-Transformation eines Rechteckimpulses.

$$X_q(n) = \frac{q_w}{l} \cdot \frac{sin\omega\pi\frac{n}{l}}{sin\pi\frac{n}{l}} \quad \forall q \in \{a, \epsilon\} \tag{5.37}$$

Erfolgt die Frequenzanalyse über mehrere Datenqualitätsfenster, so müssen neben der Transformation des Messdatenstroms $2 \cdot l/\omega$ Rechteckimpulstransformationen ausgeführt werden, um die Auswirkungen der statistischen und systematischen Messfehler der Zeitdomäne auf das Frequenzspektrum zu bestimmen. Systematischer und statistischer Messfehler a_{syn} bzw. ϵ_{syn} der erzeugten Synopse werden mit Hilfe von Gleichung 5.37 als Summe der Messfehler der eingehenden Datenqualitätsfenster w_k mit $1 \leq k \leq l/\omega$ wie folgt berechnet.

$$a_{syn} = \sum_{k=1}^{l/\omega} X_{a_w(k)}(n) \qquad \epsilon_{syn} = \sum_{k=1}^{l/\omega} X_{\epsilon_w(k)}(n) \tag{5.38}$$

Wie bereits für die DQ-Dimensionen Vollständigkeit und Datenmenge beschrieben, werden auch im Frequenzbereich Datenqualitätsfenster gebildet, um das Datenvolumen der Qualitätsinformationen zu minimieren. Nachdem die Frequenzfehler mit Hilfe der Fourier-Transformation der Rechteckimpulsfolge bestimmt wurden, müssen sie in Fenstern zusammengefasst werden. Da die Messfehler der Zeitdomäne gleichförmig auf alle Frequenzbänder wirken, muss über alle Synopsentupel gemittelt werden.

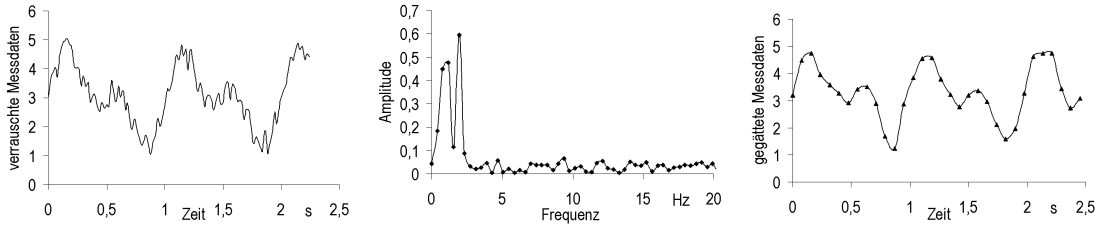
Theorem 5.11 Genauigkeit $a'_w(k)$ bzw. Konfidenz $\epsilon'_w(k)$ eines Datenqualitätsfensters des Frequenzspektrums wird als Durchschnitt der transformierten Genauigkeiten a_{syn} bzw. Konfidenzen ϵ_{syn} berechnet.

$$q'_w(k) \quad = \quad \begin{cases} \frac{1}{\omega} \sum_{i=1}^{\omega/\eta} q_{syn}(i) & \eta < \omega \\ q_{syn} & else \end{cases} \qquad q \in \{a, \epsilon\} \tag{5.39}$$

5.3.4. Frequenzfilter

Frequenzfilter dienen der Selektion bestimmter Frequenzanteile eines analogen Signals und sind damit wichtige Werkzeuge der Signalverarbeitung. Hochpassfilter löschen niedrige Frequenzen aus dem Messdatenstrom. Der Tiefpassfilter glättet das Eingangssignal, indem er störende Anteile hoher Frequenzen entfernt, um Aliasing-Effekten beim Sampling vorzubeugen. Weitere Filter arbeiten auf Basis der Signalphase oder -impedanz.

Abbildung 5.10 a zeigt das Sinussignal aus Abbildung 5.9, das durch Störungen z.B. bei der Datenübertragung verrauscht wurde. Das Frequenzspektrum in Abbildung 5.10 b weist nun neben den Hauptfrequenzen $f_1 = 1Hz$, $f_2 = 2Hz$ sehr viele hohe Frequenzanteile mit niedriger Amplitude auf. Zur Glättung des verrauschten Signals wird ein Tiefpassfilter mit der Filtergrenzfrequenz $f = 5Hz$ angewendet. Abbildung 5.10 c zeigt das rekonstruierte Signal im Zeitbereich. Die höchste Signalfrequenz wurde auf die Filtergrenzfrequenz herabgesetzt.



a) verrauschtes Sinussignal b) Frequenzspektrum c) gefiltertes Signal

Abbildung 5.10.: Tiefpassfilter zur Signalglättung

Ein Frequenzfilter kann als Komposition aus Frequenzanalyse und Selektion beschrieben werden. Zuerst wird das Sensorsignal mittels der Frequenzanalyse (z.B. Fourier-Transformation) vom Zeit- in den Frequenzbereich übertragen. Je nach Filterart werden dann alle Frequenzen des Spektrums aus dem Datenstrom gelöscht, die die Filtergrenzfrequenz, die Filtergrenzphase oder -impedanz über- bzw. unterschreiten. Schließlich wird die Rücktransformation in den Zeitbereich vorgenommen, um das gefilterte Signal auf Basis des verbleibenden Frequenzbandes zu rekonstruieren.

Zur Abschätzung der Datenqualität des gefilterten Datenstroms muss die Qualitätsverarbeitung der Basisoperatoren eingesetzt werden. Die Auswirkungen der Transformation vom Zeit- in den Frequenzbereich wurden im vorangegangenen Abschnitt 5.3.3 erläutert. Die Datenqualitätsverarbeitung der Selektion wird in Abschnitt 5.4.3 detailliert beschrieben. Die Rücktransformation erfolgt analog zur Hintransformation, so dass erneut die Theoreme der Frequenzanalyse angewendet werden können.

5.4. Relationale Datenqualität

In diesem Abschnitt wird die Datenqualitätsalgebra der Continuous Query Language (CQL) der Datenstromverarbeitung sowie der Structured Query Language (SQL) der Anfrageverarbeitung in Datenbanksystemen vorgestellt.

Bei der Datenverarbeitung in ressourcenschwachen Smart-Item-Systemen müssen die Datenqualitätsfenster durch den gesamten Verarbeitungspfad propagiert werden, um das Datenvolumen so gering wie möglich zu halten. Dazu werden zuerst die Qualitäten der einzelnen Ergebnisdatensätze berechnet, die nachfolgend zu neuen Datenqualitätsfenstern zusammengesetzt werden. In der Datenverarbeitung in Datenbanksystemen entfällt der zweite Schritt. Die Ergebnisdatenqualität wird tupelweise ausgegeben.

5.4.1. Mengenoperatoren

Die relationale Algebra umfasst die Mengenoperatoren Vereinigung, Schnittmenge und Differenz. Sie können mit Hilfe der Strom-zu-Relation-Operatoren der CQL ohne Ände-

rungen auf Datenstromfenster angewendet werden. Mengenoperatoren haben keinen Einfluss auf die Ergebnisdatenqualität. Allerdings wird die Datenstromrate durch die Vereinigung zweier Ströme vergrößert, so dass der Ergebnisstrom in neue Datenqualitätsfenster aufgeteilt werden muss. Bei der Schnittmenge und Differenz wird der Strom verkleinert, so dass die Ergebnistupel zu neuen Fenstern zusammengefasst werden. Die Datenqualität der Ergebnisfenster wird als Durchschnitt der eingehenden Fensterqualitäten bestimmt. Analog zu numerischen Operatoren wird bei unterschiedlicher Fenstergröße der Eingangsdatenströme die größere Fensterlänge für den Ergebnisstrom gewählt.

Theorem 5.12 Die Datenqualität eines Ergebnisfensters der Vereinigung, Schnittmenge und Differenz wird als Durchschnitt der eingehenden Fensterdatenqualitäten bestimmt, so dass für alle Dimensionen $q \in \{a, \epsilon, c, d\}$ gilt

$$q'_w(k) = \frac{1}{w} \sum_{j=t_b}^{t_e} q_w(k) \ . \tag{5.40}$$

5.4.2. Projektion

Die Projektion auf Attribute relationaler Datenbanktabellen beinhaltet eine implizite Duplikateliminierung. Die CQL hingegen trennt die einfache Projektion, d.h. Selektion auf Attributebene, von der Duplikaterkennung (siehe Abschnitt 2.3). Die Datenqualitätsverarbeitung wird auf Basis der separierten Basisoperatoren „Attributprojektion“ und „Duplikateliminierung“ analysiert.

Um Datenqualitätsinformationen im Datenstrom zu verarbeiten, muss die Attributprojektion auf alle relevanten Datenqualitätsinformationen ausgeweitet werden. Die Attributprojektion $o = \pi(A_i)$ wird um den Datenqualitätsoperator $o^{DQ} = \pi(a(A_i), \epsilon(A_i), c(A_i), d(A_i))$ ergänzt. Die DQ-Informationen selbst werden jedoch nicht beeinflusst. Die Fensterdatenqualitäten bleiben unverändert.

Theorem 5.13 Die Attributprojektion hat keinen Einfluss auf die im Datenstrom mitgeführten Datenqualitätsinformationen, so dass für alle Dimensionen $q \in \{a, \epsilon, c, d\}$ gilt $q'_w(k) = q_w(k)$.

Die Duplikaterkennung in Sensordatenströmen wird durch das Schlüsselattribut des Zeitstempels erleichtert (siehe Abschnitt 3.2.3). Werden aus Gründen der Sicherheit und Zuverlässigkeit redundante Sensormessungen vorgenommen, können sie mit Hilfe des gemeinsamen Zeitstempels zugeordnet werden. Dabei werden abweichende Sensormessungen aggregiert. Zum Beispiel wird der Durchschnitt der Partikelverschmutzungen gebildet, die an verschiedenen Stellen im Hydrauliksystem gemessen wurden, um

die Ölalterung in Anwendungsszenario 1 zu bestimmen. Die Duplikateliminierung entspricht im Falle der Sensordatenverarbeitung einer Aggregation, so dass die in Abschnitt 5.4.5 aufgestellten Theoreme zur Datenqualitätsverarbeitung herangezogen werden müssen.

5.4.3. Selektion

Die Selektion in relationalen Datenbanken extrahiert eine Auswahl an Datenobjekten aus einer Datenbanktabelle. Die Bedingung in der WHERE-Klausel des SQL-Befehls wird ausgewertet und nur Datenobjekte, die die geforderten Eigenschaften erfüllen, werden zur weiteren Verarbeitung herangezogen. Bei der Selektion im Kontext von Sensordatenströmen werden Datensätze aufgrund der Eigenschaften eines bzw. mehrerer Messwerte ausgewählt, die in der WHERE-Klausel des CQL-Befehls angegeben sind. Tupel, die den Selektionsbedingungen nicht entsprechen, werden aus dem Datenstrom entfernt.

Datenmenge, Vollständigkeit und Genauigkeit

Sind Datenqualitätsinformationen und Selektionsattribut unkorreliert, hat die Selektion keinen Einfluss auf die Ergebnisqualität. So werden Datensätze mit hoher Genauigkeit mit derselben Wahrscheinlichkeit selektiert wie Daten mit größeren Messfehlern. In realen Anwendungen ist dies jedoch nicht immer der Fall. Vollständigkeit und Genauigkeit können mit dem Messwert korrelieren. Extreme Umweltbedingungen, zum Beispiel hohe Temperaturen, können Sensorausfälle verursachen oder den systematischen Messfehler erhöhen. Werden dann Messwerte über 60°C in einer Selektion ausgewählt, wird die durchschnittliche Vollständigkeit und Genauigkeit des Ergebnisses herabgesetzt.

Die Datenmenge wird durch die vorgenommene Datenverarbeitung bestimmt und ist damit unabhängig von den Messwerten. Die Datenmenge des Ergebnisfensters der Selektion fasst die Datenmengen eingehender Datentupel zusammen (siehe Theorem 5.5).

Theorem 5.14 Die Datenmenge eines Datenstromfensters wird bei der Selektion als Durchschnitt der Datenmengen der in das Ergebnisfenster eingehenden Tupel $x_{input} = x \in w'$ berechnet.

$$d_w(k) \quad = \quad \frac{1}{\omega_k} \sum d_v(j \in x_{input}) \tag{5.41}$$

Die Korrelation zwischen DQ-Information Q und Selektionsattribut A wird mit Hilfe des Pearson-Korrelationskoeffizienten ς berechnet, der auf der in Abschnitt 5.2.1 auf Seite 87 eingeführten Kovarianz $cov(Q, A)$ beruht. Der Pearson-Korrelationskoeffizient wird in [HEK05] wie folgt definiert.

$$\varsigma(Q,A) = \frac{cov(Q,A)}{\sqrt{\mathrm{Var}(Q)} \cdot \sqrt{\mathrm{Var}(A)}} \tag{5.42}$$

$$\tag{5.43}$$

Liegt eine positive Korrelation vor ($\varsigma > 0$), entsprechen hohe Messwerte einer hohen Fehlerrate bzw. einem hohen systematischem Fehler. Bei einer negativen Korrelation ($\varsigma < 0$) enthalten hohe Messwerte einen kleineren Fehler und besitzen damit eine höhere Datenqualität. Um den Einfluss der Selektion auf die korrelierte Ergebnisqualität abzubilden, muss neben dem Korrelationskoeffizienten die Art der Selektion bestimmt werden. So werden im obigen Beispiel hohe Datenwerte selektiert, die positiv mit den DQ-Informationen korrelieren, so dass die Ergebnisdatenqualität verringert wird. Tabelle 5.2 fasst die möglichen Auswirkungen zusammen.

Korrelation \ **Selektion**	hohe Werte	niedrige Werte
positiv	DQ ↘	DQ ↗
negativ	DQ ↗	DQ ↘

Tabelle 5.2.: DQ-Einfluss der korrelierten Selektion

Die Verschiebung $\Delta\mu$ des Datenmittelwertes vor (apriori) und nach (aposteriori) der Selektion bestimmt die Art und den Grad der Selektion.

$$\Delta\mu = \frac{\mu_{aposteriori} - \mu_{apriori}}{max(\mu_{aposteriori}, \mu_{apriori})} \tag{5.44}$$

$$\mu = \frac{1}{\omega} \sum_{j=t_b}^{t_e} x_j \tag{5.45}$$

Er dient zusammen mit dem Korrelationskoeffizienten zur Berechnung der Datenqualitätsverbesserung bzw. -verschlechterung. In beiden Fällen bilden die selektierten Datenwerte neue Datenqualitätsfenster, in denen die Tupelqualität gemittelt wird.

Theorem 5.15 Die Genauigkeit und Vollständigkeit eines Datenstromfensters wird bei der Selektion als Durchschnitt der eingehenden Tupelgenauigkeiten bzw -vollständigkeiten berechnet, die mit Hilfe des Korrelationskoeffizienten ς und der Datenverschiebung $\Delta\mu$ gewichtet werden.

$$q_w(k) = \frac{1}{\omega_k} \sum q(j \in x_{input}) \;\; q \in \{a,c\} \tag{5.46}$$

$$q(j) = q_w(k) \cdot (1 + \varsigma \cdot \Delta\mu) \;\; q \in \{a,c\} \tag{5.47}$$

Konfidenz

Wird die Selektionsbedingung erfüllt, wird das entsprechende Tupel in den Ergebnisdatenstrom aufgenommen. Andernfalls wird das Tupel aus dem Datenstrom entfernt. Die Auswertung der Selektionsbedingung erinnert an den Schwellwertvergleich in Abschnitt 5.2.2. Systematische und statistische Messfehler definieren einen unsicheren Bereich δ um die Schwellwertfunktion. Im Kontext der Selektion hat der unsichere Bereich zwei Arten von Fehlentscheidungen zur Folge (siehe Abbildung 5.11). Sensormesswerte können selektiert werden, obwohl der wahre Messwert die Selektionsbedingung nicht erfüllt. Datensätze können aus dem Strom entfernt werden, obwohl der wahre Wert der Selektionsbedingung entspricht.

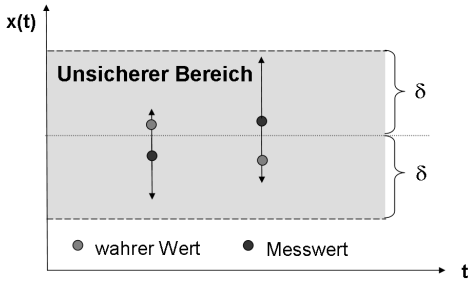


Abbildung 5.11.: Selektionsfehler im unsicheren Bereich

Sind die Datensätze im unsicheren Bereich gleich verteilt, gleichen sich die Fehler aus, da ebenso viele Datensätze falsch selektiert wie gelöscht werden. Anderenfalls - was weit häufiger der Fall ist - führen die fehlerhaften Selektionen zu Fehlern in der nachfolgenden Datenverarbeitung. Das Ergebnis einer nachfolgenden Aggregation ist höher als das wahre Ergebnis, wenn Datensätze ausgewählt wurden, obwohl sie dem Selektionskriterium nicht entsprechen. Es ist zu niedrig, wenn relevante Datensätze gelöscht wurden, obwohl der wahre Wert die Selektionsbedingung erfüllt.

Da die Messfehler im unsicheren Bereich zufällig verteilt sind, entspricht die Selektion hier einem Sampling der wahren Datenwerte. Somit können die Methoden zur Bestimmung eines Samplingfehlers angewendet werden, um den Selektionsfehler im unsicheren Bereich zu berechnen. Analog zum Samplingfehler wird er zum statistischen Messfehler in der Datenqualitätsdimension Konfidenz addiert. Gleichung 5.24 auf Seite 97 wird für die Selektion angepasst, indem die Samplingrate r_{sa} durch die Selektionsrate r_{se} ersetzt wird. Sie gibt den Anteil der selektierten Datensätze der eingehenden Datenqualitätsfenster an.

$$\epsilon^{+}_{w'} = \phi \cdot \frac{s(w')}{\sqrt{\omega \cdot r_{se}}} \cdot \sqrt{1 - r_{se}}. \tag{5.48}$$

Theorem 5.16 Die Fensterkonfidenz der Selektion wird analog dem Sampling als Summe der Konfidenzen ϵ_w der in das Ergebnisfenster eingehenden Tupel x_{input} und des Selektionsfehlers ϵ_w^+ mit Gleichung 5.25 berechnet.

5.4.4. Verbund

Es existiert eine Vielzahl an Verfahren, um den Verbund zweier Relationen effektiv und schnell zu berechnen. Die Behandlung der Datenqualitätsinformationen erfolgt jedoch unabhängig von der jeweiligen Verbundtechnik. Im Folgenden wird zuerst der zeitstempelbasierte Verbund zweier Datenströme betrachtet. Danach wird die Qualitätsberechnung beim Datenstromverbund auf Basis ungeordneter Verbundattribute vorgestellt. Zum Schluss wird die Datenqualitätsberechnung beim Verbund zweier Datenbankrelationen beschrieben.

Durch eine sequentielle Verkettung mehrerer Verbundoperatoren lassen sich beliebig viele Datenströme zusammenfassen. Die Reihenfolge der Zusammenfassung hat keinen Einfluss auf die resultierende Datenqualität.

Verbund synchroner Datenströme

Die einfachste Verbundtechnik setzt synchrone Datenströme voraus. Verbundpartner werden basierend auf identischen Zeitstempeln zusammengefasst (siehe Abbildung 5.12). Gleiche Datenraten genügen nicht für diesen Ansatz, da die Sensormessungen zeitlich gegeneinander verschoben sein könnten, so dass keine identischen Zeitstempelpaare existieren.

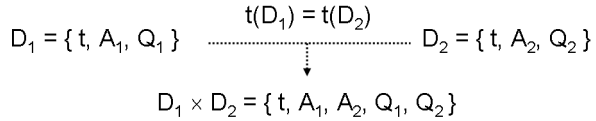


Abbildung 5.12.: Einfacher synchroner Datenstromverbund

Während des Verbundes zweier Datenströme D_1 und D_2 werden die Datenqualitätsinformationen Q_1 und Q_2 nicht verändert, sondern analog zu den Attributen A_1 und A_2 in den resultierenden Datenstrom kopiert. Da das zugrunde liegende Metadatenmodell DQMx unterschiedliche Fenstergrößen für Qualitätsinformationen unterschiedlicher Attribute erlaubt, treten keine Probleme beim Zusammenführen der Datenqualitätsströme auf.

Theorem 5.17 Der zeitstempelbasierte Verbund zweier synchroner Datenströme hat keinen Einfluss auf die Datenqualitätsinformationen, so dass für alle DQ-Dimensionen gilt: $q'_w(k) = q_w(k)$.

Verbund asynchroner Datenströme

Synchrone Datenströme treten in praktischen Anwendungen sehr selten auf. Deshalb wird in [SFL05] eine zeitstempelbasierte Verbundtechnik vorgestellt, die speziell auf die Anforderungen von asynchronen Datenströmen ausgerichtet ist. Sampling und Interpolation werden genutzt, um Datenströme mit unterschiedlichen Datenraten anzugleichen, Verschiebungen der Zeitstempel zu überwinden und somit Verbundpartner mit gleichem Zeitstempel zu finden.

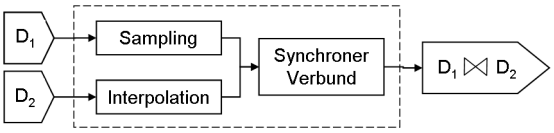


Abbildung 5.13.: Verbund asynchroner Datenströme

Abbildung 5.13 zeigt die Aufteilung des komplexen asynchronen Verbundoperators. Die Datenströme D_1 und D_2 werden durch ein Sampling abgetastet oder interpoliert und nachfolgend mit Hilfe des einfachen synchronen Datenstromverbunds verknüpft. Zur Quantifizierung des Einflusses auf die Datenqualität muss die Datenqualitätsalgebra der Basisoperatoren Sampling und Interpolation angewendet werden (siehe Abschnitt 5.3.1 bzw. 5.3.2).

Datenstromverbund auf Basis ungeordneter Verbundattribute

Ist das Verbundattribut nicht wie der Zeitstempel geordnet, muss der Datenstromverbund mit Hilfe von gleitenden Stromfenstern ausgeführt werden, da der ungeordnete Verbund unendlicher Datenströme nicht definiert ist. Die Ressourcenbeschränkung erlaubt nur die Untersuchung begrenzter Datenstrompartitionen. Die restlichen Datentupel müssen mittels eines Verfahrens zum Lastausgleich entfernt werden, so dass mit hoher Wahrscheinlichkeit Verbundpartner in einem oder beiden eingehenden Datenströmen verloren gehen. Der fensterbasierte Verbund beinhaltet somit ein implizites Sampling in einem oder beiden involvierten Datenströmen. Die implizite Samplingrate bestimmt den durch dieses Sampling eingefügten statistischen Fehler der Konfidenz.

Abbildung 5.14 zeigt die Beziehung zwischen gleitenden Fenstern des Datenstromverbundes und aufeinander folgenden Datenqualitätsfenstern. Während gleitendes Verbundfenster und Datenqualitätsfenster überlappen (Abbildung 5.14 a), werden Verbundpartner ermittelt und in den Ergebnisstrom geschrieben. Datenqualitätsinformationen werden für jedes Attribut separat behandelt. Allerdings müssen Datenstromfenster, die Tupel durch das implizite Sampling verloren haben, wieder zur vollen Länge aufgefüllt werden. Die Fensterdatenqualität wird dabei wie beim Sampling als Durchschnitt der eingehenden DQ-Informationen berechnet. So ergibt sich für das erste Ergebnisfenster von Attributstrom 1 die Datenmenge $d'_w = 1/4 \cdot (2d_{11} + 2d_{12})$.

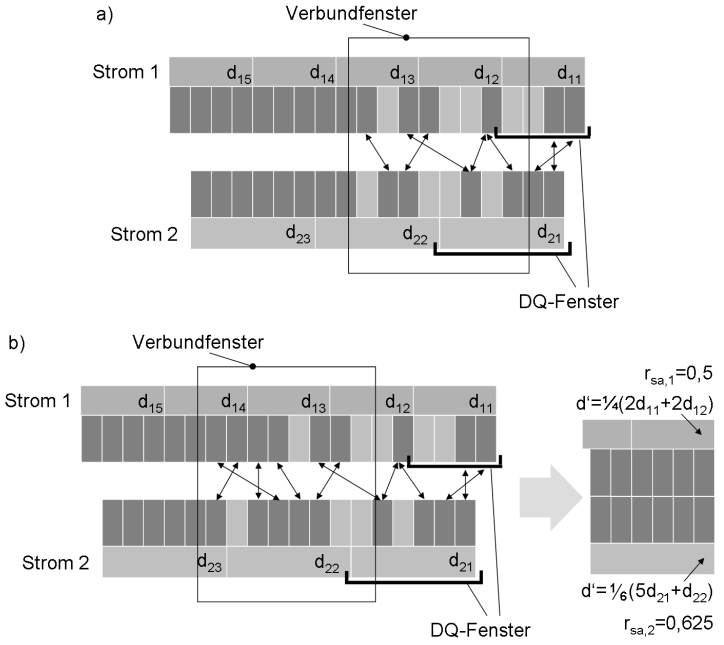


Abbildung 5.14.: Verbund gleitender Datenstromfenster

Sobald ein DQ-Fenster mit gefundenen Verbundpartnern aufgefüllt wurde, wird die implizite Samplingrate des Datenstroms ermittelt. In Abbildung 5.14 wurden zum Beispiel 4 von 8 Datentupel aus Strom 1 entfernt, so dass gilt $r_{sa,1} = 4/8 = 0,5$. Nach drei weiteren Zeitschritten ist auch das DQ-Fenster von Strom 2 gefüllt, so dass dessen Samplingrate bestimmt werden kann. Mit Hilfe von Samplingrate und Varianz der erhaltenen Datentupel wird der eingefügte Samplingfehler entsprechend Gleichung 5.24 für jeden Attributstrom berechnet und die DQ-Informationen aktualisiert.

Abschnitt 6.1 befasst sich intensiv mit dem Problem der Ressourcenbeschränkung in Verbundoperatoren und Aggregationen. Es werden Verfahren vorgestellt, die die Ergebnisqualität des ungeordneten Datenstromverbundes durch Integration der verfügbaren Datenqualitätsinformationen erhöhen.

Relationenverbund

Der Verbund zweier Datenbankrelationen entspricht im Vorgehen dem ungeordneten Verbund beliebiger Attribute. Die Datenqualitätsberechnung wird vereinfacht, da kein gleitendes Datenstromfenster benötigt wird. Allerdings wird der Verbund in Blöcken entsprechend der Datenqualitätsfenster ausgeführt. Datentupel werden in Blöcken gelesen und verarbeitet, so dass die implizite Samplingrate und damit der eingefügte statistische Fehler für jedes DQ-Fenster bestimmt werden kann.

Abbildung 5.15 zeigt den fensterbasierten Relationenverbund der Sensordaten Öltemperatur und Viskosität. In diesem Beispiel werden Tupel aus Relation $R2$ (Viskosität) entfernt, so dass der statistische Messfehler aller Viskositätsmessungen entsprechend Theorem 5.6 erhöht werden muss. Der eingefügte Samplingfehler berechnet sich aus der Standardabweichung $\sigma = 13,6$ des erzeugten DQ-Fensters, der Samplingrate $r_{sa} = 0,5$ und der Fenstergröße $\omega = 4$ zu $\epsilon^+ = 8,8$.

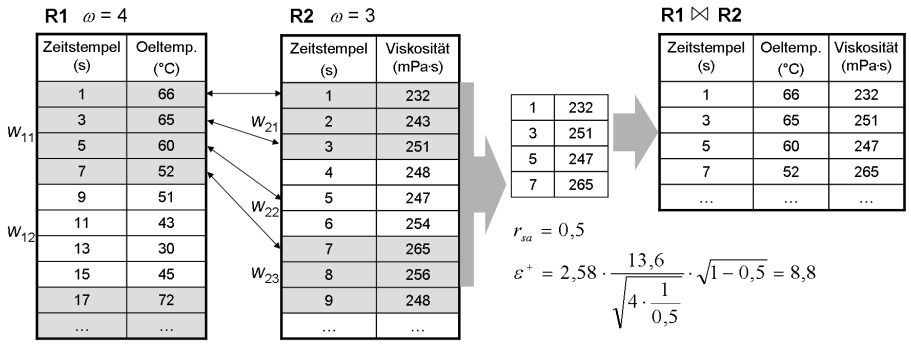


Abbildung 5.15.: Verbund zweier Relationen

Soll der Verbund auf Basis des Zeitstempels erfolgen, muss außerdem wie beim asynchronen Verbund vor der Ausführung geprüft werden, ob eine Interpolation notwendig ist, um eine nichtleere Ergebnismenge zu erzeugen. Durch Vergleich der zu verknüpfenden Tabellen wird die minimal notwendige Interpolationsrate bestimmt und die Datenqualitätswerte entsprechend Abschnitt 5.3.2 angepasst.

5.4.5. Aggregation

Bei der Aggregation werden Datensätze eines Datenstromattributes zusammengefasst, um das Datenvolumen zu verringern oder höherwertige Informationen zu extrahieren. Meist erfolgt die Aggregation nicht über das gesamte Datenvolumen, sondern für definierte Gruppen von Datensätzen, die mit Hilfe eines Datenstromattributes gebildet werden.

In der Sensordatenverarbeitung ist der Zeitstempel das meist genutzte Gruppierungsattribut. Er erlaubt Gruppierungen, die eine vorgegebene Tupelanzahl (z.B. 100 Datensätze) zusammenfassen oder einen bestimmten Zeitabschnitt (z.B. 10min) repräsentieren. Das Ergebnis einer solchen Aggregation beschreibt nicht nur einen Zeitpunkt, sondern ein Zeitintervall $[t_{begin}, t_{end}]$, das im Zeitstempel des Aggregationsergebnisses wiedergegeben werden muss. Variierende Datenstromraten führen zu unterschiedlichen Gruppengrößen $l_i (\sum l_i = m)$, unabhängig von der Größe ω der aggregierten Datenstromfenster.

Die im Folgenden vorgestellten Methoden der Datenqualitätsverarbeitung gelten nicht nur für die oben beschriebene zeitstempelbasierte Aggregation, sondern für beliebige Gruppierungsattribute. Tabelle 5.3 fasst die untersuchten Aggregationsfunktionen

zusammen. Neben traditionellen Aggregationen wie Summation oder Minimalwertfindung wird die Datenqualitätsverarbeitung für komplexe Aggregationsfunktionen anhand der Anstiegsberechnung vorgestellt.

Aggregation	Kurzbeschreibung
avg	Durchschnitt einer Menge von Datenwerten
sum	Summation der Datensätze
min	Minimalwert einer Datenmenge
max	Maximalwert einer Datenmenge
count	Tupelanzahl der Datenmenge
slope	Anstiegsberechnung für eine Menge von Sensormesswerten

Tabelle 5.3.: Liste der untersuchten Aggregationsfunktionen

Die Datenqualitätsverarbeitung der Aggregation gliedert sich ähnlich der Signaloperatoren in zwei Schritte, die in Abbildung 5.16 dargestellt sind. Fenster der Größe $\omega = 5$ werden in Gruppen der Größe $l = 4$ zusammengefasst. Zuerst wird die Qualität q eines *Aggregates* - dem Aggregationsergebnis einer Gruppe g_i - bestimmt. Die berechneten Aggregate bilden die neuen Datenqualitätsfenster, für die im zweiten Schritt die fensterbasierte Ergebnisqualität q'_w mit Hilfe der Durchschnittsberechnung ermittelt werden muss.

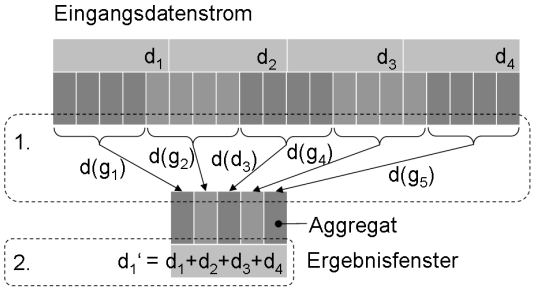


Abbildung 5.16.: Datenqualitätsberechnung bei der Aggregation

Theorem 5.18 Während der Aggregation ist die resultierende Fensterdatenqualität $q'_w(k)$ für jede Datenqualitätsdimension als Durchschnitt der Datenqualitätswerte der eingehenden Aggregate $q(g_i)$ definiert.

$$q'_w(k) \quad = \quad \frac{1}{\omega_k} \cdot \sum_{i=t_b}^{t_e} q(g_i) \tag{5.49}$$

Die Berechnung der Datenmenge und Vollständigkeit eines Aggregates $q(g_i)$ ist generisch für alle Aggregationsfunktionen. Die Verarbeitung von Genauigkeit und Konfidenz

muss hingegen spezifisch für die jeweilige Aggregationsfunktion definiert werden. Die Gruppierung über der Zeitachse erlaubt die Aggregation gleitender Datenstromfenster, die in Datenstromanwendungen häufig zum Einsatz kommt. Wird ein anderes Attribut zur Gruppierung der Aggregation genutzt, muss auf die fensterbasierte Aggregationsberechnung zurückgegriffen werden, um den Datenstrom nicht zu blockieren. Am Ende dieses Abschnitts wird erläutert, wie die im Anschluss vorgestellten Verfahren auf diesen Aggregationsansatz erweitert werden können.

Datenmenge und Vollständigkeit

Die Datenmenge eines Aggregates fasst die Datenmengen der zur Berechnung genutzten Datensätze zusammen. Alle eingehenden Tupeldatenmengen werden aufsummiert. Die Bedeutung dieser DQ-Dimension wird vor allem bei der Aggregation mit variierender Gruppengröße ersichtlich. Hohe Datenstromraten führen bei zeitlich definierten Gruppierungen (z.B. $10min$) zu hohen Datenmengen und umgekehrt.

Theorem 5.19 Die Datenmenge $d(g_i)$ des Aggregationsergebnisses der Gruppe g_i ist als Summe der eingehenden Datenmengen bestimmt.

$$d(g_i) \quad = \quad \sum d(j \in g_i) \tag{5.50}$$

Die Vollständigkeit eines Aggregationsergebnisses fasst die Vollständigkeiten der eingehenden Datensätze zusammen. Datentupel mit hoher bzw. niedriger Vollständigkeit gehen mit gleicher Wertigkeit in das Aggregationsergebnis ein. Hier wird die Berechnung der Vollständigkeit nach [BNQ06] verwendet (siehe Abschnitt 3.3.2).

Theorem 5.20 Die Vollständigkeit $c(g_i)$ des Aggregationsergebnisses der Gruppe g_i ist als Durchschnitt der eingehenden Vollständigkeiten definiert.

$$c(g_i) \quad = \quad \frac{1}{l_i} \sum c(j \in g_i) \tag{5.51}$$

Summation und Durchschnittsberechnung

Hinsichtlich der Datenqualitätsberechnung eines Aggregationsergebnisses ähneln Summation und Durchschnitt der algebraischen Addition. Genauigkeit bzw. Konfidenz eines algebraischen Operators sind als lineare bzw. quadratische Summe aller eingehenden Tupelgenauigkeiten bzw. -konfidenzen definiert, die mit den entsprechenden partiellen Ableitungen gewichtet sind (Gleichungen 5.3 und 5.4). Bei der Summation reduzieren sich alle partiellen Ableitungen zu 1, bei der Durchschnittsberechnung zum Reziproken der Gruppenlänge $1/l_i$.

Theorem 5.21 Genauigkeit $a(g_i)$ und Konfidenz $\epsilon(g_i)$ der Summation bzw. des Durchschnitts der Gruppe g_i sind als Summe bzw. Durchschnitt der eingehenden Tupelgenauigkeiten bzw. -konfidenzen definiert.

$$sum: \quad q(g_i) = \sum q(j \in g_i) \qquad avg: \quad q(g_i) = \frac{1}{l_i} \sum q(j \in g_i) \quad q \in \{a, \epsilon\} \tag{5.52}$$

Minimal- und Maximalwertberechnung

Bei der Minimal- und Maximalwertberechnung wird der kleinste bzw. größte Wert einer Gruppe gesucht. Die Genauigkeit und Konfidenz eines solchen Aggregationsergebnisses entsprechen exakt der Genauigkeit bzw. Konfidenz des gefundenen Wertes.

Theorem 5.22 Genauigkeit $a(g_i)$ und Konfidenz $\epsilon(g_i)$ des Ergebnisaggregates der Minimal- bzw Maximalwertberechnung in der Gruppe g_i werden durch die Genauigkeit bzw. Konfidenz des gefundenen Minimal- bzw. Maximalwertes $x(j)$ bestimmt.

$$min: \quad q(g_i) = q(j|x(j) = min(g_i)) \quad q \in \{a, \epsilon\} \tag{5.53}$$

$$max: \quad q(g_i) = q(j|x(j) = max(g_i)) \quad q \in \{a, \epsilon\} \tag{5.54}$$

Tupelanzahl

Die Aggregation *count()* bestimmt die Anzahl der Tupel mit jeweils gleicher Wertausprägung. So ist es zum Beispiel bei der Umsatzanalyse interessant, wieviele Produkte jedes Typs verkauft wurden. Bei der zeitstempelbasierten Aggregation würde diese Aggregation lediglich die Gruppengröße, also den schon vorher bekannten Zeitrahmen, wiedergeben. Da fehlende Datenstromtupel vom Sensorknoten interpoliert werden (siehe Abschnitt 2.2), treten bei der Zählung keine systematischen oder statistischen Fehler auf. Bei der Benutzung eines anderen Gruppierungsattributs muss die Aggregation fensterbasiert ausgeführt werden, um den blockierenden Charakter des Aggregationsoperators in der Datenstromverarbeitungskette zu überwinden. Im Kontext der Aggregation eines Datenstromfensters treten ebenfalls weder systematische noch statistische Fehler bei der Tupelzählung auf.

Theorem 5.23 Die Genauigkeit $a(g_i)$ und Konfidenz $\epsilon(g_i)$ der Tupelzählung in einer Gruppe g_i sind $a(g_i) = \epsilon(g_i) = 0$.

Anstiegsberechnung

Zur Berechnung des Anstiegs innerhalb eines gegebenen Zeitrahmens wird der Messwertstrom mit Hilfe der linearen Funktion $x = m \cdot t + a$ approximiert, die durch die Methode der kleinsten Quadrate wie folgt bestimmt wird.

$$m = \frac{\sum_{j \in g_i}(t(j) - \bar{t}) \cdot (x(j) - \bar{x})}{\sum_{h \in g_i}(t(h) - \bar{t})^2} \tag{5.55}$$

$$a = \bar{x} - m \cdot \bar{t} \tag{5.56}$$

Genauigkeit und Konfidenz der Anstiegsberechnung m können mit Hilfe der Gauß'schen Fehlerfortpflanzung aus Gleichung 5.55 berechnet werden.

Theorem 5.24 Genauigkeit $a(g_i)$ und Konfidenz $\epsilon(g_i)$ der Anstiegsberechnung der Gruppe g_i werden mit Hilfe der eingehenden Tupelgenauigkeiten $a(j)$ bzw. -konfidenzen $\epsilon(j)$ und der partiellen Ableitung des Anstiegs nach $x(j)$ berechnet.

$$a(g_i) = \sum_{j \in g_i} \left| \frac{\partial m}{\partial x(j)} \right| \cdot a(j) \tag{5.57}$$

$$\epsilon(g_i) = \sqrt{\sum_{j \in g_i} (\frac{\partial m}{\partial x(j)})^2 \cdot \epsilon(j)^2} \tag{5.58}$$

$$\frac{\partial m}{\partial x(j)} = \frac{t(j) - \bar{t}}{\sum_{h \in g_i}(t(h) - \bar{t})^2} \tag{5.59}$$

Aggregation in Datenstromfenstern

Abbildung 5.17 zeigt eine gleitende Fensteraggregation der Länge $l = 6$, Schrittweite $s = 1$ und Datenqualitätsfenstergröße $\omega = 4$. Unabhängig von der Größe des gleitenden Fensters bilden die Aggregationsergebnisse fortlaufend neue Datenqualitätsfenster.

Die fensterbasierte Aggregation beeinflusst nicht die Berechnung der fensterbasierten Datenqualitätsinformationen. Die oben beschriebenen Strategien zur Berechnung von Datenmenge, Vollständigkeit sowie Genauigkeit und Konfidenz können ohne Änderungen angewendet werden. Im Vergleich zur traditionellen Aggregation in festen Gruppen variiert lediglich die Anzahl der Ergebniswerte. Die Datenqualitätsinformationen werden nicht mehr nach $l \cdot \omega$ Eingangsdatentupeln, sondern aufgrund der überlappenden Gruppierungen nach der Verarbeitung der kleineren Menge von $l + s \cdot (\omega - 1)$ Datensätzen gesendet.

5.5. Zusammenfassung

Dieses Kapitel entwirft eine Datenqualitätsalgebra für die Verarbeitung von Sensordatenströmen. Für die Beurteilung der Datenqualität der Verarbeitungsergebnisse müssen alle ausgeführten Operatoren auf den im Datenstrom propagierten bzw. in der Datenbank

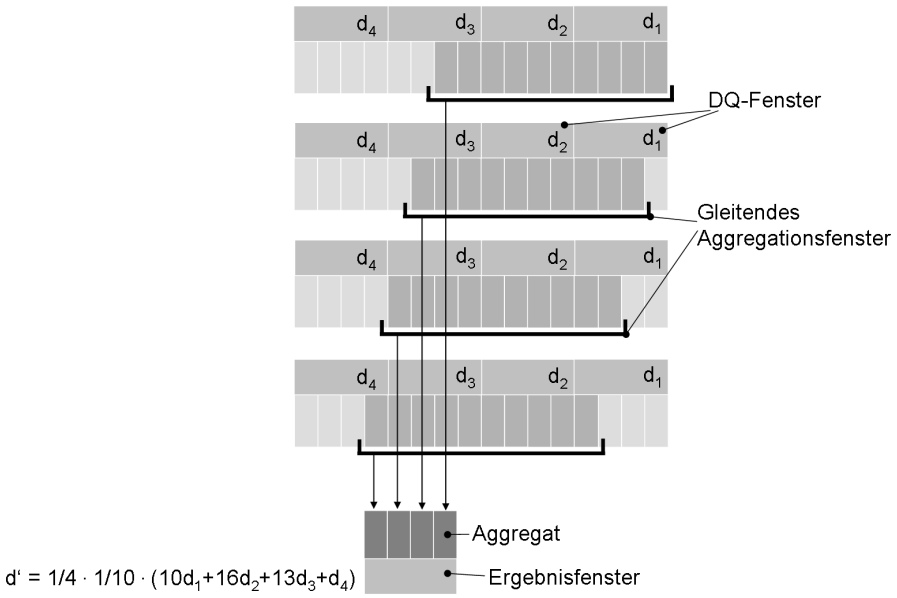


Abbildung 5.17.: Aggregation in gleitenden Datenstromfenstern

gespeicherten Datenqualitätsdimensionen nachvollzogen werden. Besondere Beachtung muss dabei auf Operatoren gelegt werden, die die Qualitätsausprägungen beeinflussen. Datenqualitätsprobleme wie Unvollständigkeiten oder numerische Messfehler können kombiniert und vergrößert werden.

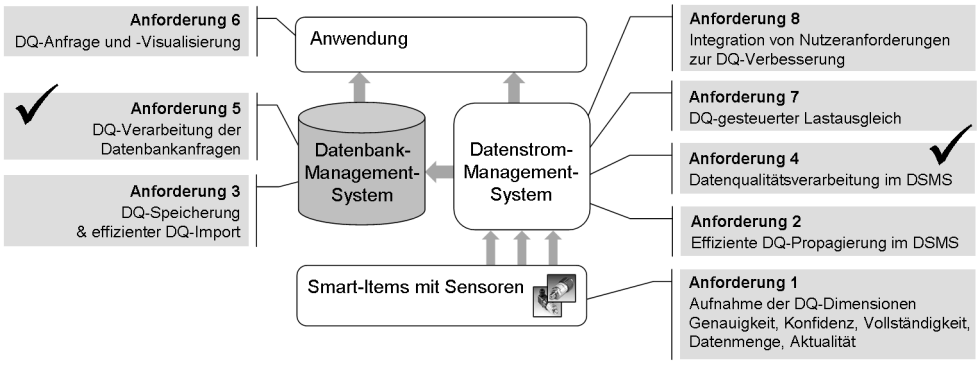


Abbildung 5.18.: Erfüllte Anforderungen 4 und 5

Alle in der Anforderungsanalyse identifizierte Operatoren der der Datenstrom- und Datenbankanfragesprachen CQL und SQL, der Signalverarbeitung sowie der Numerik wurden mit Theoremen zur Datenqualitätsverarbeitung untermauert. Anforderung 4 der Datenqualitätsverarbeitung in Datenströmen und Anforderung 5 der Qualitätsberechnung von Datenbankanfragen sind damit erfüllt. Um die Datenqualität der Ergebnisse

eines komplexen Verarbeitungsgraphen zu ermitteln, müssen diese Theoreme in der Reihenfolge der Operatoren auf die propagierten Datenqualitätsdimensionen angewendet werden.

6
Verbesserung der Sensordatenqualität

Neben der Bereitstellung von Datenqualitätsinformationen zur Bewertung von Sensordaten und abgeleiteter Informationen ist die Verbesserung der Datenqualität eine wichtige Aufgabe im Qualitätsmanagement. In Abschnitt 3.2 wurden verschiedene Verfahren des Data-Cleaning und der Online-Integration zur Verbesserung der Datenqualität in Informationssystemen vorgestellt. Sie benötigen allerdings umfangreiche Ressourcen, Referenzdatenquellen und Verarbeitungszeit, so dass diese Verfahren nicht in Datenstromsystemen anwendbar sind.

In diesem Kapitel werden zwei neue Möglichkeiten vorgestellt, die Datenqualität in Sensordatenströmen zu verbessern. Abschnitt 6.1 schildert den datenqualitätsgesteuerten Lastausgleich, der Überlastsituationen durch das Entfernen minderwertiger Datentupeln ausgleicht und damit die durchschnittliche Stromqualität anhebt. Abschnitt 6.2 befasst sich mit der qualitätsgesteuerten Optimierung der Datenstromverarbeitung, um Nutzeranforderungen an die Datenqualität zu integrieren. Operatoren des DQMx werden bezüglich der untersucht. Das Optimierungsproblem wird definiert und Heuristiken zu dessen Lösung werden vorgestellt.

6.1. Datenqualitätsgesteuerter Lastausgleich

Der Lastausgleich ist ein wichtiges Werkzeug zur Sicherung der Dienstqualität in Datenstromsystemen. Das Entfernen überzähliger Datentupel beseitigt die Überlast, führt jedoch zu hohem Datenverlust, der die Datenqualität von Verarbeitungsergebnissen signifikant reduziert. So können Aggregationsergebnisse nicht genau bestimmt werden und Verbundpartner im Datenstromverbund verloren gehen.

Der neue Ansatz des datenqualitätsgesteuerten Lastausgleichs (engl. data quality-driven load shedding, DQLS) nutzt Datenqualitätsinformationen, um die allgemeine Datenqualität von Verarbeitungsergebnissen durch Entfernen minderwertiger Tupel zu verbessern. Ziel ist die Erhöhung der Qualität der Anfrageergebnisse im Vergleich zu bestehenden Load-Shedding-Strategien. Zuerst wird die Ordnung der Datentupel auf Basis der Qualitätsinformationen beschrieben. Nach der Einführung des allgemeinen Ansatzes des DQLS, wird die Berechnung der Qualitätsverbesserung des Anfrageergebnisses erläutert. Danach werden drei spezifische Algorithmen für den Lastausgleich bei Aggregations- und Verbundanfragen vorgestellt.

6.1.1. Qualitätsgesteuerte Ordnung

Um den datenqualitätsgesteuerten Lastausgleich zu ermöglichen, müssen Datenstromtupel hinsichtlich ihrer Datenqualität geordnet werden. Abschnitt 2.4.2 stellt Dimensionen zur Beschreibung der Datenqualität in Sensordatenströmen auf. Genauigkeit und Konfidenz beschreiben die systematischen und statistischen Fehler der Sensormessungen. Sie werden im Folgenden unter dem Begriff Korrektheit $\alpha = a + \epsilon$ zusammengefasst. Hohe Fehlerwerte implizieren hier eine geringe Datenqualität. Die Vollständigkeit überwacht den Anteil originaler Sensormesswerte im Datenstrom; die Datenmenge beschreibt die Menge der genutzten Rohdaten zur Bestimmung eines Datentupelwertes. Je mehr Rohdaten aggregiert werden, umso vertrauenswürdiger ist das Ergebnis. Hohe Werte der Vollständigkeit und Datenmenge bedeuten demnach eine hohe Datenqualität.

Die skalaren Wertausprägungen der DQ-Dimensionen werden unterschiedlich beurteilt. Um eine einheitliche Vorstellung von „hoher Datenqualität“ zu erhalten, müssen diese Bewertungen harmonisiert werden. Kleine Wertausprägungen der Qualitätsinformation q bedeuten nachfolgend hohe Datenqualität. Alle Dimensionen, die dieser Beurteilung nicht entsprechen, werden durch ihre Negation ausgedrückt: $q \rightarrow -q$. Unter Beachtung dieser Qualitätsbewertung kann die datenqualitätsgesteuerte Ordnung auf beliebige Mengen von Datenqualitätsdimensionen angewendet werden.

Die Gesamtqualität eines Datentupels fasst die Einzeldimensionen zu einem skalaren Wert $\theta = \theta(q_1, q_2, \ldots, q_\vartheta)$ zusammen, der zur Vereinheitlichung der Interpretationen „sehr gut“ bis „sehr schlecht“ herangezogen wird. Definition 6.3 stellt die Funktion $\theta()$ zur Berechnung der Gesamtqualität von Sensordaten vor. Formell definiert θ nur eine Quasiordnung, da keine Rangfolge für Tupel gleicher Gesamtqualität gegeben ist. Dies hat allerdings keinen negativen Einfluss auf den geschilderten Ansatz, da Tupel gleicher Qualität beliebig, z.B. aufgrund ihres Zeitstempels, geordnet werden können.

Definition 6.1 Ein Datenstromtupel x_1 ist qualitativ hochwertiger als Tupel x_2 (ausgedrückt in der Binärrelation $\succ$), wenn die Gesamtqualität von x_1 die Gesamtqualität von x_2 übertrifft, d.h. der skalare Wert der Gesamtqualität θ_1 kleiner ist als θ_2.

$$x_1 \succ x_2 \equiv \theta_1 < \theta_2 \tag{6.1}$$

6.1.2. Allgemeiner Ansatz

Die Load-Shedding-Strategie von Babcock et al. [BDM04] zielt auf die geringste Verarbeitungszeit pro Datenstromtupel ab. Sie bestimmt die optimale Verteilung der Load-Shedding-Operatoren und beantwortet damit die Fragen, wo und wieviele Datentupel zum Lastausgleich entfernt werden müssen. Die Strategie basiert auf der Erkenntnis, dass das optimale Load-Shedding zu Beginn geteilter Anfragesegmente erfolgen muss. Geteilte Anfragesegmente sind Operatorgruppen, die in mehreren Anfragebäumen vorkommen (siehe Abbildung 6.1). Der höchste auszuführende Lastausgleich aller Anfragen und damit die höchste effektive Ausgleichsrate P_{max} bestimmt den Lastausgleich an der Wurzel des geteilten Anfragesegmentes. Die Raten der übrigen Load-Shedding-Operatoren werden als P_{child}/P_{max}, z.B. $P_1/P_{max} = 0.5/0.8 = 0.625$ definiert.

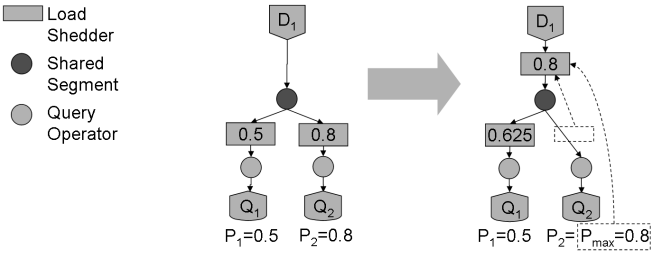


Abbildung 6.1.: Optimales Load-Shedding-Schema

Die resultierende Load-Shedding-Struktur bleibt stabil so lange sich die Arbeitslast, das heißt die Struktur der kontinuierlich parallel auszuführenden Anfragen, nicht verändert. Damit wird das Load-Shedding-Problem auf die Frage reduziert, *welche* Datentupel aus dem Strom entfernt werden müssen. Es ist verlockend, die Datentupel einfach auf Basis ihrer Gesamtdatenqualität, zusammengesetzt aus verfügbaren DQ-Dimensionen, zur Unterscheidung „guter“ und „schlechter“ Tupel zu ordnen. Das naive Sortieren aller Datenstromtupel würde jedoch den Prozess der Datenverarbeitung blockieren und kann somit nicht genutzt werden. Die Beurteilung eines aktuell betrachteten Datentupels muss auf der Qualitätsverteilung aller bisher verarbeiteten Datentupel beruhen, ohne zukünftige Tupelinformationen zu benötigen.

Abbildung 6.2 stellt die kumulative Wahrscheinlichkeitsdichteverteilung der Datenqualität dar. Die benötigte Load-Shedding-Rate r_{LS} bestimmt den Anteil der im Strom verbleibenden Datentupel, d.h. den Anteil qualitativ hochwertiger Tupel, die nicht gelöscht werden sollen. Durch Anlegen der Rate r_{LS} an die Dichteverteilung wird die Datenqualitätsschranke b definiert. Sie trennt hochqualitative Tupel, die im Strom erhalten bleiben, von Datentupeln niederer Qualität, die entfernt werden müssen. Die Qualitätsverbesserung θ^+ wird durch die Verminderung der durchschnittlichen Gesamtqualität vor $\mu(\theta)$ und nach $\mu'(\theta)$ dem Load-Shedding ausgedrückt: $\theta^+ = \mu(\theta) - \mu'(\theta)$.

Der Lastausgleich basiert auf der Distanz zwischen der Gesamtqualität des aktuell untersuchten Datenqualitätsfensters und der DQ-Schranke b. Diese Distanz bestimmt

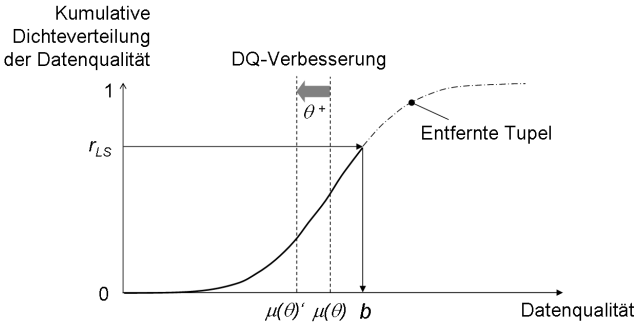


Abbildung 6.2.: Datenqualitätsschranke und -verbesserung

die Wahrscheinlichkeit, mit der Tupel des DQ-Fensters aus dem Strom entfernt werden, so dass durchschnittlich mehr Datentupel schlechter DQ-Fenster als aus Fenstern guter Qualität gelöscht werden. Unter der Annahme unabhängiger Qualitätsinformationen und Datenstromwerte erzeugt der datenqualitätsgesteuerte Lastausgleich zufällige Stichproben überlasteter Datenstrompartitionen. Das Bernoulli-Sampling wird individuell für jedes Tupel eines DQ-Fensters angewendet, sobald die Qualitätsinformationen im Datenqualitätstupel τ_q zur Verfügung stehen.

Definition 6.2 Der datenqualitätsgesteuerte Lastausgleich ist eine Load-Shedding-Strategie, die ein Bernoulli-Sampling des Datenstroms ausführt, so dass ein Tupel mit der Wahrscheinlichkeit $p_{Bernoulli}$ im Datenstrom verbleibt. Dabei wird $p_{Bernoulli} = d_{\succ}(\theta, b)$ mit Hilfe der Distanzfunktion $d_{\succ}$ aus aktueller fensterbasierter Gesamtqualität θ und der Datenqualitätsschranke b berechnet.

Unter der Annahme unabhängiger Qualitätsinformationen und Datenstromwerte erzeugt der DQLS zufällige Stichproben überlasteter Datenstrompartitionen. Das Bernoulli-Sampling wird individuell für jedes Tupel eines DQ-Fensters ausgeführt. Der Lastausgleich basiert auf der Verteilung vergangener Qualitätsbeobachtungen. Sollte die Datenqualität im Laufe der Datenstromverarbeitung abnehmen, würden Datentupel, die zu Beginn als schlechte Tupel entfernt wurden, später als gut eingestuft werden. Dies folgt aus der quasi-statischen Analyse von Momentaufnahmen der Qualitätsverteilung im Datenstrom, die zur kontinuierlichen Entfernung von Überlastvolumen erforderlich ist. Ein global optimales Load-Shedding würde die Analyse des vollständigen Datenstroms benötigen, dies widerspricht klar der Absicht der kontinuierlichen Datenstromverarbeitung. Im Zuge der vorliegenden Arbeit wurden drei Algorithmen des datenqualitätsgesteuerten Lastausgleichs entwickelt, die auf unterschiedliche Qualitätskriterien abzielen.

Im Folgenden werden drei Algorithmen des datenqualitätsgesteuerten Lastausgleichs vorgestellt, die auf unterschiedliche Qualitätskriterien abzielen: die Gesamtqualität, die Korrektheit von Aggregationsergebnissen und die Vollständigkeit des Datenstromver-

bundes. Für jeden Load-Shedding-Algorithmus wurde eine spezifische Definition der Distanzfunktion $d_{\succ}$ entwickelt.

MaxDQ maximiert die durchschnittliche Gesamtqualität von Aggregationen, indem Datenstromtupel entsprechend ihrer Datenqualität aus dem Strom entfernt werden. Die Load-Shedding-Rate bestimmt die Datenqualitätsschranke, so dass schlechte Tupel mit höherer Wahrscheinlichkeit gelöscht werden.

MaxDQcompensate verfolgt eine ähnliche Strategie wie MaxDQ. Jedoch wird hier die Korrektheit als Summe der Dimensionen Genauigkeit und Konfidenz priorisiert, um den statistischen Fehler, der durch den Datenverlust beim Load-Shedding eingeführt wird, auszugleichen.

MaxCompleteness ist auf den Datenstromverbund ausgerichtet. Die Vollständigkeit des Verbundergebnisses wird optimiert, indem die Größe der Verbundergebnismenge maximiert und die Anzahl der fehlenden Messwerte mit Hilfe der DQ-Dimension der Vollständigkeit minimiert wird.

6.1.3. Berechnung der Qualitätsverbesserung

Um die qualitätsverbessernde Wirkung des DQLS zu zeigen, wird neben der durchschnittlichen Verbesserung der Gesamtqualität θ^+, die Korrektheit α von Aggregationsergebnissen und die Integrität *int* von Ergebnismengen des Datenstromverbunds gemessen.

Das Load-Shedding führt ein Bernoulli-Sampling aus, so dass ein statistischer Fehler in die nachfolgende Datenverarbeitung eingefügt wird. Die Berechnung dieses Samplingfehlers wird in Abschnitt 5.3.1 beschrieben. Er wird in die Datenqualitätsdimension Konfidenz aufgenommen und durch die Datenverarbeitung propagiert. Die Korrektheit von Aggregationsergebnissen wird durch den systematischen und statistischen Fehler in den Dimensionen Genauigkeit a und Konfidenz ϵ beschrieben, deren Berechnung in Abschnitt 5.4.5 in den Theoremen 5.18 und 5.21 bis 5.24 für verschiedene Aggregationsfunktionen definiert ist.

Recall *rec* und Integrität *int* bemessen die Qualitätsverbesserung von MaxCompleteness. Der Recall beschreibt den Anteil der gefundenen Verbundpartner an der wahren Ergebnismenge. Er berechnet sich aus der Größe $|resultWithLS|$ der Ergebnismenge eines Datenstromverbundes, der unter Lastausgleich durchgeführt wurde, geteilt durch die Ergebnisgröße $|resultWithoutLS|$ des Verbunds ohne Ressourcenbeschränkungen: $rec = |resultWithLS| / |resultWithoutLS|$. Die Integrität *int* bezieht außerdem den Anteil fehlender Messwerte aufgrund von Sensorausfällen ein. Die durchschnittliche Vollständigkeit $c_w(k)$ aller Datenqualitätsfenster $w(k)$ mit $1 \leq k \leq \kappa$ wird mit dem Recall verknüpft.

$$int = rec \cdot \left(1 - \sum_{k=0}^{\kappa} \frac{1}{\omega_k} c_w(k)\right) \tag{6.2}$$

Die Evaluierung des DQLS in Abschnitt 7.4.1 zeigt, dass MaxDQ und MaxDQcompensate die Gesamtqualität bzw. Korrektheit von Aggregationsergebnissen im Vergleich zu existierenden Lösungen erheblich verbessern können. Der Algorithmus MaxCompleteness verbessert die durchschnittliche Vollständigkeit bezüglich fehlender Messwerte und die Integrität der Ergebnismengen des Datenstromverbundes.

6.1.4. Algorithmen des Lastausgleichs

Basierend auf dem oben geschilderten Load-Shedding-Konzept wurden zwei Algorithmen zur Verbesserung der Datenqualität von Aggregationsergebnissen sowie eine Strategie zur Maximierung der Verbundvollständigkeit bei begrenzten Speicherkapazitäten entwickelt.

MaxDQ - Verbesserung der Gesamtdatenqualität

Der Algorithmus MaxDQ setzt die in den vorherigen Abschnitten eingeführten Konzepte um. Die Gesamtqualität θ wird am Ende jedes Datenqualitätsfensters berechnet. Die Datenqualitätsschranke b wird durch Analyse der Datenqualitätsverteilung der Stromhistorie bestimmt. Schließlich wird das Bernoulli-Sampling auf Basis der Distanz zwischen Gesamtdatenqualität und DQ-Schranke für jedes Tupel des Fensters ausgeführt.

Wie bereits in Abschnitt 4.3.2 beschrieben, wird die Transformation zur Standardnormalverteilung genutzt, um unterschiedliche Datenqualitätsdimensionen zu normalisieren. Die Parameter der Normalverteilung $\mu(q_i)$ und $\sigma(q_i)$ werden iterativ über der strömenden Datenqualitätshistorie für jede Dimension q_i aufgezeichnet.

Die Zusammenfassung der Einzeldimensionen zur Gesamtdatenqualität birgt das Risiko, dass Datentupel mit hoher Qualität in einer Dimension aufgrund niedriger Qualität in den anderen Dimensionen gelöscht werden. Um dies zu vermeiden und den Fokus auf spezielle Dimensionen von besonderem Interesse lenken zu können, werden Einzeldimensionen mit Gewichten $weight_i (\sum weight_i = 1)$ versehen.

Definition 6.3 Die Gesamtdatenqualität θ wird als gewichtete Summe der ϑ normalisierten Datenqualitätsdimension q_i mit Mittelwert $\mu(q_i)$ und Standardabweichung $\sigma(q_i)$ definiert.

$$\theta = \sum_{i=1}^{\vartheta} weight_i \cdot \frac{q_i - \mu(q_i)}{\sigma(q_i)} \tag{6.3}$$

Tabelle 6.1 zeigt eine Beispielsituation des DQLS bei der Druckmessung aus Anwendungsszenario 1. Die zweite Spalte zeigt die Datenqualitätsinformationen des untersuchten Datenqualitätsfensters, wobei Vollständigkeit und Datenmenge zur Vereinheitlichung der DQ-Interpretation negiert wurden. Das untersuchte Datenqualitätsfenster weist eine gute Genauigkeit ($a_w < \mu(a)$), aber schlechte Vollständigkeit ($c_w < \mu(c)$) auf. Die dritte und vierte Spalte zeigen Mittelwert und Standardabweichung der Historie der DQ-Dimensionen, welche die Datenqualitätsverteilung jeder Dimension beschreiben. Mit ihrer Hilfe werden die Datenqualitätsinformationen q_i normalisiert (5. Spalte). Abschließend werden die normalisierten DQ-Werte mit dem einheitlichen Faktor von $weight_i = 0,25$ gewichtet. Die Gesamtdatenqualität des untersuchten DQ-Fensters beträgt nach Gleichung 6.3 $\theta = -0,075$.

DQ-Dimension	q_i	$\mu(q_i)$	$\sigma(q_i)$	norm. q_i	$weight_i$	gew., norm. q_i
Genauigkeit	1,2	1,5	0,1	-3	0,25	-0,75
Konfidenz	2,2	2,1	0,5	0,2	0,25	0,05
Vollständigkeit	-0,85	-0,9	0,02	2,5	0,25	0,625
Datenmenge	-1	-1	0,00	0	0,25	0

Tabelle 6.1.: Beispielszenario des datenqualitätsgesteuerten Lastausgleichs

Die DQ-Schranke b basiert auf der Load-Shedding-Rate r_{LS} und definiert den Wert der Gesamtdatenqualität, der von $r_{LS} \cdot 100\%$ aller Datenstromtupel unterschritten wird: $r_{LS} = 1/n \cdot |x_{\theta<b}|$. Diese Schranke wird vom Werteverlauf der Gesamtqualität der bisher verarbeiteten Datentupel abgeleitet, der mit Hilfe der inkrementell berechneten Parameter $\mu(\theta)$ und $\sigma(\theta)$ beschrieben wird (siehe Abbildung 6.3). Um die Zweideutigkeit der einfachen Dichtefunktion der Normalverteilung zu überwinden, wird die kumulative Dichteverteilung genutzt.

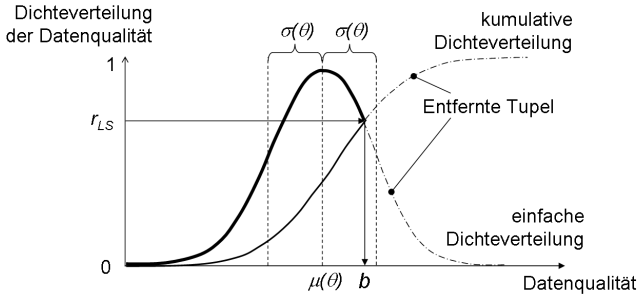


Abbildung 6.3.: Einfache und kumulative Dichteverteilung

Definition 6.4 Die Datenqualitätsschranke b ist als inverse kumulative Normalverteilung der Gesamtdatenqualität am Punkt der Wahrscheinlichkeit der Load-Shedding-Rate

r_{LS} definiert, wobei die zugrunde liegende Normalverteilung durch den Mittelwert $\mu(\theta)$ und die Standardabweichung $\sigma(\theta)$ bestimmt wird.

$$b = \Phi^{-1}(r_{LS}, \mu(\theta), \sigma(\theta)) \tag{6.4}$$

Um die inverse kumulative Dichteverteilung zu approximieren, kann der Algorithmus von Peter Acklam genutzt werden, der eine hocheffiziente und derzeit die genaueste Implementierung darstellt (relativer Fehler $< 1,15 \cdot 10^{-9}$) [Sha07].

Schließlich wird die spezifische Qualitätsdistanz $d_{\succ}$ genutzt, um die Wahrscheinlichkeit des Bernoulli-Samplings zu bestimmen. Die Distanzfunktion muss so gewählt werden, dass eine geringe Gesamtdatenqualität $\theta \ll b$ zu einer hohen Samplingwahrscheinlichkeit $p \to 1$ führt und umgekehrt ($p \to 0$ für $b \ll \theta$). Der Fall $b = \theta$ muss mit $p_{Bernoulli} = 0,5$ abgedeckt werden. Zwischen diesen drei Punkten soll die Bernoulli-Wahrscheinlichkeit stetig absinken.

Diese Eigenschaften werden von Sigmoidfunktionen [MMMR96], wie zum Beispiel dem Tangens Hyperbolicus, erfüllt (siehe Abbildung 6.4). Um den Wertebereich auf die Zieldomäne [0,1] abzubilden, muss er gewichtet und die Konstante 0,5 addiert werden.

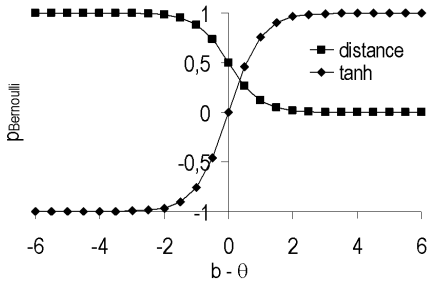


Abbildung 6.4.: Sigmoidfunktion des Tangens Hyperbolicus

Definition 6.5 Die Bernoulli-Wahrscheinlichkeit $p_{Bernoulli} = d_{\succ}(\theta, b)$ wird mittels der sigmoiden Qualitätsdistanzfunktion des Tangens Hyperbolicus berechnet.

$$d_{\succ}(\theta, b) = 0,5 \cdot tanh(b - \theta) + 0,5 \tag{6.5}$$

Zur Veranschaulichung wird das Beispielfenster aus Tabelle 6.1 aufgegriffen. Die Arbeitslast soll die Ressourcen um das dreifache überschreiten, so dass eine Load-Shedding-Rate von $r_{LS} = 0,33$ notwendig ist. Die Dichteverteilung der Gesamtdatenqualität ist durch den Mittelwert $\mu(\theta) = 0,175$ und die Standardabweichung $\sigma(\theta) = 0,158$ gekennzeichnet. Nach Gleichung 6.4 ergibt sich die Datenqualitätsschranke zu $b = 0,106$. Die Bernoulli-Wahrscheinlichkeit des Samplings jedes Datentupels des untersuchten Fensters beträgt nach Gleichung 6.5 $d_{\succ}(-0,075; 0,106) = 0,41$. Obwohl das Fenster über dem Qualitätsdurchschnitt liegt ($\theta < \mu(\theta)$), führt die hohe Überlast und geringe

Distanz zwischen Gesamtdatenqualität und DQ-Schranke zu einer geringen Samplingwahrscheinlichkeit.

Durch Anwendung des Bernoulli-Samplings wird ein Teil des Datenqualitätsfensters aus dem Strom entfernt. Die reduzierten Fenster müssen nachfolgend zu neuen Fenstern zusammengesetzt werden, um eine konsistente Fenstergröße zu erhalten.

MaxDQcompensate - Maximierung der Korrektheit

MaxDQcompensate gleicht den eingeführten Samplingfehler aus, indem der Lastausgleich auf Tupel mit schwacher Genauigkeit und/oder Konfidenz ausgerichtet wird. Dafür wird MaxDQ um eine vorgelagerte Analyse der erreichbaren Korrektheitsverbesserung erweitert. Danach wird die Load-Shedding-Entscheidung, die hinsichtlich des Fehlerausgleichs optimal wäre, mit der geforderten Load-Shedding-Rate r_{LS} verglichen.

Der erste Schritt empfiehlt das Entfernen oder Erhalten des aktuell untersuchten Datenstromtupels unter Beachtung der zwei Möglichkeiten der Korrektheitsverbesserung. Das Löschen eines Datentupels kann den hinzugefügten statistischen Fehler reduzieren, wenn die Standardabweichung $\sigma(x)$ der Messwerte durch x verringert wird (siehe Gleichung 5.23 auf Seite 97). Andererseits kann das Erhalten des Tupels die durchschnittliche Korrektheit durch hohe Tupelgenauigkeit und/oder -konfidenz verbessern. Der dominante Aspekt bestimmt die temporäre Entscheidung und Load-Shedding-Rate. Der zweite Schritt vergleicht diese Rate mit dem geforderten Lastausgleich mit Hilfe der Prozesskontrollfunktion $EWSA$, die in Abschnitt 4.3.2 vorgestellt wurde.

Im ersten Schritt (Algorithmus 6.1) wird zu Beginn das Entfernen des relevanten Tupels angenommen und die vorgeschlagene Rate r_{j+1} mit Hilfe der aktuellen Rate r_j und der gleichbleibenden Stichprobengröße $size_j$ aktualisiert (Zeile 1). Dann werden in den Zeilen 2-3 die Auswirkungen auf die Konfidenz $\Delta\epsilon$ berechnet. Die Änderung der durchschnittlichen Korrektheit $\Delta\alpha$ im Fall des Beibehaltens des Tupels im Datenstrom wird in den Zeilen 4-5 bestimmt. Eine negative Auswirkung $\Delta\epsilon$ verweist auf einen erhöhten statistischen Fehler, während eine negative Änderung der Korrektheit $\Delta\alpha$ eine Verminderung der Ergebniskorrektheit anzeigt. Führen beide Aktionen (Löschen, Erhalten) zu Auswirkungen negativer Natur, wird die kleinere Minderung als vorgeschlagene Load-Shedding-Aktion ausgewählt. Sind beide Effekte positiv, wird die Aktion ausgeführt, die eine stärkere Verbesserung zur Folge hat. Die temporäre Entscheidung wird in den Parametern $shed\_temp$ und vorgeschlagene Load-Shedding-Rate r_{j+1} ausgegeben.

Im Folgenden wird die Load-Shedding-Entscheidung $shed\_temp$ und -Rate r_{201} für ein Tupel des Beispielfensters in Tabelle 6.1 berechnet. Angenommen wird ein gemessene Druck $x(201) = 231bar$, eine bisherige Stichprobengröße von $size_{200} = 72$ Datentupeln sowie die aktuell erreichte Load-Shedding-Rate $r_{200} = 0,36$. Zuerst wird die neue Rate $r_{201} = 72 \cdot 1/0,36 + 1 = 0,358$ bestimmt. Auf Basis der Druckmessung $x(201)$ wird der bisherige statistische Messfehler $\epsilon_{200} = 2,06$ zu $\epsilon_{201} = 2,13$ aktualisiert. Damit beträgt die (negative) Auswirkung des Entfernens $\Delta\epsilon = -0,7$. Die Fenstergenauigkeit $a_w = 1,2$

Algorithmus 6.1 : Bestimmung der vorgeschlagenen Load-Shedding-Aktivität

Input : x current tuple,
a_w, ϵ_w current window accuracy & confidence
Output : r_{j+1} new load shedding rate,
$shed\_temp$ temporary load shedding suggestion

1 $r_{j+1} = \frac{size_j}{size_j \cdot 1/r_j + 1}$; * Akt. der Load-Shedding-Rate (Tupel gelöscht) * \
2 ϵ_{j+1} = updateIntroducedError(x, ϵ_j);
3 $\Delta\epsilon = \epsilon_j - \epsilon_{j+1}$; * Konfidenzänderung beim Löschen * \
4 α_{j+1} = updateCorrectnessImprovement($a_w, \epsilon_w, \alpha_j$);
5 $\Delta\alpha = \alpha_j - \alpha_{j+1}$; * Korrektheitsänderung bei Tupelerhalt * \
6 **if** $\Delta\epsilon > \Delta\alpha$ **then**
7 $shed\_temp$ = TRUE; * Tupel wird entfernt * \
8 **else**
9 $shed\_temp$ = FALSE; * Tupel bleibt erhalten * \
10 $r_{j+1} = \frac{size_j + 1}{size_j \cdot 1/r_j + 1}$; * Akt. der Load-Shedding-Rate (Tupelerhalt) * \

und -konfidenz $\epsilon_w = 2,2$ aktualisieren die Korrektheit $\alpha_{200} = 3,6$ zu $\alpha_{201} = 3,59$. Die Auswirkung beträgt $\Delta\alpha = 0,01$ und würde damit die Ergebniskorrektheit verbessern. Da $\Delta\epsilon < \Delta\alpha$, sollte das Tupel nicht aus dem Datenstrom entfernt werden ($shed\_temp$ = FALSE). Die Load-Shedding-Rate wird dem Tupelerhalt angepasst: $r_{201} = 0,361$.

Der zweite Schritt von MaxDQcompensate (Algorithmus 6.2) prüft die temporäre Entscheidung gegen die geforderte Load-Shedding-Rate r_{LS}. Zeile 1 aktualisiert die Prozesskontrollfunktion $EWSA$ der angewendeten Load-Shedding-Raten mit der vorgeschlagenen Rate r_{j+1}. Um die notwendige Lastreduzierung zu garantieren, wird die vorgeschlagene Load-Shedding-Rate mit einem Kontrollintervall verglichen, ähnlich der Kontrolle des Datenqualitätsfehlers in Gleichung 4.9. Zeile 2 berechnet dieses Kontrollintervall $[lowerRateBound, upperRateBound]$ wie folgt.

$$upperRateBound = r_{LS}, \tag{6.6}$$
$$lowerRateBound = r_{LS} - \rho \cdot \sigma(EWSA) \tag{6.7}$$

Wenn die vorgeschlagene Rate r_{j+1} innerhalb dieser Grenzen liegt, kann die ermittelte optimale Load-Shedding-Aktivität ausgeführt werden (Zeilen 7-8). Die durchschnittliche Korrektheit des approximierten Aggregationsergebnisses wird entweder durch Reduzierung des Samplingfehlers oder durch Erhöhung der Eingangskorrektheit verbessert.

Wenn die obere Intervallgrenze $upperRateBound$ überschritten wird, kann die geforderte Lastreduktion durch die vorgeschlagene Aktivität nicht erreicht werden (Zeilen 3-4). Obwohl Algorithmus 6.1 den Erhalt des analysierten Tupels vorgeschlagen haben

Algorithmus 6.2 : Kontrolle der vorgeschlagenen Load-Shedding-Rate

Input : r_{j+1} suggested load shedding rate,
$shed\_temp$ temporary load shedding activity,
Q data quality information
Output : $shed$ load shedding decision

```
1  EWSA_{j+1} = β · r_{j+1} + (1 − β) · EWSA_j;   \* Akt. der Prozesskontrollfunktion * \
2  updateControlIntervallBounds(EWSA_{j+1});      \* Akt. der Kontrollintervalls * \
3  if r_{j+1} > upperRateBound then
4      shed = FALSE;                              \* Tupel muss entfernt werden * \
5  else if r_{j+1} < lowerRateBound then
6      shed = shed_temp ∨ MaxDQ(Q, r_LS);         \* Ressourcen nicht ausgelastet * \
7  else
8      shed = shed_temp;                          \* vorgeschlagene Aktion wird ausgeführt * \
9  if shed then
10     size_{j+1} = size_j + 1;
```

könnte, muss es aus dem Datenstrom gelöscht werden. Die gewünschte Verbesserung der Korrektheit konnte nicht erreicht werden.

Im Gegensatz dazu bleibt verfügbare Kapazität ungenutzt, wenn die vorgeschlagene Load-Shedding-Rate die untere Grenze des Kontrollintervalls $lowerRateBound$ unterschreitet (Zeilen 5-6). Obwohl ein Entfernen des Tupels bezüglich der Korrektheit optimal wäre ($shed\_temp$ = FALSE), kann der Datenstrom vom Erhalt dieses Tupels profitieren, wenn es gute Qualitäten in den anderen Dimensionen aufweist. Der Algorithmus MaxDQ zur Verbesserung der Gesamtqualität wird angewendet. Die Priorität der Korrektheit wird reduziert, um freie Kapazitäten des Datenstromsystems voll auszunutzen. Das erlaubt eine bessere Einbeziehung weiterer Qualitätsaspekte.

Die vorgeschlagene Load-Shedding-Rate $r_{201} = 0,361$ wird im Folgenden gegen die benötigte Rate von $r_{LS} = 0,33$ verglichen. Hierfür werden die Prozesskontrollfunktion $EWSA_{200} = 0,347$ sowie $\beta = 0,05$ angenommen. Zuerst wird die Prozesskontrollfunktion $EWSA_{201} = 0,348$ bestimmt, die das Kontrollintervall $[0,33; 0,351]$ definiert. Damit zeigt sich, dass die vorgeschlagene Load-Shedding-Rate außerhalb des Kontrollintervalls liegt. Obwohl der Tupelerhalt die Korrektheit des Anfrageergebnisses erhöhen würde, muss das Tupel gelöscht werden, um den geforderten Lastausgleich zu gewährleisten.

MaxCompleteness - Maximierung des Datenstromverbundes

Wird der Datenstromverbund auf ungeordneten Verbundattributen ausgeführt, müssen Datenstromfenster zur Suche von Verbundpartnern zwischengespeichert werden. Die endlichen Hardware-Ressourcen erlauben jedoch nicht die Speicherung des vollständigen Datenstroms, so dass potentielle Partner gelöscht werden, bevor sie zum Verbundergebnis beitragen können. Reale Anwendungsumgebungen fügen weitere

Ursachen für unvollständige Verarbeitungsergebnisse hinzu. Sensoren können ausfallen oder Datenpakete bei unsicherer Übertragung verloren gehen.

MaxCompleteness ist der erste Load-Shedding-Algorithmus, der Informationen über derartige Unvollständigkeiten in die Maximierung der Integrität des Verbundergebnisses einbezieht. Die Ordnung in Definition 6.1 wird angewendet, um während der Verbundausführung diejenigen Tupel im gespeicherten Datenstromfenster zu erhalten, die die höchste Vollständigkeit erreichen.

MaxCompleteness ist in Algorithmus 6.3 zusammengefasst. Wenn Datenqualitätsinformationen am Ende des DQ-Fensters w eintreffen, wird die Vollständigkeit mit der temporär zwischengespeicherten Datentupelmenge X verglichen. Tupel mit geringerer Vollständigkeit werden durch die neuen Tupel ersetzt, bis entweder alle Tupel aus w in die gespeicherte Menge X aufgenommen wurden oder der Anteil geringerer Vollständigkeit ausgetauscht wurde.

Algorithmus 6.3 : MaxCompleteness

```
   Input : w current data quality window, X set of stored tuples
1  forall x ∈ w do
2      if ∃y ∈ X|c(x) < c(y) then
3          exchange(x, y);
4  end
```

Die Datenqualitätsschranke b wird durch die niedrigste Vollständigkeit in X bestimmt. Die Bernoulli-Wahrscheinlichkeit ist $p = 1$, wenn die Schranke überschritten wurde, $p = 0$ andernfalls.

$$p_{Bernoulli}(x) = \begin{cases} 0 & c(x) < c(b), \\ 1 & else \end{cases} \tag{6.8}$$

Die Vollständigkeit des untersuchten Beispielfensters in Tabelle 6.1 beträgt $c_w = 0,85$. Enthält die zwischengespeicherte Menge X Datentupel, die eine kleinere Vollständigkeit aufweisen, werden diese ausgetauscht.

MaxCompleteness nutzt Datenqualitätsinformationen, um Unvollständigkeiten im Anfrageergebnis zu reduzieren, die durch Sensorausfälle entstehen. Die Tupelausprägungen selbst werden nicht untersucht, so dass ein schlechter Recall erreicht wird, da Tupel entfernt werden, bevor Verbundpartner gefunden wurden. Deshalb wird MaxCompleteness mit den Load-Shedding-Strategien von Kang [KNV02] und Srivastava [SW04] (siehe Abschnitt 3.3.1) verknüpft. Die hybriden Algorithmen verbinden die Vorteile beider Ansätze. Die Load-Shedding-Wahrscheinlichkeit p_{hybrid} vereint die Entscheidung hinsichtlich der genutzten Datenstromstatistik (Tupelalter bzw. -frequenz) und der Vollständigkeit der Datentupel.

$$p_{hybrid}(x) \quad = \quad p_{Statistics}(x) \cdot p_{Bernoulli}(x) \tag{6.9}$$

Durch die Integration der Aussagen über die Werteverteilung in der Stromstatistik kann der Recall der Verbundergebnismengen verbessert werden. Die Validierung in Abschnitt 7.4.1 zeigt, dass damit die Nachteile des strengen MaxCompleteness überwunden werden können.

6.2. Qualitätsgesteuerte Optimierung der Datenstromverarbeitung

Die Untersuchungen der Eigenschaften von Sensormessdaten und der Datenverarbeitung im Datenstromsystem ergaben zwei Ursachen mangelhafter Datenqualität. Einerseits können Sensoren nur eingeschränkte Messdatenqualität bieten. Der Sensoraustausch ist jedoch in der Regel mit hohen Kosten verbunden und soll daher nicht verfolgt werden. Andererseits kann die Datenqualität durch spezifische Verarbeitungsoperatoren in Mitleidenschaft gezogen werden. Dieser Abschnitt widmet sich der datenqualitätsgesteuerten Datenstromverarbeitung, um Datenqualitätsprobleme hervorgerufen durch die Datenverarbeitung zu reduzieren. Um die Datenqualität von Verarbeitungsergebnissen zu verbessern und Nutzeranforderungen zu integrieren, muss die Konfiguration der Verarbeitungsoperatoren optimiert werden.

Abschnitt 6.2.1 stellt die zu verfolgenden Zielfunktionen sowie Konfigurationsmöglichkeiten der Operatoren vor. Darauf aufbauend wird das Optimierungsproblem definiert. Abschnitt 6.2.2 präsentiert das Architekturschema der qualitätsgesteuerten Optimierung und beschreibt die qualitätsgesteuerte Evolutionsstrategie als Beispielheuristik zur approximativen Problemlösung.

Motivation

Abbildung 6.5 zeigt einen Auszug des Datenstroms des dritten Anwendungsszenarios der Kontaktlinsenkontrolle (siehe Abschnitt 2.1). Zur Kontrolle der Qualität der produzierten Linsen werden Dicke und Axialverschiebung jeder Linse gemessen. Die Linsenqualität wird als gewichtete Summe dieser Messwerte berechnet und mit dem Qualitätsschwellwert $threshold = 0,5$ verglichen.

Die Datenqualitätsdimensionen Genauigkeit und Konfidenz sind als Fehlerbalken zur Markierung des unsicheren Entscheidungsbereichs angetragen. Die Produktionsqualität der Linse zum Zeitpunkt $t = 12$ ist mit $0,45$ angegeben, so dass der Schwellwert eingehalten wird. Die DQ-Informationen bestimmen allerdings den unsicheren Bereich $[0,37; 0,53]$, so dass die wahre Produktionsqualität der Linse den Schwellwert überschreiten könnte. Linse 12 muss somit aus Sicherheitsgründen aus der weiteren Produktion

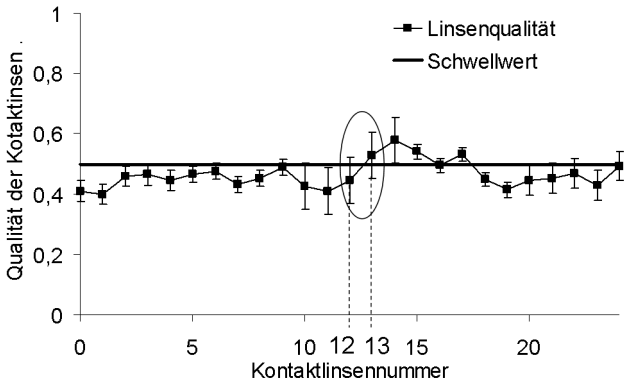


Abbildung 6.5.: Datenqualitätsprobleme bei der Kontaktlinsenproduktion

ausgeschlossen werden. Außerdem könnte die Linse 13 eine korrekt produzierte Linse darstellen, die fälschlicherweise als schlecht kategorisiert wurde.

Ziel der Datenqualitätsverbesserung ist es nun, den unsicheren Messfehlerbereich so schmal wie möglich zu gestalten, um die beschriebenen Fehleinschätzungen zu reduzieren. Sofern möglich, sollen neben dem im Beispiel beschriebenen numerischen Messfehler auch alle weiteren zur Verfügung stehenden Qualitätsdimensionen optimiert werden.

Um die Datenqualität der Ergebnisse der Datenstromverarbeitung zu maximieren, muss der numerische Messfehler in den DQ-Dimensionen Genauigkeit a und Konfidenz ϵ minimiert, die Vollständigkeit c, Datenmenge d und Aktualität u hingegen maximiert werden. Zudem werden drei weitere Optimierungsziele hinzugefügt und im folgenden Abschnitt detailliert beschrieben. Die Ressourcenbeschränkungen im Umfeld der Datenstromverarbeitung erfordern die Minimierung des Datenstromvolumens V. Um Detailinformationen bezüglich der Sensormesswerte und Datenqualitätsinformationen zur Verfügung zu stellen, wird die Granularität G der Datenstromtupel maximiert, der Datenqualitätsfehler δq dagegen minimiert.

6.2.1. Definition des Optimierungsproblems

Um Kandidaten zur Datenqualitätsverbesserung zu identifizieren, wurden alle in Kapitel 5 präsentierten Operatoren hinsichtlich ihrer Parameter untersucht und in Tabelle 6.2 in Relation zu den oben genannten Optimierungszielen gestellt. Neben der Konfiguration der Datenstromoperatoren ist die Fenstergröße ω ein wichtiger Optimierungsparameter. Die Erhöhung des jeweiligen Konfigurationsparameters führt zu einer Erhöhung (+) bzw. Verringerung (-) der aufgestellten Optimierungsziele.

Operator	Parameter	a	ϵ	c	d	u	V	G	δq
Projektion	—								
Selektion	—								
Verbund	—								
Aggregation	Gruppengröße l				+		-	-	+
Sampling	Rate r_{sa}		-	-		+	+		
Interpolation	Rate r_{in}		-	-		+	+		
Frequenzanalyse	Gruppengröße l				+	-	-	-	+
Filter	Gruppengröße l								+
num. Algebra	—								
Schwellwertvgl.	—								
	Fenstergröße ω					-	-		+

Tabelle 6.2.: Konfigurationsmöglichkeiten zur Datenqualitätsverbesserung

Definition der Zielfunktionen

Dieser Abschnitt definiert die Zielfunktion für jedes zu verfolgende Teilziel der Datenqualitätsverbesserung. Da die Genauigkeit a systematische Messfehler aufgrund unpräziser Sensordatenquellen beschreibt, kann sie nicht durch Konfigurationen der Verarbeitungsoperatoren verbessert werden und wird aus der Liste der Optimierungsziele entfernt.

Die verbleibenden Optimierungsziele besitzen unterschiedliche Wertebereiche. Zum Beispiel nimmt die Vollständigkeit Werte zwischen 0 und 1 an, während die Konfidenz unbegrenzt ist ($0 \leq \epsilon \leq \infty$). Um den quantitativen Vergleich der Optimierungsziele zu ermöglichen, werden alle Zielfunktionen auf den Wertebereich $[0;1]$ normalisiert.

Konfidenz Um die durchschnittliche Datenstromkonfidenz zu maximieren, muss der statistische Fehler jedes Attributs minimiert werden. Zur Bewertung wird die durchschnittliche Konfidenz $\epsilon_w(k)$ aller κ_i Datenqualitätsfenster aller Attribute A_i mit $1 \leq i \leq n$ des Datenstroms herangezogen. Die Zielfunktion wird mit Hilfe der Division durch die maximale Konfidenz des untersuchten Datenstroms ϵ_{max} wie folgt normalisiert.

$$f_\epsilon \quad : \quad min \quad \frac{1}{n \cdot \epsilon_{max}} \sum_{i=1}^{n} \frac{1}{\kappa_i} \sum_{k=1}^{\kappa_i} \epsilon_w(k) \tag{6.10}$$

Vollständigkeit Um die Optimierungsrichtung zu harmonisieren, wird das Ziel der maximalen Vollständigkeit zum Minimierungsproblem f_c transformiert, das die Rate der interpolierten Datenwerte minimiert. Hier ist keine Normalisierung notwen-

dig, da der Wertebereich $[0;1]$ bereits durch die Definition der (Un-)vollständigkeit gegeben ist.

$$f_c \quad : \quad min \quad \frac{1}{n}\sum_{i=1}^{n}\frac{1}{\kappa_i}\sum_{k=1}^{\kappa_i}(1 - c_w(k)) \tag{6.11}$$

Datenmenge Die Datenmenge gibt die Rohdatenbasis eines berechneten Datenwertes an. Je größer die Datenmenge, umso verlässlicher ist das Datentupel. Um das Ziel der größtmöglichen Datenmenge in ein Minimierungsproblem umzuwandeln, wird die Differenz zur maximalen Datenmenge $d = m$ gebildet, die alle Datensätze eines Datenstroms zusammenfasst. Zur Normalisierung wird wiederum durch die maximale Datenmenge m dividiert.

$$f_d \quad : \quad min \quad \frac{1}{n \cdot m}\sum_{i=1}^{n}\frac{1}{\kappa_i}\sum_{k=1}^{\kappa_i}(m - d_w(k)) \tag{6.12}$$

Aktualität Um die Aktualität zu maximieren, wird das durchschnittliche Alter als Differenz des aktuellen Systemzeitpunktes *clock* und des Messzeitstempels $t(j)$ aller Datentupel minimiert und mit Hilfe des maximalen Alters u_{max} normalisiert.

$$f_u : \quad min \quad \frac{1}{m \cdot u_{max}}\sum_{j=1}^{m} u(j) \tag{6.13}$$

$$= \quad \frac{1}{clock - t_{min}} \cdot \left[clock - \frac{1}{m}\sum_{j=1}^{m} t(j)\right] \tag{6.14}$$

Datenstromvolumen Das Volumen V eines Datenstroms gibt die Anzahl der enthaltenen Messwerte als Produkt aus Datenstromlänge m und Attributzahl n an. Außerdem müssen die transferierten Datenqualitätsinformationen und Qualitätskontrolltupel in das Datenvolumen eingerechnet werden. Dazu wird für jedes Attribut $A_i(1 \leq i \leq n)$ die Anzahl der propagierten Dimensionen ϑ, die durchschnittliche Fenstergröße $\bar{\omega}_i$ und die feste Kontrollrate r_c benötigt. Der erste Term der rechten Seite bestimmt das Datenvolumen des traditionellen Datenstrommodells. Der zweite Term beschreibt das Volumen zur Verwaltung der fensterbasierten Datenqualitätsinformationen. Schließlich definiert der dritte Term das benötigte Datenvolumen der DQ-Kontrolle zur Adaption der Datenqualitätsfenstergröße und Minimierung des Datenqualitätsfehlers wie in Abschnitt 4.3.2 beschrieben.

$$V \quad = \quad m \cdot (n+1) + \sum_{i=1}^{n}\frac{m}{\bar{\omega}_i} \cdot \vartheta + m \cdot r_c \tag{6.15}$$

Um das Datenstromvolumen auf den Wertebereich $[0, 1]$ abzubilden, muss analog zu oben genannten DQ-Dimensionen durch das maximal mögliche Volumen dividiert werden, welches sich aus der maximal übertragbaren Datenstromrate r_{max} (z.B. $r_{max} = 1/ms$) im Verhältnis zur aktuellen Datenrate r, der Tupelanzahl m und der minimalen Datenqualitätsfenstergröße $\omega = 1$ in Gleichung 6.16 ergibt. In diesem Fall ist keine Fenstergrößenadaption notwendig, so dass gilt $r_c = 0$.

$$V_{max} = \frac{r_{max}}{r} m \cdot (n+1) + \frac{r_{max}}{r} m \cdot \sum_{i=1}^{n} \vartheta, \tag{6.16}$$

$$f_V : min \quad \frac{V}{V_{max}} \tag{6.17}$$

Granularität Die Granularität G beschreibt den Zeitraum $[t_e - t_b]$, der im Durchschnitt von einem Datenstromtupel abgedeckt wird. Zum Beispiel beträgt die Granularität eines Aggregationsergebnisses der Gruppengröße $l = 20$ und Datenstromrate $r = 1/s$ den Wert $G = 20s$. Rohdaten, die einen einzelnen Messzeitpunkt beschreiben haben eine Granularität von $G = 0$, da gilt $t_e = t_b$. Um die Granularität zu maximieren, muss der durchschnittliche Zeitrahmen aller Datenstromtupel minimiert und mit dem Maximum $G_{max} = m/r_{max}$ normalisiert werden.

$$f_G : min \quad \frac{1}{G_{max}} \sum_{j=1}^{m} t_e(j) - t_b(j) \tag{6.18}$$

Datenqualitätsfehler Der Datenqualitätsfehler gibt die Distanz zwischen tupel- und fensterbasierter Datenqualitätsinformation an. In Abschnitt 4.3.1 wurde gezeigt, dass diese Abweichung mit der Größe der Datenqualitätsfenster wächst. Die Optimierung der Datenverarbeitung verfolgt das Ziel, den durchschnittlichen Datenqualitätsfehlers aller Attribute $A_i (1 \leq i \leq n)$ (1), Datenqualitätsfenster $w_i (1 \leq k \leq \kappa_i)$ (2) und -dimensionen $q \in Q_i$ (3) zu minimieren. Entsprechend Abschnitt 4.3.2 erfüllt der Datenqualitätsfehler den Zieldatenbereich $[0, 1]$, so dass keine Normalisierung notwendig ist.

$$f_{\delta q} : min \quad \underbrace{\frac{1}{n} \sum_{i=1}^{n}}_{(1)} \underbrace{\frac{1}{\kappa_i} \sum_{k=1}^{\kappa_i}}_{(2)} \underbrace{\frac{1}{\vartheta} \sum_{q \in Q_i} \delta q(w_i)}_{(3)} \tag{6.19}$$

Analyse der Konfigurationsparameter

Die Analyse des Operatoren-Repository des DQMx identifizierte die Operatoren Aggregation, Sampling, Interpolation, Frequenzanalyse sowie Frequenzfilter als Kandidaten

der qualitätsgesteuerten Optimierung. Im Folgenden wird zuerst die Dimension des Suchraumes der Optimierung bestimmt. Anschließend werden die möglichen Konfigurationen diskutiert.

Abbildung 6.6 zeigt die Datenverarbeitung der Kontaktlinsenkontrolle. Vier Messungen der Stärke des Kontaktlinsenrandes th_e werden zusammengefasst (1) und zur gewichteten Dicke der Linsenmitte th_c addiert (2). Alle Sensorströme werden mittels Sampling abgetastet oder interpoliert, um die Datenstromraten anzupassen und das Gesamtdatenvolumen zu reduzieren. Die Axialverschiebung ax wird aggregiert, um mögliche Fehler zu summieren (3). Um die Gesamtgüte der Produktion zu überwachen, werden alle Messungen zu einem Qualitätsindikator qi zusammengefasst (4). Überschreitet dieser Indikator den Schwellwert $0,5$, muss die Kontaktlinse als Ausschuss deklariert und aus der Produktion entfernt werden (5). Außerdem wird der Drift des durchschnittlichen Qualitätsindikator in gleitenden Datenstromfenstern aggregiert, um die Wartungsplanung der Produktionslinie zu unterstützen (6).

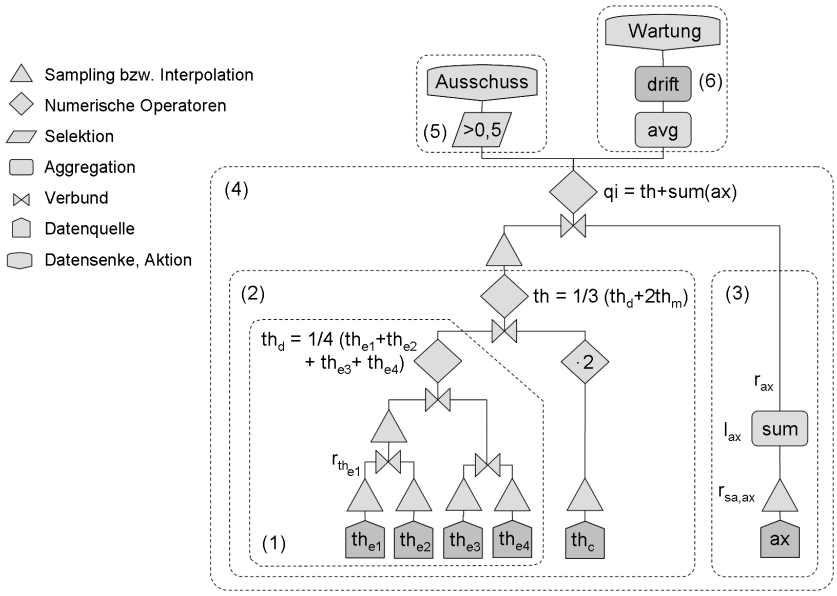


Abbildung 6.6.: Verarbeitungsgraph der Kontaktlinsenkontrolle

Die maximale Tiefe $depth$ des Verarbeitungsgraphen wird durch die Anzahl der eingesetzten Sensoren S definiert: $depth = log_2(S)$. Um Verbundpartner zu garantieren, müssen die Datenstromraten einer Verarbeitungsebene in konstanter Relation verbleiben. Zum Beispiel führt die Verdopplung der Rate $r_{th_{e1}} = 0,1$ zur Verdopplung der Samplingrate $r_{th_c} = 0,6$, so dass dieser Samplingoperator durch die Interpolation mit $r'_{th_c} = 1,2$ ersetzt werden muss.

Neben der Konfiguration der Sampling- bzw. Interpolationsraten können die Gruppengrößen der ausgeführten Aggregationen agg, Frequenzanalysen $frequ$ und Frequenz-

filter fil verändert werden. Sie unterliegen keinen Beschränkungen. Es können beliebig viele Operatoren dieser Klassen im Verarbeitungspfad enthalten sein. Schließlich kann die Fenstergröße zur Initialisierung der Datenqualitätsfenster für jeden Sensor einzeln konfiguriert werden. Der Suchraum des Optimierungsproblems hat demnach $dim = depth + |agg| + |frequ| + |fil| + S$ Dimensionen.

Samplingrate

Das Sampling reduziert das Datenstromvolumen, indem Datenwerte zufällig aus dem Datenstrom entfernt werden. Kleine Samplingraten entfernen einen großen Datenstromanteil, was zu einer Reduzierung des Datenvolumens, aber einer signifikanten Steigerung des statistischen Fehlers führt. Die Optimierungsziele der minimalen Konfidenz ϵ und des minimalen Datenstromvolumens V widersprechen einander. Ein minimales Datenstromvolumen führt außerdem zur Minimalisierung der Aktualität u, da kleine Tupelmengen kurze Verarbeitungszeiten bedeuten. Die minimale Konfidenz steht somit ebenfalls der Aktualität entgegen.

Interpolationsrate

Während der Interpolation werden Datentupel in den Datenstrom eingefügt. Beispielsweise wird das Datenvolumen durch die lineare Interpolation der Rate $r_{in} = 2$ verdoppelt. Je höher die Interpolationsrate, umso größer ist das nachfolgende Datenstromvolumen. Durch das Einfügen neuer Datenwerte wird außerdem das Verhältnis zwischen originalen und interpolierten Datenstromtupeln verändert. Je höher die Interpolationsrate, umso geringer ist die resultierende durchschnittliche Vollständigkeit des Datenstroms. Somit haben hohe Interpolationsraten einen negativen Einfluss sowohl auf das Datenstromvolumen V als auch auf die Vollständigkeit c.

Wie oben erläutert, müssen die Datenraten einer Verarbeitungsebene konstant bleiben. Wird die Interpolationsrate eines Datenstroms (z.B. th_c) erhöht, können die Samplingraten in dieser Ebene (th_{e1-e4}) ebenfalls erhöht werden. Die Interpolation verbessert damit indirekt den statistischen Fehler der DQ-Dimension Konfidenz. Die Optimierungsziele der maximalen Vollständigkeit c und des minimalen Datenstromvolumens V, die durch niedrige Sampling- bzw. Interpolationsraten $r_{sa/in}$ erreicht werden, stehen in Konflikt mit der Minimierung der Konfidenz ϵ erzielt durch hohe Raten $r_{sa/in}$.

Gruppengröße

In erster Linie stellt die Gruppengröße l einen Parameter der Anfragedefinition dar. Soll zum Beispiel der Geschäftsumsatz berechnet werden, ist es entscheidend, ob die Verkaufswerte eines Tages oder eines Monats summiert werden. Bei der Verarbeitung von Sensormessdaten können allerdings Situationen auftreten, in denen die Gruppengröße nicht strikt vorgegeben ist. Erstens können Schwankungen der Datenstromrate die Gruppengrößen beeinflussen. Zweitens kann dem Nutzer die optimale Gruppengröße z.B. der Frequenzanalyse unbekannt sein.

Aggregation und Frequenzanalyse fassen Datentupel zu einer Synopse zusammen und reduzieren so das Datenstromvolumen. Je größer die Aggregations- bzw. Analysegruppe,

umso stärker wird das Volumen verkleinert und umso größer ist die resultierende Datenmenge. Jedoch hat das Zusammenfassen mehrerer Datenstromtupel einen negativen Einfluss auf die Granularität, da die Ergebnistupel einen größeren Zeitrahmen beschreiben. Bei der Konfiguration der Gruppengröße treten somit Widersprüche zwischen den Optimierungszielen der maximalen Granularität G und des minimalen Datenstromvolumens V bzw. der maximalen Datenmenge d auf.

Der Zielkonflikt kann mit Hilfe einer Konfiguration der Datenstromrate durch Sampling und Interpolation gelöst werden. Wird die Datenrate im gleichen Maße wie die Gruppengröße erhöht, wird eine konstante Datenmenge bzw. konstantes Datenstromvolumen erreicht. Jedoch entstehen in diesem Fall Konflikte zu minimaler Konfidenz und maximaler Vollständigkeit, wie bereits oben erläutert.

Der Frequenzfilter erzeugt die Synopse im Frequenzbereich, um Frequenzen entsprechend dem Filterkriterium zu selektieren und anschließend das Signal im Zeitbereich wieder aufzubauen. Da die ursprüngliche Tupelzahl wieder hergestellt wird, werden weder Datenmenge noch Datenstromvolumen beeinflusst.

Alle Datenqualitätsdimensionen werden im Zuge der Aggregation, Frequenzanalyse und Frequenzfilterung über den verarbeiteten Eingangsdatentupeln gemittelt. Ähnlich der Durchschnittsberechnung der Fensterdatenqualität werden interessante Ausreißer der Datenqualitätswerte ausgeglichen, so dass der Datenqualitätsfehler steigt. Je größer die Gruppe der Durchschnittsbildung, umso größer ist der resultierende Datenqualitätsfehler. Die Optimierungsziele des minimalen Datenvolumens V bzw. der maximalen Datenmenge d, erreicht durch große Gruppen, stehen in Konflikt zum minimalen Datenqualitätsfehler δq.

Fenstergröße

Bereits in Abschnitt 4.3 wurde die dynamische Adaption der Größe der Datenqualitätsfenster betrachtet. Minimale Datenqualitätsfehler δq in kleinen DQ-Fenstern stehen dem zusätzlich benötigten Datenvolumen V gegenüber.

Die Fenstergröße beeinflusst außerdem die Aktualität der Datenstromtupel. Je kleiner die Datenqualitätsfenster definiert sind, umso mehr DQ-Informationen müssen verarbeitet werden. Große Fenster beschleunigen damit den Prozess der Daten- und Datenqualitätsverarbeitung. Das Optimierungsziel der maximalen Aktualität kann nur auf Kosten eines erhöhten Datenqualitätsfehlers erreicht werden.

Tabelle 6.3 fasst alle Konflikte der Optimierungsziele zusammen. Die Einträge der Zellen bestimmen den jeweiligen Konfigurationsparameter: Gruppengröße l, Sampling- bzw. Interpolationsrate r_{sa}, Fenstergröße ω. Die dichte Besetzung der Tabelle zeigt die Komplexität des Optimierungsproblems.

	ϵ	c	d	u	V	G	δq
ϵ	-	r_{sa}		r_{sa}	r_{sa}	r_{sa}	
c	r_{sa}	-	r_{sa}	r_{sa}	r_{sa}	r_{sa}	
d		r_{sa}	-			r_{sa}, l	l
u	r_{sa}	r_{sa}		-		l	ω, l
V	r_{sa}	r_{sa}			-	l	ω, l
G	r_{sa}	r_{sa}	r_{sa}, l	l	l	-	
δq			l	ω, l	ω, l		-

Tabelle 6.3.: Konflikte zwischen den Optimierungszielen

Klassifizierung des Optimierungsproblems

Im Operations Research werden mathematische und formale Methoden und Algorithmen zur optimalen Lösung komplexer Problemstellungen kombiniert. Optimierungsprobleme werden entsprechend ihres Wertebereichs, der Zielfunktion und Nebenbedingungen klassifiziert. Im folgenden Abschnitt wird das Optimierungsproblem der qualitätsgesteuerten Datenstromverarbeitung in dieses Klassifikationsschema eingeordnet.

Die Ziele der Datenqualitätsverbesserung stellen ein *multikriterielles* Optimierungsproblem dar, für dessen Lösung verschiedene Strategien zur Verfügung stehen. Die Pareto-Optimierung erlaubt das Finden einer Menge von optimalen Lösungskompromissen, die alle weiteren Lösungen dominieren. Eine Problemlösung ist Pareto-dominant, wenn ein Teilziel nur unter Verschlechterung eines anderen verbessert werden kann.

Definition 6.6 Die multikriterielle Zielfunktion $f_{multi} : \mathbb{R}^{dim} \mapsto \mathbb{R}^7$ bestimmt die Güte einer Problemlösung mittels der Pareto-Dominanz der Teilziele $f_i(i \in \{\epsilon, c, d, u, V, G, \delta q\})$.

Neben der Pareto-Optimierung der Teilziele können diese auch in einer Zielfunktion zusammengefasst werden, die zur Lösung des Gesamtproblems maximiert oder minimiert werden muss. Gewichtungen bestimmen die Reihenfolge bzw. den Grad der Optimierung der einzelnen Teilziele. Die nutzerdefinierte Gewichtung kann jedoch den Suchraum ungewollt einschränken und damit zu minderwertigen Optimierungsergebnissen führen. Es sind meist mehrere Optimierungsdurchläufe notwendig, um eine optimale Gewichtung zu bestimmen.

Definition 6.7 Die monokriterielle Zielfunktion $f_{single} : \mathbb{R}^{dim} \mapsto \mathbb{R}$ berechnet die gewichtete Summe aller Teilziele.

$$f_{single} = \sum_{i \in \{\epsilon, c, d, u, V, G, \delta q\}} c_i \cdot f_i \tag{6.20}$$

Die Datenqualitätsdimensionen Vollständigkeit, Datenmenge und Aktualität werden mit Hilfe linearer Berechnungsfunktionen bestimmt, so dass die Zielfunktionen f_c, f_d

und f_u *lineare* Optimierungsprobleme definieren. Die Ziele f_ϵ, f_V und $f_{\delta q}$ stellen dagegen *nicht-lineare* Probleme dar. Zum Beispiel wird die Konfidenz mit Hilfe der quadratischen Fehlerfortpflanzung bestimmt. Bis auf f_V und f_d, die nur diskrete Fenster- bzw. Gruppengrößen erlauben, besitzen die gegebenen Zielfunktionen einen *kontinuierlichen* Suchraum. Außerdem unterliegen die Sampling- und Interpolationsraten sowie die Gruppen- oder Fenstergrößen einschränkenden Nebenbedingungen wie zu Beginn des Abschnittes 6.2.1 beschrieben. Bei multikriteriellen Optimierungsproblemen definiert das komplexeste Teilziel die Komplexität des Gesamtproblems.

Definition 6.8 Die datenqualitätsgesteuerte Optimierung der Datenstromverarbeitung stellt ein multikriterielles, nicht-lineares, kontinuierliches, beschränktes Optimierungsproblem dar.

Es existiert kein deterministischer Algorithmus, um Optimierungsprobleme dieser Komplexität in akzeptabler Zeitspanne zu lösen [SM07]. Die qualitätsgesteuerte Optimierung der Datenstromverarbeitung soll jedoch während der Laufzeit erfolgen, ohne den Datenfluss zu unterbrechen. Gute Problemlösungen müssen schnell gefunden werden. Andererseits werden keine global optimalen Lösungen benötigt. Vielmehr sollen nutzerdefinierte Datenqualitätslevel erfüllt werden. Heuristische Algorithmen liefern schnelle Lösungen, indem Optimierungsergebnisse angenähert werden und bieten damit eine gute Umsetzung der qualitätsgesteuerten Optimierung der Datenstromverarbeitung.

6.2.2. Lösung des Optimierungsproblems

Um das definierte Optimierungsproblem zu lösen, wird im Folgenden ein generisches Rahmenwerk zur Verbesserung der Datenqualität der Datenstromverarbeitung vorgestellt und die qualitätsgesteuerte Evolutionsstrategie als beispielhafte Spezifikation des verwendeten Optimierungsalgorithmus beschrieben.

Rahmenwerk zur Optimierung der Datenstromverarbeitung

Der Prozess der qualitätsgesteuerten Optimierung der Datenstromverarbeitung muss kontinuierlich ausgeführt werden, um das Datenstromsystem während der Laufzeit an dynamische Änderungen der Nutzeranforderungen oder Datenstromeigenschaften anzupassen. Sobald eine optimale Konfiguration der Verarbeitungsoperatoren gefunden wurde, muss sie im Online-Tuning mit Hilfe des laufenden Datenstroms überprüft werden. Das erlaubt die nahtlose Adaption an variierende Datenstromraten, Datenqualitätswerte und -anforderungen.

Zuerst wird geprüft, ob die Nutzeranforderungen technisch realisierbar sind, oder durch Konflikte der Teilziele eine Erfüllung ausgeschlossen ist. Im letzteren Fall muss der Konflikt an den Nutzer zurückgeleitet werden, damit dieser neue Anforderungen definieren kann.

Heuristische Optimierungsalgorithmen nähern die optimale Problemlösung an, indem die erreichten Zielfunktionswerte iterativ verbessert werden. Verschiedene Lösungsmöglichkeiten werden untersucht, bewertet und verglichen. Da der Prozess der Optimierung die laufende Datenstromverarbeitung nicht stören darf, wird er auf eine unabhängige logische Systemkomponente ausgegliedert. Wie in Abbildung 6.7 dargestellt, wird die Optimierung parallel zur traditionellen Datenstromverarbeitung ausgeführt. Der Verarbeitungspfad wird in der Optimierungskomponente gespiegelt. Alle Operatoren müssen kopiert werden, um Lösungsmöglichkeiten der optimalen Konfiguration zu testen und zu vergleichen. In jeder Iteration des Optimierungsalgorithmus bestimmt die aktuell geprüfte Lösungsmöglichkeit die Konfiguration der Sampling- bzw. Interpolationsoperatoren, die Gruppengrößen der angewandten Aggregationen, Frequenzanalysen und -filter, sowie die initiale Größe der Datenqualitätsfenster.

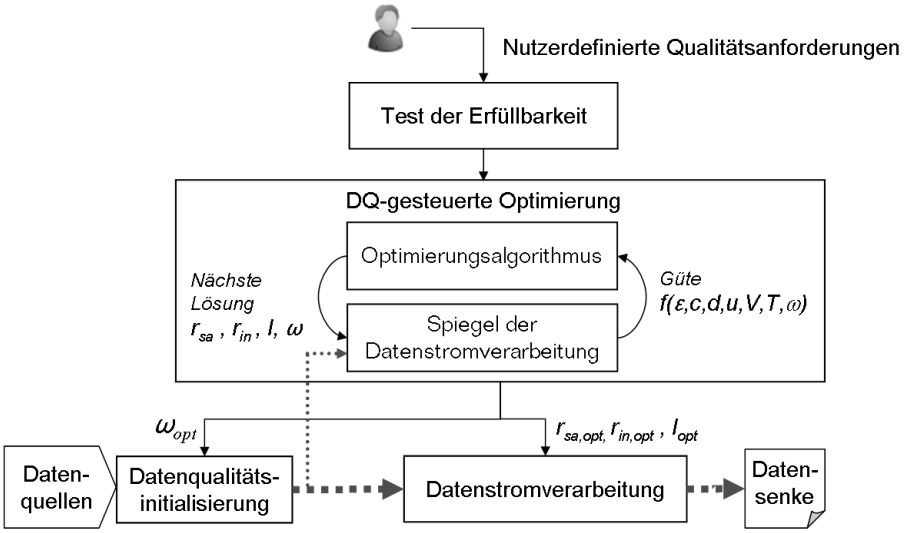


Abbildung 6.7.: Optimierungsprozess

Nach der Konfiguration wird eine repräsentative Partition des Datenstroms durch den gespiegelten Verarbeitungspfad geleitet. Die Auswahl dieser Partition wird im nachfolgenden Abschnitt beschrieben. Sobald sie vollständig verarbeitet wurde, werden die erreichte durchschnittliche Datenqualität und Granularität, der durchschnittliche Datenqualitätsfehler und das benötigte Datenstromvolumen bestimmt und an den Optimierungsalgorithmus zurückgegeben, um die Güte (engl. fitness) der geprüften Lösungsmöglichkeit zu bestimmen. Sie wird entweder mit Hilfe der monokriteriellen Zielfunktion f_{single} als gewichtete Summe aller Teilziele oder als Pareto-Dominanz der erreichten Zielfunktionswerte in f_{multi} berechnet. Wurden die Nutzeranforderungen nicht erreicht, dient der Zielfunktionswert zur Auswahl der nächsten zu prüfenden Lösungsmöglichkeit, so dass die erreichte Güte schrittweise verbessert wird.

Die Optimierung terminiert, wenn alle Nutzeranforderungen erfüllt werden. Die gefundenen Optimierungsparameter werden an den realen Verarbeitungspfad übergeben. Die Sampling- und Interpolationsoperatoren werden mit Hilfe der optimierten Sampling- und Interpolationsraten $r_{sa,opt}$ und $r_{in,opt}$ aktualisiert. Aggregationen, Frequenzanalysen und -filter werden mit ihren jeweiligen Gruppengrößen l_{opt} versehen. Die Berechnung der initialen fensterbasierten Datenqualitätsinformationen erfolgt an den Sensorknoten. Sie werden mit Hilfe der Fenstergrößen ω_{opt} aktualisiert.

Die logische Trennung von Optimierung und Datenstromverarbeitung ermöglicht auch die physische Trennung dieser Prozesse. So kann die Optimierung zum Beispiel auf einem separaten Serverknoten ausgeführt werden und hat keinen negativen Einfluss auf die Performanz der traditionellen Datenstromverarbeitung.

Erfüllbarkeit der Nutzeranforderungen

Die Erfüllbarkeit der Qualitätsanforderungen muss in vier Kontrollen überprüft werden, um die Optimierung bezüglich unerfüllbarer Nutzeranforderungen zu vermeiden (siehe Algorithmus 6.4).

Zuerst wird der gewünschte maximale statistische Messfehler untersucht. Die bestmögliche Konfidenz wird erreicht, wenn kein Sampling-Operator ausgeführt wird. Allein die initialen statistischen Messfehler der Sensoren beeinflussen die Konfidenz entsprechend der Gauß'schen Fehlerfortpflanzung. Die geforderte Konfidenz ϵ_{req} kann daher nie kleiner als die Wurzel der quadratischen Summe der initialen Messfehler ϵ_{S_i} der Sensoren S_i ausfallen (Zeile 1).

Zweitens können nur Unvollständigkeiten aufgrund von Interpolationen durch die datenqualitätsgesteuerte Optimierung reduziert bzw. behoben werden. Die geforderte Vollständigkeit c_{req} darf die durchschnittliche Vollständigkeit, welche die Sensoren S_i zur Verfügung stellen können c_{S_i}, nicht überschreiten (Zeile 2).

Die konfligierenden Optimierungsziele des minimalen Datenstromvolumens V, des minimalen Datenqualitätsfehlers δq und der maximalen Datenmenge d werden in Zeile 3 bewertet. Der geforderte Datenqualitätsfehlers δq_{req} gibt die Fenstergröße ω_{req} vor. Mit Hilfe des gewünschten Datenstromvolumens V_{req} und Gleichung 6.15 auf Seite 134 kann die benötigte Datenstromlänge m_{req} bestimmt werden. Die Datenmenge eines Datentupels kann die Gesamtstromlänge nicht überschreiten, so dass gelten muss: $d_{req} \leq m_{req}$.

Schließlich wird in Zeile 4 der Konflikt zwischen Datenmenge d und Granularität G auf Basis der maximalen Datenstromrate r_{max} kontrolliert. Zum Beispiel erlaubt die geforderte Datenmenge $d_{req} = 100$ bei einer maximalen Datenrate von $r_{max} = 1/ms$ eine minimale Granularität von $G_{min} = 100ms$.

Algorithmus 6.4 : Überprüfung der Erfüllbarkeit

Input : $\epsilon_{req}, c_{req}, V_{req}$ user-defined requirements
Output : sat=FALSE satisfiability

1 **if** $\epsilon_{req} \geq \sqrt{\sum_{i=1}^{|S|} \epsilon_{S_i}^2}$ * *Erfüllbarkeit der gewünschten Konfidenz* * \
2 $\wedge c_{req} \leq \frac{1}{|S|} \sum_{i=1}^{|S|} c_{S_i}$ * *Erfüllbarkeit der geforderten Vollständigkeit* * \
3 $\wedge d_{req} \leq \frac{V_{req}}{n+1+\sum_{i=1}^{n} \frac{1}{\omega_{req}} \vartheta}$ * *Konflikt zw. Datenmenge, -volumen und DQ-Fehler* * \
4 $\wedge G_{req} \geq \frac{d_{req}}{r_{max}}$ * *Konflikt zwischen Granularität und Datenmenge* * \
5 **then**
6 sat = TRUE;

Auswahl der Datenstrompartition

Die zur Optimierung verwendete Datenstrompartition stellt jeweils das Fenster der letzten ζ Datenstromtupel dar. Es wird entweder im Batch-Modus zu Beginn jeder Optimierung ausgewählt und ungeändert für jede Iteration eines Optimierungsdurchlaufes gleichbleibend verwendet werden (siehe Abbildung 6.8 a). Zum Anderen kann das Fenster für jede Iteration mit den aktuellen Datenstromtupeln gefüllt werden, um den dynamischen Verlauf des Datenstroms widerzuspiegeln und eine kontinuierliche Optimierung zu ermöglichen.

Die Evaluierung der Konfigurationsmöglichkeiten anhand der gespiegelten Datenverarbeitung ist der zeitaufwändigste Schritt der Optimierung. Um die Optimierungszeit zu reduzieren, muss im Batch-Modus ein kurzer Datenstromausschnitt genutzt werden, der trotzdem den vollen Wertebereichsumfang und Verlauf der Messdaten repräsentiert. Der Parameter ζ bestimmt somit den Kompromiss zwischen repräsentativer Partition und der Dauer des Optimierungsprozesses.

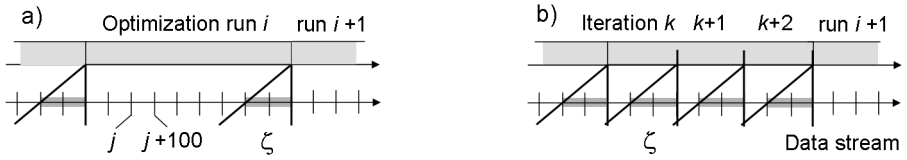


Abbildung 6.8.: Vergleich der kontinuierlichen und Batch-Optimierung

Der kontinuierliche Optimierungsansatz verfolgt das dynamische Stromverhalten, indem die Partition für jede Iteration des Optimierungsalgorithmus mittels der jeweils letzten ζ Datenstromtupel aktualisiert wird. Wie in Abbildung 6.8 b zu sehen, wird dadurch ein größerer Datenstromausschnitt zur Optimierung herangezogen. Die Datenbasis der Optimierung wird in jeder Iteration ausgetauscht. Dadurch kann eine gute Lösungsmöglichkeit, die aus der in Iteration k erreichten Gütefunktion abgeleitet

wurde, bei Anwendung auf die Datenstrompartition $k+1$ in Iteration $k+1$ zu sehr schlechten Ergebnissen führen. Die Gütefunktion steigt nicht monoton an, sondern kann divergieren, so dass kein globales Optima erreicht wird. Um die Terminierung des Optimierungsalgorithmus zu gewährleisten, müssen zusätzliche Kriterien wie die maximale Optimierungsdauer oder erlaubte Anzahl an Güteberechnungen definiert werden.

Die statische Optimierung im Batch-Modus garantiert das Konvergieren der Gütefunktion und damit das Erfüllen der Nutzeranforderungen. Die berechnete optimale Lösung gilt jedoch nur für eine geringe Strompartition und muss fortwährend durch weitere Optimierungsdurchläufe kontrolliert werden. Der kontinuierliche Ansatz erlaubt die Einbeziehung dynamischer Datenstromveränderungen in die Optimierung. Dadurch können jedoch Divergenzen der Gütefunktion entstehen, denen mittels zusätzlicher Terminierungsregeln begegnet werden muss.

Heuristische Evolutionsstrategie

Evolutionäre Algorithmen [Mic94], [Gol89] wurden durch die Prinzipien der biologischen Evolution inspiriert. Sie nutzen die evolutionären Operatoren Rekombination, Mutation und Selektion, um eine Population möglicher Lösungen an die geforderten Zielfunktionen anzupassen. Sie stellen stochastische, populationsbasierte Suchheuristiken dar, die nur die Zielfunktion benötigen, um das optimale Ergebnis aufzufinden. Während die Klasse der Genetischen Algorithmen auf binären Repräsentationen der Optimierungsparameter arbeiten, wurden die Methoden der Evolutionsstrategien speziell für die Verarbeitung reellwertiger Parameter entwickelt und sind somit auf das gegebene Optimierungsproblem der Datenqualitätsverbesserung anwendbar.

Um die Ergebnisdatenqualität durch Konfiguration der Datenstromverarbeitung zu verbessern, muss die generische Algorithmenstruktur an das definierte Optimierungsproblem angepasst werden. Dazu wird in diesem Abschnitt die qualitätsgesteuerte Evolutionsstrategie einschließlich spezifischer Funktionen für Rekombination, Mutation und Selektion vorgestellt.

Algorithmus 6.5 zeigt die Gesamtstruktur der QES. Zuerst wird in Zeile 2 die Population $P(0)$ als zufällige Auswahl von Lösungen des gesamten Suchraumes *domain* initialisiert. Der erste Schritt des stetig wiederholten Iterationsprozesses kombiniert Individuen der aktuellen Population $P(t)$ in Zeile 4, um neue Lösungskandidaten $P_c(t)$ zu erzeugen. Diese werden mutiert, um neuen Lösungen den Zutritt in die Population zu ermöglichen (Zeile 5). Schließlich werden sie mit Hilfe der Zielfunktionen f_{single} oder f_{multi} bewertet und geordnet, um die besten Individuen in $P_c(t)$ und $P(t)$ zur Erzeugung der nächsten Generation $P(t+1)$ zu selektieren. Die qualitätsgesteuerte Evolutionsstrategie terminiert, wenn alle Nutzeranforderungen erfüllt wurden. Weitere Terminimierungskriterien sind die erlaubte Anzahl der mit Hilfe des gespiegelten Verarbeitungspfades überprüften oder die vorgegebene Optimierungsdauer.

Algorithmus 6.5 : Qualitätsgesteuerte Evolutionsstrategie

Input : *domain* of possible inputs,
DQ user-defined requirements
Output : P population of optimal solutions

```
1  t = 0;
2  initialize(P(t), domain);          \* Zufällige Auswahl einer Initialpopulation * \
3  while DQ.notAchieved() do
4      P_c(t) = recombine(P(t));      \* Rekombination der Populationselemente * \
5      mutate(P_c(t));                \* Mutation * \
6      P(t+1) = selectNextGeneration(P_c(t), P(t));          \* Selektion * \
7      t = t + 1;
8  end
```

Die Rekombination dient zur Kombination positiver Eigenschaften der Lösungsmöglichkeiten einer Population. Zuerst werden μ zufällige Eltern aus der aktuellen Generation ausgewählt, die in Zweierpaaren zu λ Kindern kombiniert werden. Die dim Konfigurationsparameter der Kinder werden als Mittelwert der elterlichen Parameter berechnet. Da die Größe der Aggregationsgruppen und Datenqualitätsfenster nur ganzzahlige Werte erlaubt, werden diese gerundet.

Algorithmus 6.6 : Mutation der Evolutionsstrategie

Input : $P_c(t)$ current candidate population
Output : $P_c(t)$ mutated candidate population

```
1  forall candidate ∈ P_c(t) do
2      index = random(1, dim + 3);        \* Auswahl des zu mutier. Parameters * \
3      oldValue = candidate.getParameterAt(index);
4      if type(index) = sampling || interpolation then
5          newValue = oldValue ± Δr_sa/in;      \* Mutation einer Samplingraten * \
6      else if type(index) = groupSize then
7          newValue = oldValue ± Δl;            \* Mutation einer Gruppengröße * \
8      else if type(index) = windowSize then
9          newValue = oldValue ± Δω;            \* Mutation einer Fenstergröße * \
10     else
11         newValue = oldValue ± 0.1 · oldValue;     \* Mutationsschrittweite * \
12     candidate.setParameterAt(index, newValue);    \* Aktualisierung des
       Parameters * \
13 end
```

Algorithmus 6.6 illustriert die Mutation der Population der erzeugten Lösungskandidaten. Aufgrund unterschiedlicher Wertebereiche wird für jede Parameterklasse eine spezifische Mutationsschrittweite Δp verwendet: Sampling- und Interpolationsratenänderung $\Delta r_{sa/in}$ in Zeile 4, Änderung der Gruppengrößen Δl (Zeile 6) sowie der Daten-

qualitätsfenstergröße $\Delta\omega$ (Zeile 8). Um die Verwendung der QES zu vereinfachen und die automatische Anpassung der Schrittweite zu ermöglichen, werden die spezifischen Schrittweiten als zusätzliche Optimierungsparameter hinzugefügt. Der Parametervektor wird um drei Variablen erweitert: $dim' = dim + 3$. Die Schrittweiten selbst werden um je $\Delta = 10\%$ mutiert (siehe Zeile 10).

Die Mutation wird individuell für jeden Lösungskandidaten *candidate* der aktuellen Population $P_c(t)$ ausgeführt. Zuerst wird der zu mutierende Parameter zufällig ausgewählt (Zeile 1-2) und mit Hilfe der gegebenen Schrittweite geändert. Das neue Lösungsindividuum wird dann durch Austausch des mutierten Parameters in der aktuellen Lösung *candidate* erzeugt (Zeile 11).

Schließlich evaluiert die Selektion die neuen Lösungskandidaten, um die nächste Populationsgeneration aufzubauen. Die qualitätsgesteuerte Evolutionsstrategie folgt dem $(\mu + \lambda)$-Ansatz, der eine monoton steigende Zielfunktion bietet. Eine Generation besteht dabei aus μ Lösungsmöglichkeiten, die zu λ Kindern kombiniert werden. Die Güte der Eltern und Kinder wird mit Hilfe der verwendeten Zielfunktion berechnet. Die μ besten Lösungen bilden die neue Generation $P(t + 1)$ als Startpunkt der nächsten Algorithmeniteration. In Abschnitt 7.4.2 werden die vorgestellten Zielfunktion f_{single} und f_{multi} hinsichtlich ihrer Performanz und der erreichten Optimierung miteinander verglichen.

6.3. Zusammenfassung

Um den Ressourceneinschränkungen in Datenstromsystemen zu genügen, werden Load-Shedding-Verfahren angewendet, um in Überlastsituationen überzählige Datentupel aus dem Datenstrom zu entfernen. Alle existierenden Verfahren teilen das Problem des Datenverlustes. Aggregationsanfragen können nur angenähert beantwortet werden. Während des Datenstromverbunds müssen Datentupel entfernt werden, bevor sie zum Verbundergebnis beitragen können.

Um diese Probleme zu lösen und Anforderung 7 aus Abschnitt 2.5 zu genügen, wurde der datenqualitätsgesteuerte Lastausgleich vorgestellt. Durch die Integration von Datenqualitätsinformationen ist es mit *MaxDQ* möglich, die Gesamtqualität der Aggregationsergebnisse zu verbessern, indem „schlechte“ Tupel aus dem Datenstrom entfernt werden. *MaxDQcompensate* fokussiert die Dimensionen Genauigkeit und Konfidenz, um den eingefügten statistischen Fehler zu kompensieren. Da nicht nur der Lastausgleich die Vollständigkeit von Anfrageergebnissen beeinflusst, integrieren *MaxCompleteness* und seine hybriden Erweiterungen Informationen über fehlende Datentupel in der DQ-Dimension Vollständigkeit, um die Integrität der Verbundergebnismengen zu maximieren.

Genügt die Datenqualität nicht den Anwendungsanforderungen, wird die Optimierung der Datenverarbeitung zur weiteren Datenqualitätsverbesserung genutzt. Die

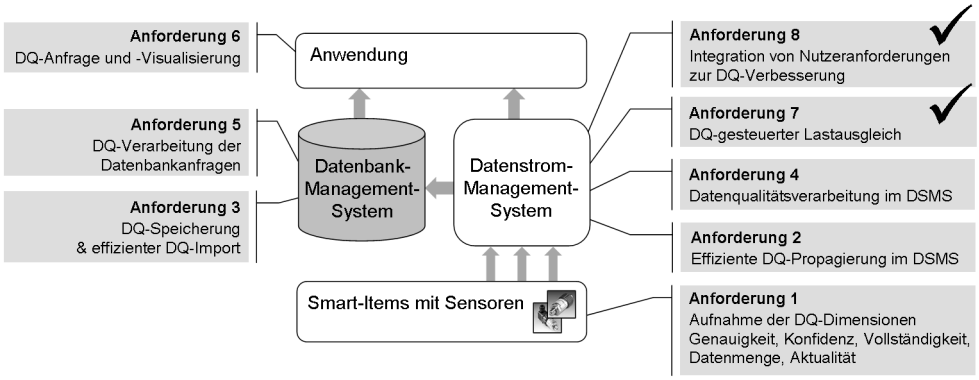


Abbildung 6.9.: Erfüllung der Anforderung 8

Operatoren des Datenqualitätsmodells DQMx wurden untersucht, um Kandidaten für die Datenqualitätsverbesserung zu identifizieren. Des Weiteren wurde die Größe der Datenqualitätsfenster als wichtiger Parameter bestimmt.

Die qualitätsgesteuerte Konfiguration der Datenstromverarbeitung definiert ein multikriterielles, nicht-lineares, kontinuierliches, beschränktes Optimierungsproblem, das mit Hilfe der vorgestellten qualitätsgesteuerten Evolutionsstrategie gelöst werden kann. Um den Optimierungsprozess parallel zur traditionellen Datenstromverarbeitung ausführen zu können, wurde zudem ein generisches Rahmenwerk entworfen. Zuerst wird die Erfüllbarkeit der gestellten Anforderungen überprüft. Dann wird die Operatorkonfiguration anhand einer repräsentativen Datenstrompartition optimiert. Die Validierung in Abschnitt 7.4.2 zeigt, dass damit auch Anforderung 8 erfüllt ist.

7 Validierung

Dieses Kapitel dient zur umfassenden Validierung aller vorgestellten Konzepte und Methoden des DQMx. Abschnitt 7.1 stellt zunächst die Validierungsziele und die Experimentierumgebung vor. Danach zeigt Abschnitt 7.2 die Evaluierungsergebnisse der Datenqualitätsmodellierung. Während Abschnitt 7.3 die in Kapitel 5 vorgestellten Methoden zur Berechnung der Datenqualitätseinflüsse unterschiedlicher Operatoren evaluiert, widmet sich Abschnitt 7.4.1 speziell dem qualitätsgesteuerten Lastausgleich. Abschnitt 7.4.2 fasst die Validierung der qualitätsgesteuerten Optimierung der Datenstromverarbeitung zusammen. Schließlich werden in Abschnitt 7.5 verschiedene Ideen der Datenqualitätsvisualisierung illustriert.

7.1. Ziele und Vorgehen

In der Validierung soll festgestellt werden, ob das entwickelte Datenqualitätsmodell DQMx alle Anforderungen erfüllt, die zu Beginn der Arbeit aufgestellt wurden. Dazu werden die in Abschnitt 2.1 eingeführten Anwendungsszenarien aufgegriffen. Die Eigenschaften der verwendeten Datenströme und Anfragen werden in den entsprechenden Abschnitten detailliert beschrieben.

Die Evaluierung der Datenqualitätsmodellierung und -verarbeitung erfolgt anhand der vorausschauenden Wartungsplanung von Hydraulikanlagen im Anwendungsszenario 1. Der qualitätsgesteuerte Lastausgleich wird mit Hilfe des vierten Szenarios anhand hochvolumiger Wetterdaten validiert. Der Einfluss des Load-Shedding auf Aggregationsergebnisse wird am Beispiel der durchschnittlichen Wolkendichte untersucht. Um die Auswirkungen des Lastausgleichs beim Verbund ungeordneter Datenströme zu zei-

gen, werden Wetterdaten aufeinander folgender Jahre verknüpft. Darüber hinaus dient die Kontrolle der Kontaktlinsenproduktion zur Analyse der Optimierungsalgorithmen zur Verbesserung der Datenqualität, da hier besonders auf die Geschwindigkeit der Datenstromverarbeitung geachtet werden muss.

Abbildung 7.1 illustriert die Architektur von QPIPEZ (Quality PIPEs & visualiZation), der prototypischen Umsetzung des entwickelten Datenqualitätsmodells DQMx. Modellierung und Verarbeitung strömender Datenqualitätsinformationen sowie der qualitätsgesteuerte Lastausgleich wurden in QPIPES realisiert, das eine Erweiterung des Datenstrom-Management-Systems PIPES (Public Infrastructure for Processing and Exploring Streams) [KS04] darstellt. Zur persistenten Speicherung der Verarbeitungsergebnisse und ihrer Qualitätseigenschaften wurde das Datenbank-Management-System Apache Derby [Apa09] zu Derby/Q erweitert. Die heuristischen Optimierungsalgorithmen zur Datenqualitätsverbesserung wurden auf Basis von JavaEva [Zel09] implementiert.

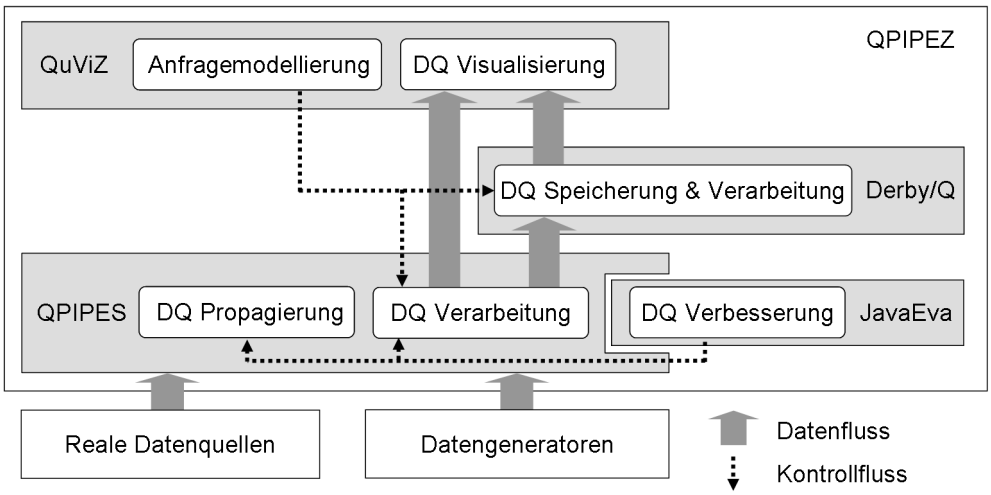


Abbildung 7.1.: Architektur der prototypischen Implementierung

Um Anforderung 6 der vorliegenden Arbeit zu erfüllen, muss Nicht-Informatikern die Nutzung des entwickelten Systems ermöglicht werden. Deshalb wurde die graphische Benutzeroberfläche QuViZ (Quality VisualiZation) mit zwei Hauptkomponenten entworfen und als Eclipse-Plugin sowie als eigenständige Anwendung umgesetzt. Zum Einen können Operatorketten zur Verarbeitung von Sensordatenströmen in QPIPES sowie persistenter Messwerttabellen in Derby/Q graphisch modelliert werden. Zum Anderen werden die Ergebnisse der Datenverarbeitung sowie die resultierende Datenqualität visuell dargestellt. Alle Aspekte der entwickelten Benutzeroberfläche werden in Abschnitt 7.5 beschrieben.

7.2. Effiziente Qualitätsmodellierung

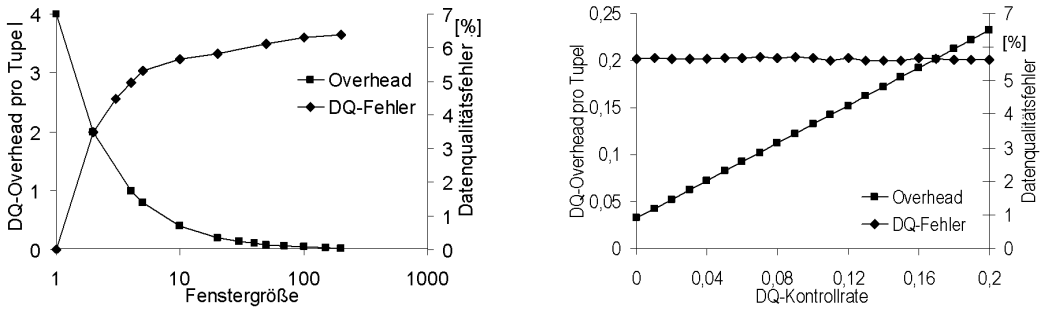
Modellierung und Verarbeitung von Datenqualitätsinformationen werden anhand der voraussichtlichen Wartung eines hydraulisch gesteuerten Baggerarms evaluiert. Mit Unterstützung des Instituts für Fluidtechnik der Technischen Universität Dresden wurde die Überwachung des Hydrauliksystems in QPIPEZ modelliert und simuliert. Anhang A.1) stellt die verwendeten Datenströme sowie das Modell der Datenverarbeitung im Detail vor.

7.2.1. Datenqualitätsvolumen und Datenqualitätsfehler

Datenqualitätsinformationen stellen zusätzliches Datenvolumen dar, das im Datenstrom propagiert und verarbeitet werden muss. Da die zur Verfügung stehenden Ressourcen, wie Speicherplatz und CPU-Kapazität, beschränkt sind, muss das benötigte Volumen minimiert werden. Dazu wurden in Abschnitt 4.1.2 Datenqualitätsfenster eingeführt, die die Datenqualität für eine Gruppe von Tupeln zusammenfassen.

Abbildung 7.2 zeigt das zusätzliche Datenvolumen (DQ-Overhead) pro Tupel in Abhängigkeit von der Größe der verwendeten Datenqualitätsfenster. Entsprechend Definition 2.2 der Sensordatenqualität wurden vier Datenqualitätsdimensionen modelliert. Mit steigender Fenstergröße sinkt das benötigte Datenvolumen. Bereits ab einer Fenstergöße von $\omega > 20$ liegt es unter 20% des Gesamtvolumens. Der naive Ansatz (siehe Abschnitt 4.1.1) hingegen würde das Gesamtvolumen des Datenstroms verfünffachen.

Durch die Übertragung der fensterbasierten Datenqualitätsverwaltung auf das relationale Datenmodell in Abschnitt 4.2.1 erfolgt auch die persistente Speicherung von DQ-Informationen mit geringem zusätzlichen Aufwand.



a) Fenstergröße b) Qualitätskontrollrate

Abbildung 7.2.: Datenqualitätsfehler und DQ-Overhead

Um die fensterbasierte Datenqualität zu berechnen, werden die tupelweisen DQ-Informationen eines Datenqualitätsfensters gemittelt. Der Datenqualitätsfehler als Distanz zwischen Fenster- und Tupeldatenqualität ist ebenfalls in Abbildung 7.2 angetragen.

Beispielhaft ist der Genauigkeitsfehler einer Druckmessung aus Anwendungszenario 1 dargestellt.

Wie in Theorem 4.1 postuliert, steigt der Datenqualitätsfehler mit wachsender Fenstergröße. Auffällig ist der steile Anstieg bis zu einer Fenstergröße von $\omega = 10$. In diesem Bereich variieren die DQ-Informationen stark von Fenster zu Fenster, so dass Detailwissen bei Fenstervergrößerungen verloren geht. Die Genauigkeit der Druckmessung wird um $\delta a \leq 5\%$ über- bzw. unterschätzt. Je größer die Datenqualitätsfenster sind ($\omega \geq 100$), umso gleichförmiger werden die ihnen anhaftenden Qualitätsinformationen. Eine weitere Vergrößerung führt nur zu geringem Anstieg des Datenqualitätsfehlers $\delta a \leq 7,5\%$. Der Nachteil sehr großer Fenster liegt in der Verzögerung der DQ-Informationen, die erst am Ende eines Datenqualitätsfensters ausgeliefert werden können.

Die dynamische Adaption der Fenstergröße auf Basis des Datenqualitätsfehlers benötigt neben den fensterbasierten DQ-Informationen zusätzliches Datenvolumen der Qualitätskontrolltupel. Das Datenvolumen steigt mit der Qualitätskontrollrate (siehe Abbildung 7.3b). Der berechnete DQ-Fehler ist konstant, allerdings steigt seine Zuverlässigkeit mit der verwendeten Kontrollrate und damit dem Datenvolumen.

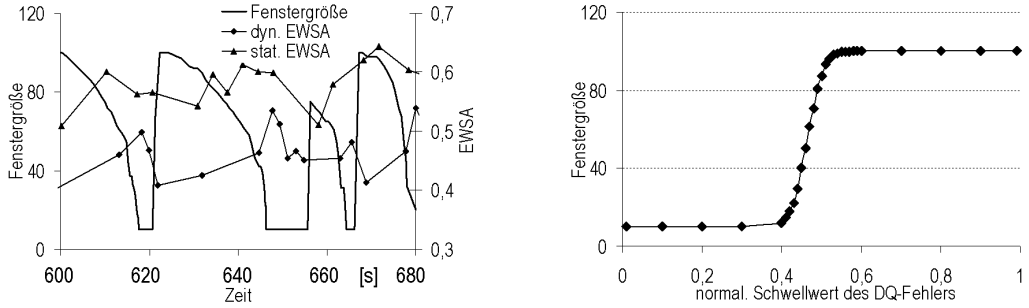
7.2.2. Dynamische Fenstergrößenadaption

Zur Bereitstellung hochwertiger DQ-Informationen mit minimalem Speicheraufwand wurden in Abschnitt 4.3 zwei Methoden zur Bestimmung der optimalen Fenstergröße vorgestellt, die im Folgenden die anhand des Hydraulikszenarios validiert werden.

Qualitätskontrolle

Mit Hilfe der Qualitätskontrolle wird sichergestellt, dass der Datenqualitätsfehler δq den gewünschten Schwellwert nicht überschreitet und ein akzeptables Datenvolumen zur Verwaltung der DQ-Informationen verwendet wird. Die Fenstergröße wurde auf $10 \leq \omega \leq 100$ begrenzt, um das maximale Datenvolumen zu beschränken und DQ-Informationen innerhalb eines akzeptablen Zeitrahmens zur Verfügung zu stellen.

Die Relation zwischen Fenstergröße ω und normalisiertem Trend des Datenqualitätsfehlers $EWSA$ (siehe Gleichung 4.8 auf Seite 78) ist in Abbildung 7.3a dargestellt. Geringe Datenqualitätsfehler, widergespiegelt in kleinen Werten des Fehlertrends $EWSA$, führen zu großen Datenqualitätsfenstern und umgekehrt. Steigt der Fehlertrend an, wird die Fenstergröße reduziert bis das akzeptierte Kontrollintervall um den gewünschten Schwellwert wieder erreicht ist. Liegt der Fehler im Kontrollintervall, wird die Fenstergröße erhöht, um das Datenvolumen zu minimieren und wichtige Ressourcen zu sparen. Ein globales Konvergieren zu einer optimalen Fenstergröße ist sehr unwahrscheinlich, da Sensormessdaten sowie Datenqualitätsinformationen stetigen Veränderungen unterliegen.



a) Adaption der Fenstergröße b) Einfluss des Fehlerschwellwertes

Abbildung 7.3.: Evaluation der gütegesteuerten Fenstergrößenadaption

Die Parameter der Qualitätskontrolle sind die initiale Fenstergröße ω_{init}, der Glättungsfaktor β und der Schwellwert $s_{\delta q}$ des akzeptierten Datenqualitätsfehlers (siehe Abschnitt 4.3.2). Experimente zeigen, dass die dynamische Fenstergröße unabhängig von der initialen Größe ist. Der Glättungsfaktor verlangsamt die Schwingungen, ohne einen direkten Einfluss auf die resultierenden Fenstergrößen zu haben. Abbildung 7.3b zeigt die durchschnittliche Fenstergröße in Abhängigkeit vom gewünschten Schwellwert des Datenqualitätsfehlers, der das akzeptierten Kontrollintervall definiert. Ein niedriger bzw. hoher Schwellwert führt zu kleinen bzw. großen Datenqualitätsfenstern, die zu den Schranken $\omega = 10$ und $\omega = 100$ konvergieren. Der Kurvenverlauf der Fenstergröße spiegelt die in Abschnitt 4.3.2 eingeführte Transformation des Datenqualitätsfehlers sowie des Schwellwertes zur kumulativen Normalverteilung wider.

Um die Vorteile der dynamischen Fenstergrößenanpassung zu unterstreichen, wurde das Experiment mit statischer Fenstergröße mit gleichem Volumenaufwand $V_{stat} = V_{dyn}$ wiederholt. Um das zusätzliche Volumen der Qualitätskontrolltupel zu kompensieren, ist eine kleinere statische Fenstergröße ω_{stat} zum Vergleich erlaubt. Sie wird wie folgt von der durchschnittlichen dynamischen Fenstergröße ω_{dyn} abgeleitet.

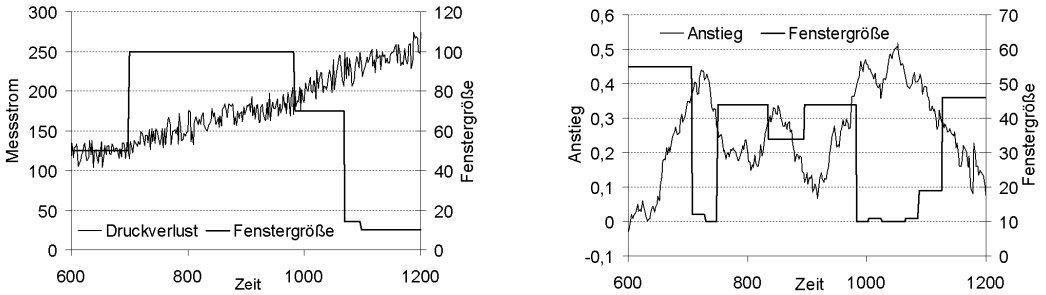
$$\omega_{stat} = \frac{1}{\frac{1}{\omega_{dyn}} + r_c} \tag{7.1}$$

Obwohl eine kleinere Fenstergröße erlaubt ist, liegt der Fehlertrend der statischen Fenstergröße im gesamten Datenstrom über dem dynamischen Fehler (siehe Abbildung 7.3a). Die Größenanpassung mit Hilfe der Qualitätskontrolle kann demnach zur Reduzierung des Datenvolumens und/oder Verbesserung der Qualitätsgüte genutzt werden.

Größendefinition mit Hilfe der Datenstrominteressantheit

Die Adaption der Fenstergröße auf Basis der Interessantheit wird an zwei Beispielfunktionen evaluiert.

Abbildung 7.4a zeigt den Druckverlust in einem hydraulischen Zylinder. Sobald Messwerte den Schwellwert $p_{loss} = 200bar$ überschreiten, d.h. im kritischen Bereich liegen, wird die Fenstergröße reduziert, um eine genaue Analyse der Verarbeitungsergebnisse zu ermöglichen. Die Fenstergröße wird entsprechend Gleichung 4.14 auf Seite 82 so lang verkleinert, wie der Schwellwert überschritten wird.



a) Hoher Druckverlust b) Anstieg des Druckverlustes

Abbildung 7.4.: Fenstergrößenadaption anhand der Datenstrominteressantheit

Die Fenstergrößenanpassung mit Hilfe interessanter Messwertanstiege ist in Abbildung 7.4b dargestellt. Interessante Datenstrompartitionen werden durch einen Anstieg größer $0,4bar/s$ definiert. Stark steigende Druckverlustwerte werden mit kleinen Datenqualitätsfenstern beschrieben, während relativ konstante Messwerte große Datenqualitätsfenster zur Folge haben. In diesem Beispiel wird die zeitlich unabhängige Definition der Fenstergröße genutzt (siehe Gleichung 4.13). Da die Anstiegsberechnung eine komplexere Aggregationsfunktion darstellt als der einfache Schwellwertvergleich, entsteht eine Verzögerung der Fenstergrößenanpassung.

7.3. Qualitätsabschätzung in der Datenverarbeitung

Die Datenqualitätsverarbeitung wird ebenfalls am Beispiel des Hydraulikszenarios evaluiert. Abschnitt 7.3.1 validiert die DQ-Verarbeitung der numerischen Algebra. Die Datenqualitätstheoreme der Signalverarbeitung werden in Abschnitt 7.3.2 überprüft. Anschließend werden die Methoden der relationalen DQ-Algebra in Abschnitt 7.3.3 kontrolliert. Die Operatoren Projektion, Verbund und Schwellwertvergleich werden nicht einzeln evaluiert, da sie entweder keinen Einfluss auf die Datenqualität haben, aus Basisoperatoren zusammengesetzt werden können oder in komplexeren Operatoren (z.B. Selektion) enthalten sind. Die Evaluierung beschränkt sich für jeden Operator auf die jeweils signifikant beeinflussten Datenqualitätsdimensionen.

Um Funktion und Anwendbarkeit der aufgestellten Theoreme zu kontrollieren, werden zuerst die wahren Datenströme $\hat{X}$ des Hydraulikszenarios entsprechend der Beschreibungen in Anhang A.1 simuliert. Anschließend werden sie mit Hilfe der gegebenen Initialfehler verfälscht, um statistische und systematische Messfehler zu simulieren. Außerdem werden Sensorausfälle durch zufälliges Löschen einiger Datentupel nachgestellt. Um die Datenqualitätsverarbeitung der DQ-Operatoren $o^{DQ} \in F^{DQ}$ zu beurteilen, erfolgt die Datenverarbeitung F für den wahren Datenstrom $\hat{X}$ sowie den verrauschten Strom X, der realen Sensormesswerten entspricht. Schließlich wird die berechnete Datenqualität DQ_Y mit der Differenz des wahren Verarbeitungsergebnisses $\hat{Y}$ und des Messwertergebnisses Y verglichen (siehe Abbildung 7.5).

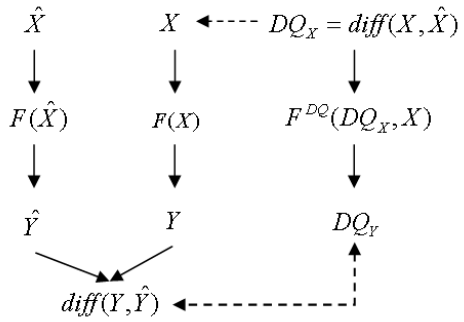


Abbildung 7.5.: Validierungsstrategie der Datenqualitätsverarbeitung

Die Datenqualitätsdimensionen Genauigkeit und Konfidenz repräsentieren numerische Messfehler. Sie werden mit Hilfe der relativen Messfehlerabweichung *rel_dev* evaluiert, die die normalisierte Differenz aus wahrem durchschnittlichen Messfehler und mittleren Datenqualitätswerten aller Datenqualitätsfenster beschreibt.

$$rel\_dev = \frac{\frac{1}{m}|Y - \hat{Y}| - \frac{1}{\kappa}\sum_{k=1}^{\kappa} a_w(k) + \epsilon_w(k)}{\frac{1}{m}|Y - \hat{Y}|} \tag{7.2}$$

7.3.1. Numerische Operatoren

Die numerischen Operatoren werden am Beispiel der Subtraktion zur Berechnung des Druckverlustes untersucht. Ausgehend von einem Datenstrom mit zwei Druckmessreihen wird die relative Fehlerabweichung des Subtraktionsergebnisses und die Propagierung der Vollständigkeit und des Datenvolumens analysiert. In Abschnitt 5.2.3 wurde gezeigt, wie Boolesche Operatoren auf numerische abgebildet werden, so dass die im Folgenden vorgestellten Validierungsergebnisse auch für diese Operatorklasse gelten.

Abbildung 7.6 zeigt die relative Messfehlerabweichung in Prozent. Je größer die Datenqualitätsfenster, umso größer ist die Messfehlerabweichung, das heißt umso schlechter wird der Ergebnismessfehler mit Hilfe der Gauß'schen Fehlerfortpflanzung (siehe Theoreme 5.3 und 5.4) abgeschätzt. Dies liegt im Qualitätsfehler δq begründet, der nach Theorem 4.1 mit der Fenstergröße wächst. Da der Qualitätsfehler gegen eine obere Schranke konvergiert, nähert sich auch die relative Messfehlerabweichung einer Schranke von $rel\_dev = 7\%$ an. Die relative Messfehlerabweichung für Datenqualitätsfenster praktikabler Größe zwischen $\omega = 10$ und $\omega = 100$ liegt bei $1\% \leq rel\_dev \leq 5,5\%$ und ist damit ausreichend klein, um eine Abschätzung von Genauigkeit und Konfidenz zu erlauben.

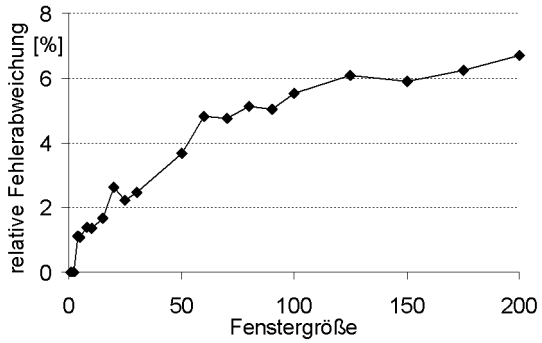
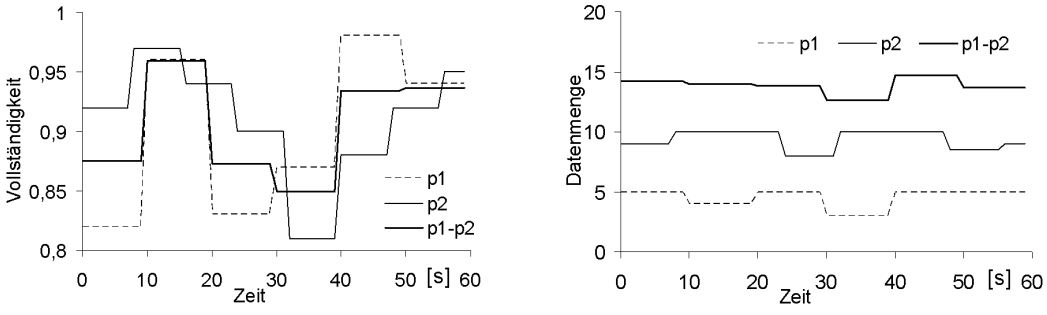


Abbildung 7.6.: Relative Fehlerabweichung bei der Subtraktion

Die Ergebnisvollständigkeit in Abbildung 7.7a wird entsprechend Theorem 5.2 als Durchschnitt der eingehenden Vollständigkeiten berechnet. Das Datenvolumen (Abbildung 7.7b) summiert die Rohdatenvolumina der Eingangsattributströme nach Theorem 5.1. Um Speicherplatz und Verarbeitungszeit zu sparen, bestimmt bei unterschiedlichen Eingangsfenstergrößen die jeweils größere Fensterlänge die Struktur der resultierenden Datenqualitätsfenster.



a) Vollständigkeit b) Datenmenge

Abbildung 7.7.: Evaluation der Subtraktion

7.3.2. Signalanalyse

Dieser Abschnitt befasst sich mit der Evaluierung des Sampling- und Interpolationsoperators am Beispiel der Stromratenangleichung zweier asynchroner Druckmessreihen $p1$ und $p2$. Anschließend wird die Frequenzanalyse mit Hilfe simulierter Schwingungsmessungen untersucht.

Sampling

Abbildung 7.8 zeigt die relative Fehlerabweichung, die durch Sampling mit verschiedenen Samplingraten und einer nachfolgenden Summation von jeweils 10 Datentupeln entsteht. Wieder zeigt sich, unabhängig von der Samplingrate, dass die Messfehler umso genauer bestimmt werden können, je kleiner die Datenqualitätsfenster definiert wurden.

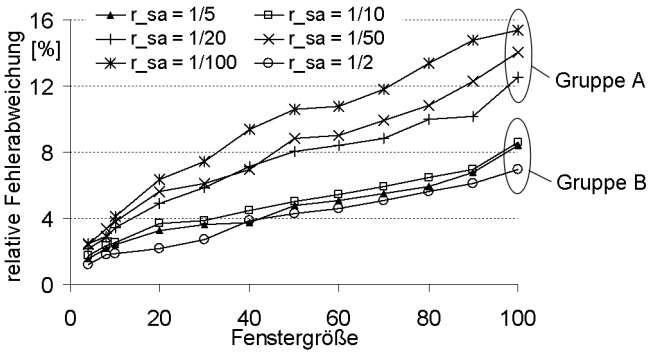


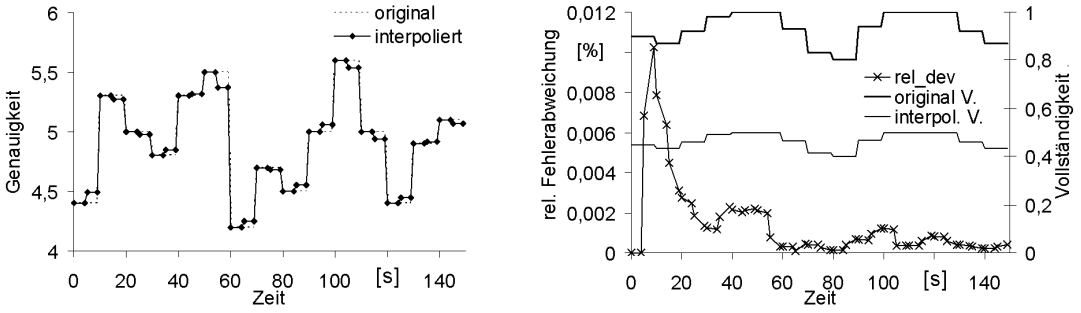
Abbildung 7.8.: Evaluation des Samplingoperators

Kleine Samplingraten ($r_{sa} \leq 0,05$, Gruppe A in Abbildung 7.8) entfernen einen beträchtlichen Teil der Messwerte aus dem Datenstrom. Der eingefügte statistische Fehler dominiert den Gesamtmessfehler. Da der Samplingfehler nur mit eingeschränkter Güte [Haa97] abgeschätzt werden kann, wird eine relativ hohe Messfehlerabweichung eingefügt. Bei einer Samplingrate von $r_{sa} \geq 0,1$ verbleibt ein ausreichender Tupelanteil im Datenstrom, so dass der systematische Fehler, der durch initiale Sensorungenauigkeiten hervorgerufen wurde, den eingefügten Samplingfehler überwiegt. Die Propagierung der Genauigkeit kann besser abgeschätzt werden, so dass die Messfehlerabweichung geringer ausfällt (Gruppe B).

Interpolation

Die Interpolation wurde anhand eines Messdatenstroms mit der Fenstergröße $\omega = 10$ und der Interpolationsrate $r_{in} = 2$ untersucht. Da die Datenqualitätswerte während der Interpolation ebenfalls interpoliert werden, sind die generierten Fenstergenauigkeiten

in Abbildung 7.9a nahezu kongruent mit dem ursprünglichen Genauigkeitsverlauf. Der interpolierte Datenstrom weicht lediglich im jeweils zweiten Kindfenster vom Original ab (siehe Theorem 5.7). Selbiges gilt für die DQ-Dimensionen Konfidenz und Datenvolumen.



a) Genauigkeit b) Fehlerabweichung und Vollständigkeit

Abbildung 7.9.: Evaluation der Interpolation

Die Interpolation bringt nur sehr geringfügige Unsicherheiten in die Fehlerabschätzung. Deshalb geht die relative Fehlerabweichung in Abbildung 7.9b für diesen Operator gegen $rel\_dev = 0\%$. Die Vollständigkeit wird um die Interpolationsrate reduziert, d.h. halbiert, um die neu eingefügten Datentupel widerzuspiegeln.

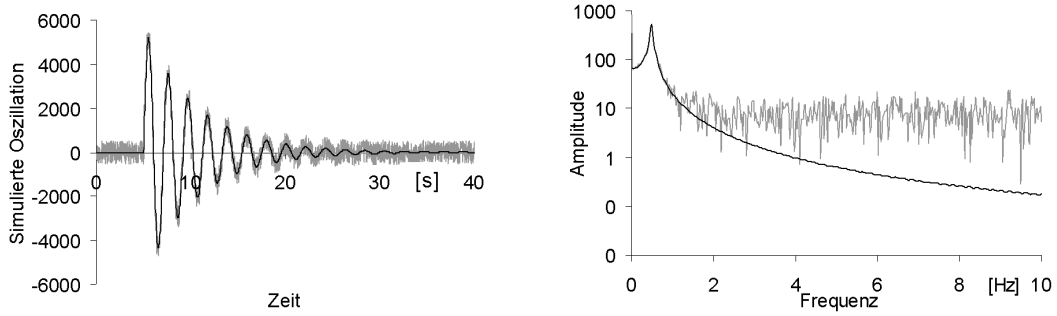
Frequenzanalyse

Frequenzanalyse und -filter wurden anhand eines simulierten oszillierenden Messdatenstroms validiert. Die Funktion $\hat{f}(t)$ in Gleichung 7.3 definiert eine abfallende Oszillation beginnend zum Zeitpunkt $t_0 = 5s$ mit der Startamplitude $A = 5720$, dem Schwächungskoeffizienten $a = -0,18/s$ und der Schwingungsfrequenz $f = 0,48Hz$. Um numerische Messfehler zu simulieren, wurde die wahre Funktion $\hat{f}(t)$ mit Hilfe zufälliger Genauigkeits- bzw. Konfidenzfehler $a(t) \leq 0,1 \cdot A,\ \epsilon(t) \leq 0,1 \cdot A$ zu $f(t)$ in Gleichung 7.4 verrauscht.

$$\begin{aligned} \hat{f}(t) &= A \cdot e^{(a(t-t_0))} sin(2\pi f(t-t_0)) \\ &= 5720 \cdot e^{(-0,18/s(t-5s))} sin(2\pi 0,48Hz(t-5s)) \quad (7.3) \\ f(t) &= \hat{f}(t) + a(t) + \epsilon(t) \quad (7.4) \end{aligned}$$

Beide Funktionen wurden im Intervall $[0s; 41s]$ mit einer Frequenz von $50Hz$ abgetastet, so dass 2048 Messdatenpunkte aufgenommen und die FFT mit $N = 2^p = 2048 = 2^{11}$ ermöglicht wurde. Abbildung 7.10a zeigt den wahren Datenstrom (schwarz) sowie den verrauschte Messstrom (grau).

Die Fourier-Transformation beschreibt das Frequenzband zwischen $0Hz$ und der Faltungsfrequenz, die der halben Abtastfrequenz entspricht, mit Hilfe von $\eta = N/2 + 1 = 1025$ komplexen Koeffizienten. Der reelle Anteil entspricht der Amplitude der Signalfrequenzen (siehe Abbildung 7.10b). Die Frequenzanalyse sowohl des wahren als auch des verrauschten Datenstroms zeigt die korrekte Schwingungsfrequenz $f = 0,48Hz$ mit maximaler Amplitude an. Die zufällig gestreuten Messfehler führen jedoch zusätzliche hochfrequente Signalanteile mit kleiner Amplitude ein.



a) Signal im Zeitbereich b) Signal im Frequenzbereich

Abbildung 7.10.: Frequenzanalyse eines oszillierenden Signals

Ziel der Datenqualitätsverarbeitung ist die Voraussage dieser Signalanteile im Frequenzbereich mit Hilfe der Genauigkeits- und Konfidenzinformationen im Zeitbereich. Die Differenz zwischen wahrem Frequenzband $\hat{Y}$ und verrauschtem Frequenzband Y wird mit den transformierten Datenqualitätsinformationen $DQ_Y = X_{a_w} + X_{\epsilon_w}$ verglichen. Wie in Abbildung 7.11 zu sehen, können Fehler im Frequenzband sehr exakt $(rel\_dev \leq 0,17\%)$ bestimmt werden. Die Fourier-Transformation fügt demnach keinen Fehler in die Datenverarbeitung oder Qualitätspropagierung ein.

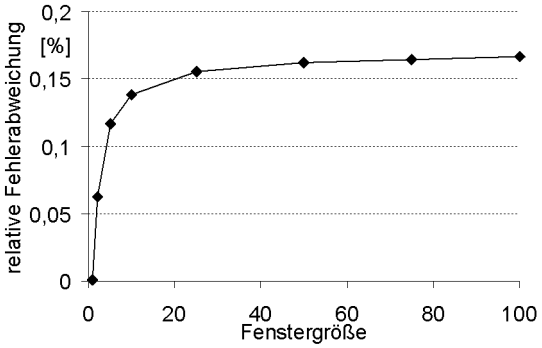


Abbildung 7.11.: Relative Fehlerabweichung bei der Frequenzanalyse

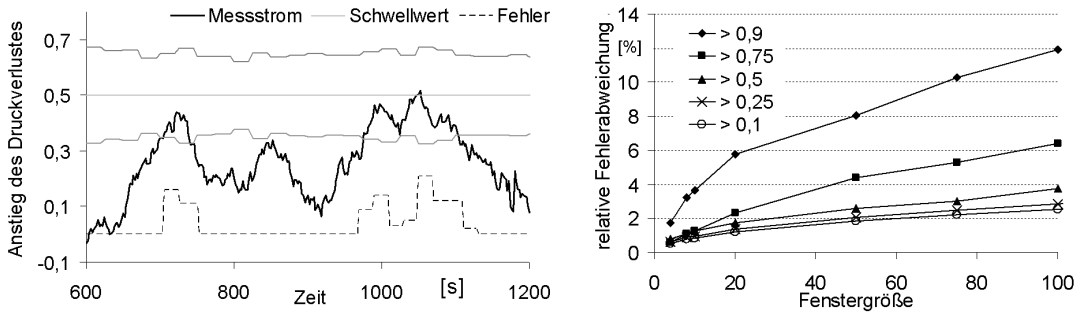
7.3.3. Relationale Operatoren

Im Folgenden werden die in Abschnitt 5.4 aufgestellten Theoreme der Datenqualitätspropagierung der relationalen Algebra evaluiert. Sie können sowohl auf strömende Sensordaten als auch auf statische Datenbanktabellen angewendet werden. Die Attributprojektion hat keinen Einfluss auf Qualitätsinformationen (siehe Theorem 5.13). Eine Evaluierung ist daher nicht erforderlich.

Selektion

Den kritischen Punkt bei der Datenqualitätsverarbeitung der Selektion bildet der unsichere Bereich um den Selektionsschwellwert (siehe Abschnitt 5.4.3). Systematische und statistische Messfehler des Datenstroms und der Schwellwertfunktion führen zu fehlerhaften Selektionsentscheidungen. Abbildung 7.12a zeigt den für die Selektion interessanten Ausschnitt des untersuchten Beispieldatenstroms. Die Steigung des Druckverlustes wird gegen den Schwellwert $0,5 bar/s$ verglichen. Der unsichere Bereich ist mit Hilfe der Genauigkeits- und Konfidenzinformationen am Schwellwert angetragen.

In den Intervallen $[706;750]$ und $[970;1128]$ liegt der Anstieg des Messdatenstroms im unsicheren Bereich, so dass ein statistischer Fehler in der DQ-Dimension Konfidenz (siehe gestrichelte Linie) eingefügt werden muss. Je stärker die Messwerte eines Datenqualitätsfensters variieren, umso größer fällt dieser Konfidenzfehler aus (siehe Gleichung 5.48 auf Seite 108).



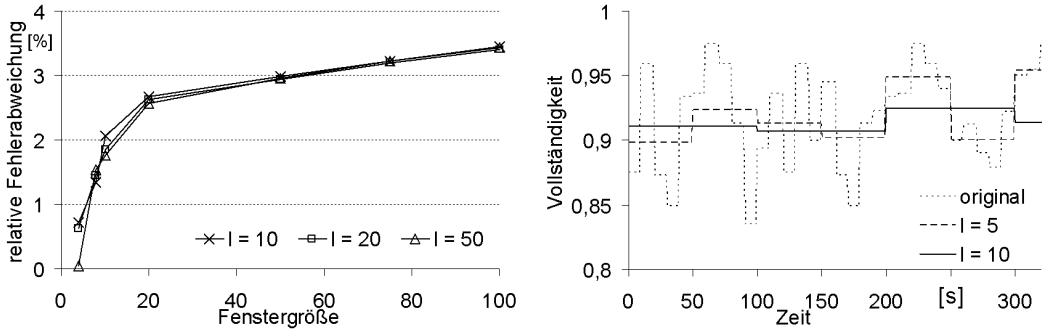
a) Eingefügter Fehler der Konfidenz b) Relative Fehlerabweichung

Abbildung 7.12.: Evaluation der Selektion

Der durch unsichere Selektion hinzugefügte Konfidenzfehler spiegelt sich in der relativen Fehlerabweichung wider. Abbildung 7.12b zeigt erneut, dass die Fehlerabweichung mit der Größe der genutzten Datenqualitätsfenster steigt. Die Auswirkung unterschiedlicher Selektionsschwellwerte zeigt die Verwandtschaft mit dem Samplingoperator. Je mehr Datentupel aus dem Datenstrom entfernt werden (je kleiner der Schwellwert), umso größer ist die resultierende relative Fehlerabweichung, da die Abschätzung des dominierenden statistischen Konfidenzfehlers der Genauigkeitspropagierung unterliegt.

Aggregation

Die interessanteste Aggregationsfunktion aus Abschnitt 5.4.5 ist die Anstiegsberechnung. Daher wird diese zur Evaluierung herangezogen. Abbildung 7.13 zeigt die relative Fehlerabweichung in Abhängigkeit von der Größe der Datenqualitätsfenster. Je größer die Datenqualitätsfenster, umso unpräziser kann der tatsächliche Messfehler abgeschätzt werden. Die Fehlerabweichung ist jedoch unabhängig von der Gruppengröße, so dass sich die Kurvenverläufe überlagern. Die Bestimmung von Genauigkeit und Konfidenz der Aggregationsergebnisse erfolgt mit Hilfe der Gauß'schen Fehlerfortpflanzung. Daher ähnelt der Verlauf der Fehlerabweichung der Aggregation dem Verhalten bei Operatoren der numerischen Algebra (siehe Abbildung 7.7a).



a) Relative Fehlerabweichung b) Vollständigkeit

Abbildung 7.13.: Evaluation der Aggregation

Während der Aggregation von Messdaten werden auch Datenqualitätsinformationen zusammengefasst. Abbildung 7.13b zeigt den Informationsverlust am Beispiel der aggregierten Vollständigkeit. Je größer die Aggregationsgruppe, umso stärker werden Messdaten und Qualitätsinformationen gemittelt.

7.4. Methoden zur Datenqualitätsverbesserung

In diesem Abschnitt werden die in Kapitel 6 vorgestellten Methoden der Datenqualitätsverbesserung evaluiert.

7.4.1. Datenqualitätsgesteuerter Lastausgleich

Im Folgenden werden die in Abschnitt 6.1 vorgestellten Load-Shedding-Algorithmen *MaxDQ*, *MaxDQcompensate* und *MaxCompleteness* untersucht, um die folgenden Fragen empirisch zu beantworten.

1. Bieten die entwickelten Load-Shedding-Algorithmen sichtbare Vorteile bezüglich der Datenqualität der Verarbeitungsergebnisse im Vergleich zu existierenden Verfahren?
2. Können die datenqualitätsgesteuerten Load-Shedding-Algorithmen hinsichtlich Verarbeitungszeit und Performanz mit gegenwärtigen Lösungen konkurrieren?

Der datenqualitätsgesteuerte Lastausgleich (DQLS) wird mit Hilfe des Anwendungsszenarios der Wetterprognose evaluiert. Zur Simulation hochvolumiger Datenströme, die zu Überlastsituationen führen und damit Load-Shedding notwendig machen, werden reale Wetterdaten [HWL08] verwendet. Jedes Datentupel wird durch den Zeitstempel (Jahr, Monat, Tag, Stunde) und Ort (Längen- und Breitengrad) der Messung gekennzeichnet und enthält unter anderem Messungen der Wolkendichte, der Sonneneinstrahlung und des aktuellen Wetters. Nähere Informationen sind im Anhang A.3 zu finden.

Zur Evaluierung der angenäherten Aggregationsanfragen, werden die Wetterdaten von Juni 1990 ausgewählt. Aggregationsanfragen mit unterschiedlichen Selektionskriterien und Gruppierungsattributen, deren Datenverarbeitungspfade Teilsegmente gemein haben, werden parallel ausgeführt. Zum Beispiel bestimmt die Anfrage

```
SELECT AVG(Gesamtwolkenbedeckung)
FROM Juni90
WHERE Breitengrad > 37470 AND Breitengrad < 37480
AND Längengrad > 12220 AND Längengrad < 12230
GROUP BY day
```

die durchschnittliche Wolkendichte in San Francisco im Juni 1990. Der Fokus dieser Validierung liegt nicht auf dem Vergleich unterschiedlicher Load-Shedding-Strukturen. Vielmehr wird die erreichte Datenqualität der Anfrageergebnisse analysiert.

Zur Evaluierung des Load-Sheddings für fensterbasierte Datenstromverbünde werden die Messungen im Juni 1991 hinzugenommen. Mit Hilfe der Verbundattribute Längen- und Breitengrad können die Wetterdaten der Landstationen für zwei aufeinander folgende Jahre verknüpft werden. Zum Beispiel kann diese Verbundstrategie genutzt werden, um Wetterveränderungen über mehrere Jahre zu untersuchen.

Der systematische Fehler wird optimistisch auf 1% der maximalen Messwertebereiche geschätzt. Die statistische Fehler kann mit Hilfe der Varianz der Messwerte berechnet werden. Um die Vollständigkeit zu initialisieren, werden fehlende Wettermessungen durch Vergleich der verfügbaren Zeitstempel mit der geplanten Messrate $r = 1/3h$ aufgedeckt. Da der Wetterdatensatz gleichmäßige Zeitstempel beinhaltet, werden variierende Datenstromraten durch das Einfügen von Verzögerungen zwischen dem Einlesen einzelner Datentupel simuliert. So können Lastfaktoren von 1 bis 10 modelliert werden, die zu effektiven Load-Shedding-Raten von $1 \geq r_{LS} \geq 0.1$ führen.

Der DQLS benötigt Qualitätsinformationen im Datenstrom. Um dieses zusätzliche Datenvolumen zu kompensieren, müssen bei gleichen Ressourcenbedingungen mehr Da-

tentupel aus dem Strom entfernt werden als bei traditionellen Load-Shedding-Verfahren. Zum fairen Vergleich der Verfahren wird die Load-Shedding-Rate wie folgt verkleinert.

$$r_{DQdrivenLS} = r_{LS} \cdot \frac{1}{1 + \frac{1}{\omega} + r_c} \tag{7.5}$$

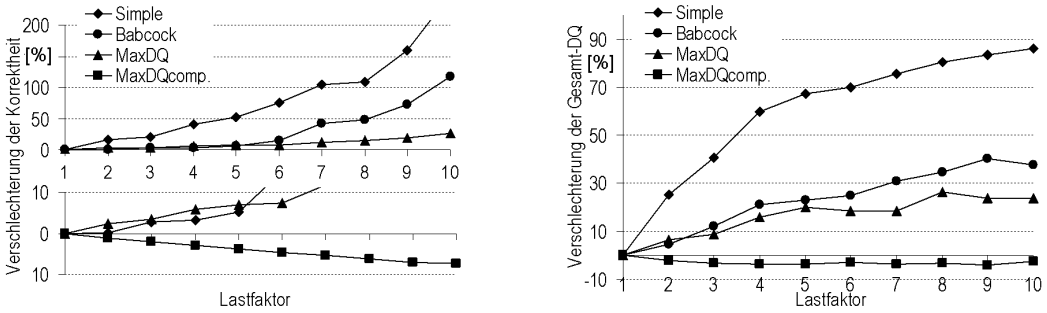
Die Größe der Datenqualitätsfenster ω beschreibt das zusätzliche Datenvolumen der Qualitätspropagierung. Die Kontrolltupelrate r_c spiegelt das Volumen der Qualitätskontrolltupel bei automatischer Fenstergrößenadaption wider. Durch Anpassung der Rate wird bei traditionellem und DQ-gesteuertem Lastausgleich das gleiche Gesamtdatenvolumen erreicht.

Gesamtqualität und Korrektheit werden beim Lastausgleich mit einfachen Zufallsstichproben (*Simple*), dem Fehler minimierenden Ansatz von Babcock et al. (*Babcock*) und den DQLS-Algorithmen *MaxDQ* und *MaxDQcompensate* verglichen. Außerdem werden Integrität und Recall der Verbundergebnismengen unter Anwendung des altersbasierten Algorithmus von (*Srivastava*) und des frequenzbasierten Load-Sheddings von (*Kang*) im Vergleich mit dem neu entwickelten *MaxCompleteness* und den hybriden Ansätzen *HybridSrivastava* und *HybridKang* untersucht. Zuletzt wird die Verarbeitungszeit pro Datentupel aller Algorithmen evaluiert.

Aggregationsanfragen

Im ersten Experiment wird die durchschnittliche Korrektheit $\alpha = a + \epsilon$ der Aggregationsergebnisse ermittelt. Abbildung 7.14a zeigt die prozentuale Verschlechterung der Korrektheit im Verhältnis zur Aggregation ohne Load-Shedding für ansteigende Lastfaktoren. Je höher der Lastfaktor, umso größer ist der eingefügte Load-Shedding-Fehler, der die Korrektheit reduziert. Die Zufallsstichprobe *Simple* ohne Mittel zur Datenqualitätsverbesserung schneidet am schlechtesten ab. Der *Babcock*-Ansatz minimiert den eingefügten Fehler, so dass die Korrektheit signifikant verbessert werden kann. Er übertrifft MaxDQ für niedrige Lastfaktoren, bei denen der Lastfaktor zu klein ist, um ausreichende Anteile der „schlechten" Datenstromtupel zu entfernen. Bei hohen Lasten in unregelmäßigen Datenströmen führt der datenqualitätsgesteuerte Lastausgleich jedoch zu besseren Ergebnissen. MaxDQcompensate hat sogar einen positiven Einfluss auf die Datenstromkorrektheit. Durch die geschickte Auswahl der Datenstromtupel kann der eingefügte Load-Shedding-Fehler kompensiert werden.

In Abbildung 7.14b wird die Analyse auf die Gesamtqualität θ^+ von Aggregationsergebnissen ausgeweitet. Das Diagramm weist Ähnlichkeiten mit Abbildung 7.14a auf. Während die einfache Zufallsstichprobe den größten Fehler hinzufügt, bietet MaxDQcompensate die besten Ergebnisse, da der Konfidenzfehler kompensiert wird. Die Dominanz von MaxDQ wird zu geringeren Lastfaktoren verschoben. Da Babcock nur auf die Minimierung der Konfidenz ausgerichtet ist, wird er früher von MaxDQ übertroffen, der die Gesamtqualität maximiert.

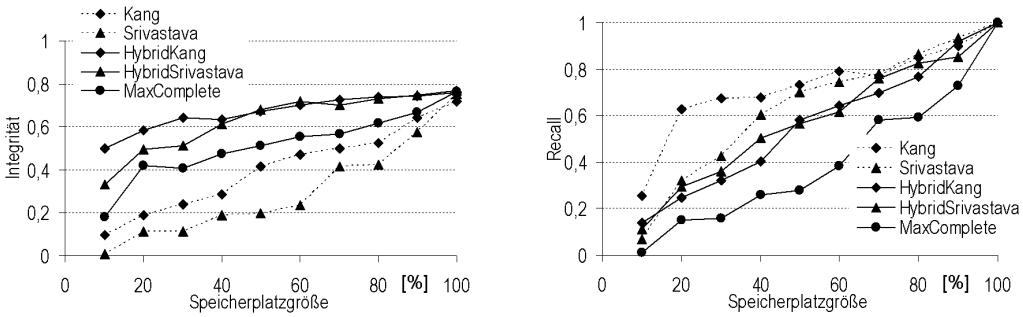


a) Korrektheit b) Gesamtqualität

Abbildung 7.14.: Qualität der Aggregationsergebnisse in Abhängigkeit vom Lastfaktor

Verbundanfragen

Im Folgenden wird der datenqualitätsgesteuerte Lastausgleich für den Datenstromverbund mit beschränktem Speicherplatz evaluiert. Die Integrität I der Ergebnismenge, die fehlende Werte aufgrund von Sensorausfällen sowie gelöschten Verbundpartnern kombiniert, ist in Abbildung 7.15a dargestellt. Um die Anwendbarkeit von MaxCompleteness zum Finden akzeptabler Verbundmengen zu beweisen, wird außerdem der Recall *rec* der Algorithmen in Abbildung 7.15b gezeigt.



a) Integrität b) Recall

Abbildung 7.15.: Verbundqualität in Abhängigkeit von der Speicherplatzgröße

Der verwendete Speicherplatz ist als prozentualer Anteil der Größe des untersuchten Datenstromfensters angetragen. Allgemein gilt, je mehr Speicherplatz zur Verfügung steht, umso weniger Datentupel müssen gelöscht werden. Der Recall nähert sich $rec = 1$ an und die Integrität konvergiert gegen die initiale Datenstromvollständigkeit.

Da Mess- und Vollständigkeitswerte der Wetterdaten nicht korreliert sind, führt MaxCompleteness eine einfache Zufallsstichprobe hinsichtlich potentieller Verbundpartner aus. Deshalb wird hier der geringste Recall erreicht, der zu mittlerer Integrität führt. Die Basisalgorithmen von Kang und Srivastava bieten zwar den besten Recall, wählen

Datentupel geringer bzw. hoher Vollständigkeit jedoch mit derselben Wahrscheinlichkeit aus, was die Integrität reduziert.

Die hybriden Algorithmen kombinieren die maximierte Vollständigkeit von MaxCompleteness mit dem hochwertigen Recall von Kang und Srivastava. Deshalb präsentieren sie die beste Ergebnisintegrität für alle Speicherplatzkonfigurationen. Einerseits ist der Recall nur geringfügig geringer als der der Basisalgorithmen. Anderseits erhöhen sie die durchschnittliche Datenstromvollständigkeit, indem Datentupel mit geringer Vollständigkeit aufgrund initialer Sensorausfälle mit höherer Wahrscheinlichkeit aus dem Eingangsdatenstrom gelöscht werden.

Performanzanalyse

Abbildung 7.16 vergleicht die Verarbeitungszeit pro eintreffendem Datentupel für alle untersuchten Load-Shedding-Algorithmen. Die höheren Verarbeitungszeiten des Verbundoperators resultieren aus der enorm höheren Komplexität dieses Datenstromoperators. Während MaxDQ im Bereich der einfachen Zufallsstichprobe liegt, ist MaxDQ-compensate ein wenig langsamer als Babcock, wobei beide den Konfidenzfehler minimieren. Gleichermaßen zeigen die hybriden Ansätze nur geringe Verlangsamungen im Vergleich zu Basisalgorithmen. Da MaxCompleteness keine Messwerte in die Load-Shedding-Entscheidung einbezieht, sondern nur auf den Fensterqualitäten arbeitet, ist hier der Tupeldurchsatz am höchsten.

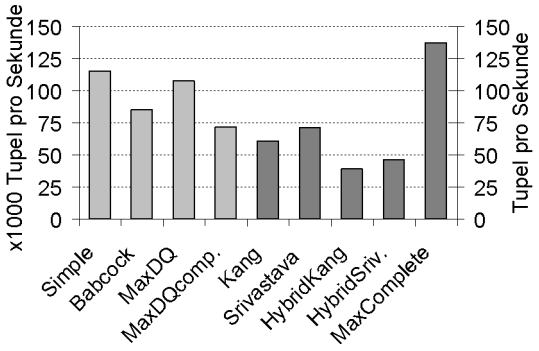


Abbildung 7.16.: Verarbeitungszeit pro Tupel

Die neuen Strategien des datenqualitätsgesteuerten Lastausgleichs übertreffen die verglichenen existierenden Algorithmen im Kontext der erreichten Datenqualität. Im Besonderen konnte die Korrektheit der approximierten Aggregationen sowie die Vollständigkeit der Verbundergebnismengen verbessert werden. Der Nachteil der etwas höheren Verarbeitungszeit ist in der zusätzlichen Analyse der Datenqualitätsinformationen begründet.

7.4.2. Optimierung der Datenstromverarbeitung

In diesem Abschnitt wird die qualitätsgesteuerte Optimierung der Datenstromverarbeitung evaluiert, um die folgenden Fragen zu beantworten.

1. Welchen Einfluss hat die Gewichtung der Teilziele auf das Ergebnis der monokriteriellen Optimierung?
2. Welche Vorteile haben mono- bzw. multikriterielle Optimierung?
3. Welche Vorteile bietet der Batch-Modus im Vergleich zur kontinuierlichen Optimierung?
4. Skalieren die vorgestellten Verfahren bei komplexen Datenstromanfragen mit hoher Sensoranzahl?

Zum Test der Gewichtungen, Optimierungs-Modi und Algorithmenperformanz werden künstliche Datenströme generiert. Sie unterliegen der Standardnormalverteilung ($\mu = 0, \sigma = 1$) und haben variierende Datenstromraten im Bereich von $1/ms \leq r \leq 100/ms$. Anfragen werden simuliert, die 2 bis 128 dieser Datenströme in Zweierpaaren verknüpfen. Jeder Datenstrom wird in jeder Anfrageebene mit einem Sampling bzw. einer Interpolation versehen, um synchrone Verbundpartner zu finden. Aggregationen werden zufällig in den Anfragebaum eingestreut.

Darüber hinaus wird das Anwendungsszenario 3 aus Abschnitt 2.1 genutzt, um die Auswirkungen der Optimierung der Datenverarbeitung an realen Messdaten zu zeigen. Der Datensatz, verfügbar unter [EA90], enthält Messwerte der Linsendicke sowie Axialverschiebung. Zur Erzeugung von Datenströmen ausreichender Länge werden Sensorwerte auf Basis der realen Messwerte simuliert (siehe Anhang A.2) und mit Hilfe der Anfragegraphen in Abbildung 6.6 auf Seite 136 verarbeitet.

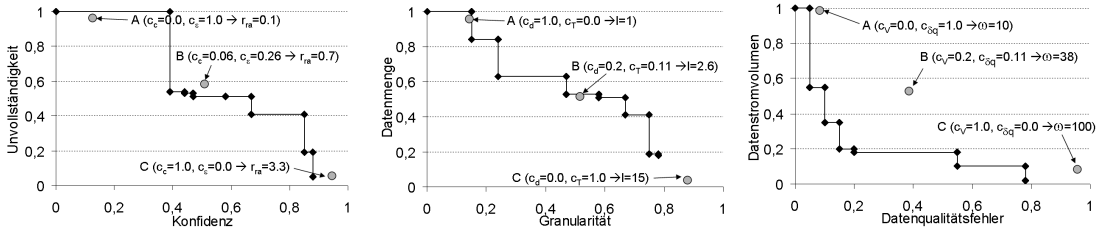
Für alle Datenströme wird ein systematischer Fehler von $a_{init} = 1\%$ angenommen. Der statistische Fehler der Konfidenz wird mit Hilfe der Tupelvarianz und der Konfidenzwahrscheinlichkeit von $p = 99\%$ bestimmt. Um Sensorausfälle zu simulieren, werden 2% der Tupel zufällig aus jedem Datenstrom gelöscht.

Auswirkung der Gewichtungen

Dieser Abschnitt beantwortet die erste der Evaluierungsfragen. Anhand signifikanter Beispielkonflikte (siehe Abschnitt 6.3) wird gezeigt, wie Teilziele mittels der Gewichtungen priorisiert werden können und welchen Einfluss dies auf die resultierende Konfiguration der Datenstromverarbeitung hat.

Abbildung 7.17a zeigt die Pareto-Front der multikriteriellen Optimierung von Konfidenz und Vollständigkeit. Minimale statistische Fehler der Konfidenz durch hohe Samplingraten können nur auf Kosten hoher Unvollständigkeit erreicht werden und umgekehrt. Die optimalen Kompromisse der Pareto-Front können in der monokriteriellen Optimierung

mit Hilfe geschickter Gewichtungen reproduziert werden. Die Punkte A,B und C in Abbildung 7.17 zeigen Beispiele der resultierenden Lösungen. Je höher die Gewichtung eines Teilzieles ist, umso stärker wird dieses Ziel optimiert.



a) Unvollständigkeit vs. Konfidenz

b) Datenmenge vs. Granularität

c) Datenstromvolumen vs. Datenqualitätsfehler

Abbildung 7.17.: Pareto-Fronten und Einfluss der Gewichtungen

Wenn die Kosten für unvollständige Datentupel die Gewichtung der Konfidenz übersteigen, werden niedrige Sampling- und Interpolationsraten vorgeschlagen (Punkt C in Abbildung 7.17a). Hohe Kosten für statistische Konfidenzfehler in Punkt A hingegen führen zu hohen Samplingraten, die einen geringeren Datenverlust verursachen. Gleichmäßig verteilte Kosten führen zu einem ausgewogenen Kompromiss der Teilziele (Punkt B).

Abbildung 7.17b stellt die Pareto-Front der Ziele der maximalen Datenmenge und maximaler Granularität dar. Hohe Datenmengen bedeuten einen langen Gültigkeitszeitraum pro Datentupel, so dass die Granularität gering ausfällt. Je stärker die Gewichtung der Datenmenge diejenige der Granularität übersteigt, umso größer ist die durchschnittliche Gruppengröße, die in der optimalen Lösung vorgeschlagen wird (siehe Gewichtungen der Punkte A,B und C).

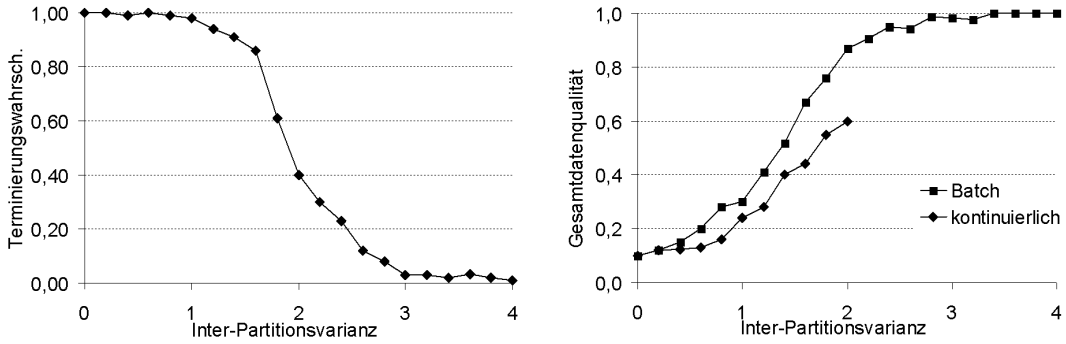
Des Weiteren wurde der Konflikt zwischen minimalem Datenstromvolumen und minimalem Datenqualitätsfehler untersucht, der die Definition der DQ-Fenstergröße bestimmt. Hohe Gewichte des Datenstromvolumens führen zu großen Datenqualitätsfenstern, so dass der DQ-Fehler erhöht wird, und umgekehrt (siehe Abbildung 7.17c).

Vergleich der Optimierungs-Modi

Im Folgenden wird die zweite Frage anhand der monokriteriellen Optimierung der Verarbeitung eines künstlich generierten Datenstroms (s.o.) beantwortet. Die Varianz innerhalb einer Partition (Intra-Partitionenvarianz) bleibt konstant bei $(\sigma^2 = 1)$. Allerdings wird eine Varianz zwischen den einzelnen Partitionen (Inter-Partitionenvarianz) $\check{\sigma}^2$ eingefügt, indem der Mittelwert jeder Partition modifiziert wird.

Abbildung 7.18a zeigt die Terminierungswahrscheinlichkeit bei steigender Inter-Partitionenvarianz. Die kontinuierliche Optimierung terminiert voraussichtlich bei einer

Varianz bis $\check{\sigma}^2 < 1$. Die Erfolgswahrscheinlichkeit sinkt für $1 \leq \check{\sigma}^2 \leq 3$ und geht gegen $p = 0$, falls $\check{\sigma}^2 > 3$.



a) Terminierungswahrscheinlichkeit b) Erreichte Gesamtdatenqualität

Abbildung 7.18.: Vergleich des kontinuierlichen und des Batch-Modus

Um die Optimierungsergebnisse des kontinuierlichen und des Batch-Modus zu vergleichen, werden die berechneten optimalen Operatorkonfigurationen auf die Verarbeitung des nachfolgenden Datenstroms angewandt und die resultierende Gesamtdatenqualität aufgezeichnet. Abbildung 7.18b zeigt die normalisierte Gesamtqualität für eine steigende Inter-Partitionenvarianz. Der Batch-Modus erlaubt gute Ergebnisse für kleine Varianzen $\check{\sigma}^2 \leq 0.5$. Der kontinuierliche Ansatz liefert auch für höhere Varianzen eine angemessene Datenqualität, da er sich besser an verändernde Datenströme anpasst. Jedoch beschränkt die Terminierungswahrscheinlichkeit die Anwendung dieses Ansatzes auf Varianzen $\check{\sigma}^2 \leq 2$. Darüber hinaus ist eine Datenqualitätsverbesserung nur mit Hilfe des Batch-Modus möglich.

Performanzanalyse

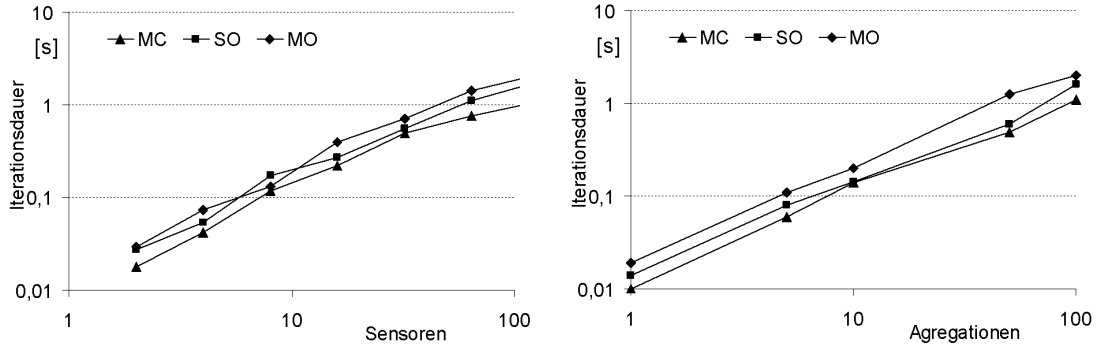
In diesem Abschnitt wird die Skalierbarkeit der vorgestellten Verfahren getestet. Es wird keine Inter-Partitionenvarianz in den verwendeten künstlichen Datenstrom eingefügt, so dass die folgenden Tests im Batch-Modus ausgeführt werden.

Zuerst wird die zeitliche Performanz in Abhängigkeit von der Anzahl der Sensoren sowie der Aggregationen bzw. Frequenzanalysen untersucht. Da die Anzahl der verwendeten Sampling- und Interpolationsoperatoren von der Sensoranzahl abhängt, ist hier keine eigene Evaluierung notwendig. Nachfolgend wird die Iterationsdauer in Abhängigkeit von der verwendeten Partitionslänge ermittelt. Zum Schluss wird die Zeitdauer bestimmt, die für die Verbesserung der Datenqualität benötigt wird.

Die Monte-Carlo-Suche (MC) basiert auf einer Zufallsstichprobe des Lösungsraumes und wird als neutrale Referenz für die Performanzmessung genutzt [PSZ89]. Die monokriterielle Optimierung wird einerseits mit zufällig gewählten Gewichtungen (SO-R),

andererseits mit zuvor erarbeiteten ausgewogenen Gewichtungen (SO-O) ausgeführt. Schließlich approximiert die multikriterielle Optimierung (MO) die Pareto-Front aller optimaler Kompromisse.

Abbildung 7.19a zeigt den Einfluss der Sensoranzahl auf die Dauer einer Iteration des Optimierungsalgorithmus bei einer festen Partitionslänge von 1000 Tupeln. Für alle Algorithmen steigt die Iterationsdauer linear mit der Anzahl der Sensoren. Je komplexer der Algorithmus, umso länger benötigt eine Iteration. Der Performanzunterschied zwischen mono- und multikriterieller Optimierung wird durch die aufwändige Pareto-Front-Berechnung verursacht. Abbildung 7.19b illustriert die Skalierbarkeit bezüglich der Anzahl an Aggregationen, Frequenzanalysen sowie -filter. Auch hier steigt die Iterationsdauer etwa linear mit der Anzahl der Aggregationen, d.h. mit der Komplexität des Verarbeitungsgraphen.

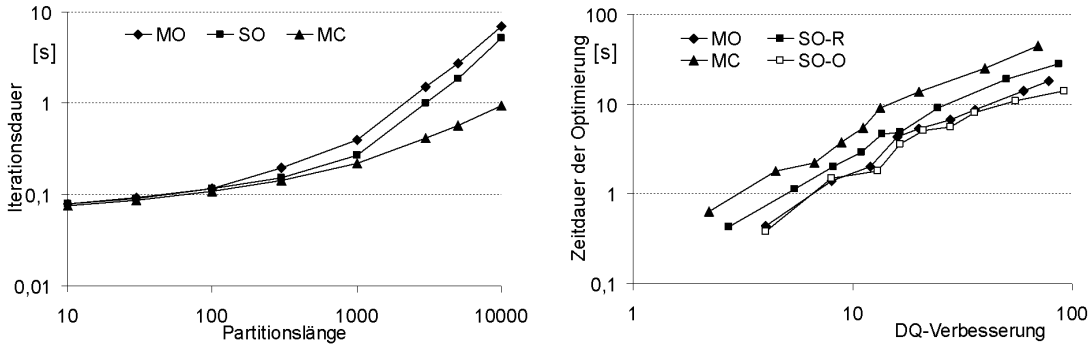


a) Sensoranzahl b) Anzahl an Aggregationen

Abbildung 7.19.: Dauer einer Optimierungsiteration

In Abbildung 7.20a wird die Länge einer Iteration für 16 Sensoren und 10 Aggregationen in Abhängigkeit von der Partitionslänge dargestellt. Die Verarbeitungszeit steigt wieder etwa linear für Partitionen bis zu 1000 Datentupeln. Erst für sehr große Partitionen zeigt die Iterationsdauer einen exponentiellen Anstieg.

Abbildung 7.20b vergleicht die zeitliche Performanz von mono- und multikriterieller Optimierung hinsichtlich der erreichten Datenqualitätsverbesserung. Die Qualitätsverbesserung wird als relative Qualitätsänderung ausgedrückt: $(\theta - \theta')/\theta$. Die Monte-Carlo-Suche (MC) erreicht das schlechteste Ergebnis, gefolgt von der zufällig initialisierten SO-R. SO-O erreicht die schnellste Qualitätsverbesserung (z.B. 1,6s für 10%). Jedoch erfordert die Bestimmung der optimalen Gewichtungen mehrere Algorithmendurchläufe der Vorverarbeitung, die wiederholt werden müssen, sobald sich die Datenstromeigenschaften oder die Nutzeranforderungen an die Datenqualität ändern. Die multikriterielle Optimierung (MO) ist zwar ein wenig langsamer (1,9s), erarbeitet aber die vollständige Liste aller optimaler Kompromisse ohne eine Vorverarbeitung zu benötigen.



a) Dauer vs. Partitionslänge b) Dauer der Datenqualitätsverbesserung

Abbildung 7.20.: Dauer einer Iteration bzw. der DQ-Verbesserung

Die Evaluierung beweist die gute Skalierbarkeit der qualitätsgesteuerten Optimierung in Bezug auf die verwendete Partitionslänge sowie die Komplexität der Datenstromverarbeitung. Daten- und Dienstqualität können innerhalb weniger Sekunden verbessert werden. Des Weiteren kann abgeleitet werde, dass die monokriterielle Optimierung im Batch-Modus die besten Ergebnisse für konstante Nutzeranforderungen und Stromeigenschaften liefert. Wenn die Datenstromtupel Fluktuationen aufweisen oder die Anforderungen oft angepasst werden müssen, bietet die multikriterielle Optimierung die bessere Lösung. In diesem Fall muss die Inter-Partitionenvarianz bestimmt werden, um zwischen kontinuierlichem oder Batch-Modus zu wählen.

Kontrolle der Kontaktlinsenproduktion

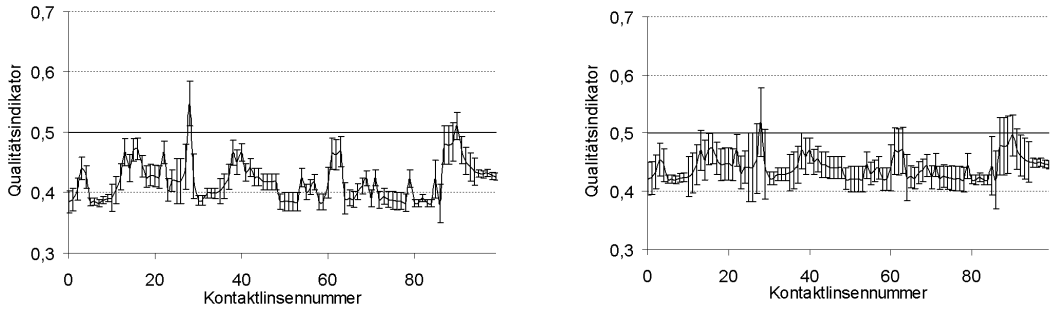
Um die praktische Anwendbarkeit der vorgestellten Algorithmen nachzuweisen, wird das Anwendungsszenario der Kontaktlinsenproduktion verwendet, das die besondere Anforderung einer schnellen Datenverarbeitung aufweist. Die Mitten- und Randstärke sowie die Axialverschiebung werden zur Bewertung der Qualität der produzierten Kontaktlinse gemessen (siehe Verarbeitungsgraph in Abbildung 6.6).

Wird der Schwellwert von $0.5mm$ überschritten, muss die Linse als Ausschuss deklariert und aus der Produktionsline entfernt werden. Aufgrund von Messfehlern und Sensorausfällen können nicht alle Kontaktlinsen eindeutig bewertet werden. Systematische und statistische Messfehler definieren einen unsicheren Bereich um den Schwellwert, in dem keine klare Entscheidung möglich ist. Um die Qualität der Kontaktlinsenproduktion zu garantieren, müssen alle nicht identifizierbaren Linsen ebenfalls als Ausschuss behandelt werden.

Ziel der Optimierung der Datenverarbeitung ist es, den unsicheren Bereich durch Minimierung des statistischen Fehlers zu verkleinern, so dass die Anzahl nicht entscheidbarer

Linsen minimiert wird. Dabei darf jedoch weder Vollständigkeit noch Datenvolumen in Mitleidenschaft gezogen werden.

Abbildung 7.21a zeigt einen Ausschnitt der Messung der Kontaktlinsenqualität vor der Optimierung der Datenverarbeitung. 15 Kontaktlinsen können nicht eindeutig beurteilt werden. Die Optimierung reduziert den statistischen Fehler, so dass nur noch vier Kontaktlinsen im nun schmaleren unsicheren Bereich liegen (siehe Abbildung 7.21b).



a) Qualitätsmessung vor der Optimierung b) Qualitätsmessung nach der Optimierung

Abbildung 7.21.: Kontaktlinsenkontrolle

Abbildung 7.22 zeigt die durchschnittliche Anzahl nicht entscheidbarer Kontaktlinsen nach 100 Iterationsdurchläufen jedes Optimierungsalgorithmus. Bereits die einfache Zufallssuche MC kann die Zahl falsch klassifizierter Linsen halbieren. Mit Hilfe der SO-R wird die Anzahl auf durchschnittlich 6,6 von 100 Linsen reduziert. Die multikriterielle Optimierung (MO) sowie die optimal gewichtete SO-O erzielen die besten Ergebnisse. Nur durchschnittlich 4,5 bzw. 4,3 von 100 Kontaktlinsen verbleiben im unsicheren Bereich.

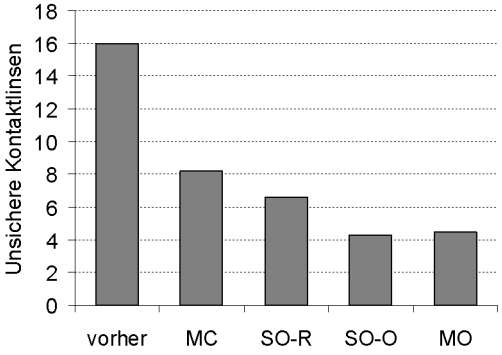


Abbildung 7.22.: Vergleich der Ergebnisse der Optimierungsverfahren

7.5. Visualisierung der Datenqualitätsergebnisse

In diesem Abschnitt wird die Benutzeroberfläche des entwickelten DQ-Management-Toolkits QPIPEZ vorgestellt, die zur Visualisierung der Datenstromanfrage einerseits und der Ergebnisdatenqualität andererseits dient.

7.5.1. Modellierung von Datenqualitätsanfragen

Um allen Nutzergruppen Zugang zur Sensordaten- und Qualitätsauswertung zu gewährleisten, wurde eine Oberfläche entwickelt, die auf bekannten, einfach zu erlernenden Konzepten graphischer Zeichenprogramme basiert (siehe Abbildung 7.23). Verfügbare Datenquellen und -senken sowie Operatoren werden in einer Palette (A) dargestellt und können durch einfaches Anklicken auf die Zeichenfläche (B) zur Modellierung eines Anfragebaumes gezogen werden. Das Verbindungswerkzeug (E) wird verwendet, um den Datenfluss zwischen den Quellen, Operatoren und Senken des Anfragegraphen zu definieren, indem Verbindungslinien zwischen jeweils zwei aufeinander folgenden Knoten gezeichnet werden. Weiterhin beinhaltet die Palette ein Werkzeug zur Objektauswahl zur Strukturierung der Anfrage.

Um in komplexen Anfragegraphen zu navigieren, wird oben rechts eine Übersichtsdarstellung (C) angeboten. Das Eigenschaftsformular (D) in der rechten unteren Ecke erlaubt die Definition von benötigten Parametern der Operatoren oder Datenquellen, wie zum Beispiel Samplingraten oder Tabellennamen. Die modellgesteuerte Entwicklung (engl. model-driven software development) [BG05b] erlaubt die einfache Erweiterung der Werkzeugpalette, um zum Beispiel neue Operatoren zu integrieren oder Smart-Item-Sensoren als Datenquellen einzubinden.

Die Schaltfläche (F) startet die modellierte Datenverarbeitung. Der Modellgraph wird in die Anfragesprache der definierten Datenquellen übersetzt, zum Quellsystem (QPIPES oder Derby/Q) transferiert und ausgeführt. Die Ergebnisse der Datenverarbeitung können wiederum in einer Datenbank abgelegt oder in der Visualisierungskomponente von QPIPEZ angezeigt werden.

7.5.2. Visualisierung der Datenqualität

Die gängigen Werkzeuge zur Visualisierung von Datenqualität orientieren sich an der tabellarischen Darstellung, bei der die absoluten Daten- und Qualitätswerte in Spalten angegeben werden. Einige Programme unterstützen die Auswertung durch farbige Hinterlegung bzw. Schriftfarben. Diese Darstellungsform ist jedoch für kontinuierliche Sensormessdaten wenig geeignet. Hier bietet sich eine graphische Darstellung in Diagrammform an.

Abbildung 7.24 zeigt die Visualisierung von Genauigkeit, Konfidenz, Vollständigkeit, Datenmenge und Aktualität der vorausschauenden Wartung eines Hydraulikbaggers.

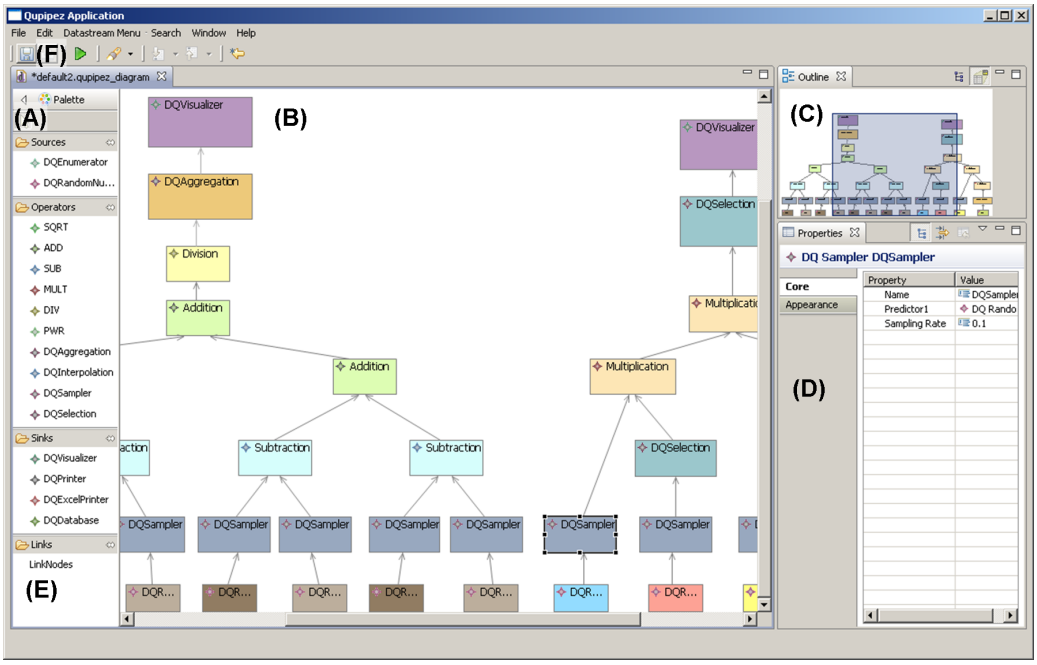


Abbildung 7.23.: Modellierungskomponente von QPIPEZ

Alle zur Verfügung stehenden Datenqualitätsinformationen werden im Kontext der Sensormessdaten (A) dargestellt. Der Nutzer kann alle bzw. eine beliebige Teilmenge der DQ-Dimensionen zur Visualisierung in Liniendiagrammen (B) auswählen. Darüber hinaus stehen erweiterte Ansichten (C) zur Verfügung. Genauigkeit und Konfidenz stellen numerische Messfehler dar, die untere und obere Schranken um den Messwert beschreiben, die den wahren Wert enthalten (siehe Abbildung 7.25a). Darüber hinaus ist eine zusammenfassende Darstellung der Verteilung der Datenqualitätswerte des gesamten Datenstroms in Kreis- oder Balkendiagrammen möglich (siehe Abbildung 7.25b).

Während Datenbankanfragen statische Verarbeitungsergebnisse liefern, werden Datenstromergebnisse in dynamischen Diagrammen dargestellt. So kann die Entwicklung der Sensordaten sowie deren qualitativer Eigenschaften überwacht werden. Außerdem bieten alle Diagramme eine Zoomfunktion, um interessante oder kritische Ergebnismengen in Detail zu betrachten.

7.6. Zusammenfassung

In der Validierung wurde gezeigt, dass die ressourcensparende Propagierung von Datenqualitätsinformationen in Datenqualitätsfenstern eine sehr gute Abschätzung der Tupelqualitäten erlaubt. Die automatische Fenstergrößenadaption kann den Datenqualitätsfehler als Abstand zwischen tupel- und fensterbasierten Qualitätsinformationen

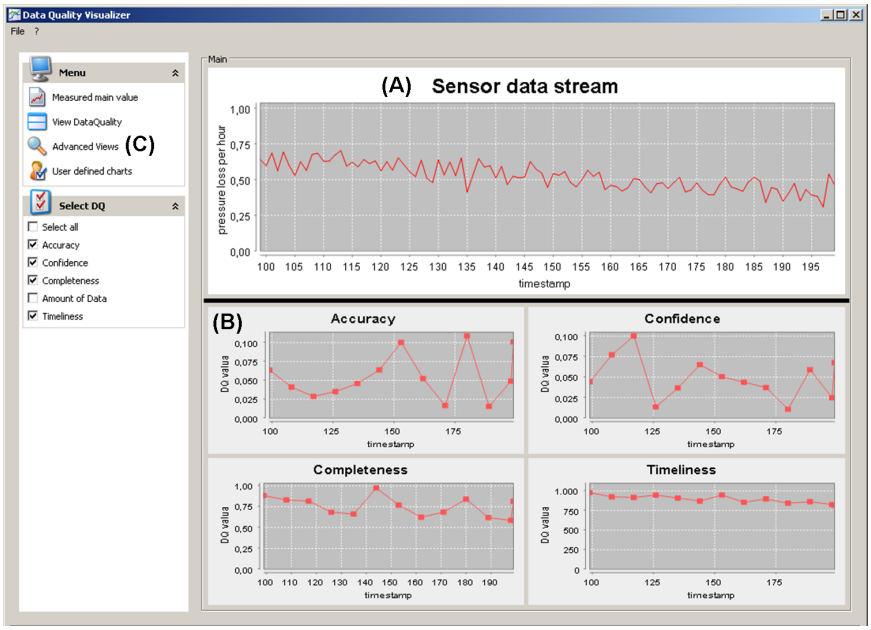
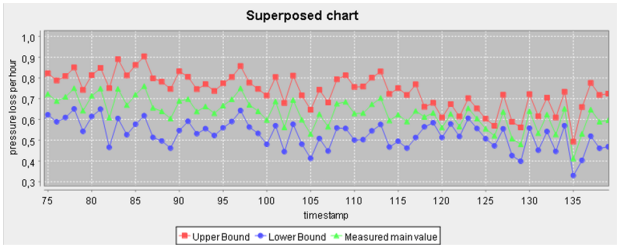
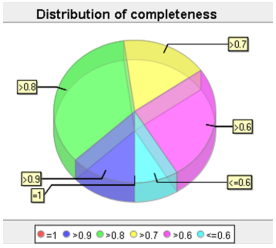


Abbildung 7.24.: Visualisierungskomponente von QPIPEZ



a) Genauigkeit und Konfidenz



b) Vollständigkeit

Abbildung 7.25.: Erweiterte DQ-Ansichten

weiter verringern. Anschließend wurde die Datenqualitätsverarbeitung validiert. Die in Kapitel 5 vorgestellten Theoreme erlauben die Datenqualitätsberechnung mit Abweichungen von weniger als 0, 5% bei Interpolation und Frequenzanalyse bis zu 15% bei Sampling und Selektion. Je kleiner die Datenqualitätsfenster definiert sind, umso genauere Aussagen können über die Fehlerfortpflanzung getroffen werden.

Erreichte Qualitätsverbesserung und Performanz des datenqualitätsgesteuerten Lastausgleichs wurden anhand der Wetterprognose untersucht. Sowohl bei Aggregations- als auch bei Verbundanfragen konnte die Datenqualität der Verarbeitungsergebnisse im Vergleich zu traditionellen Load-Shedding-Algorithmen verbessert werden. Das Szenario der Produktionskontrolle bildet den Hintergrund der Evaluierung der heuristischen Anfrageoptimierung zur Integration von Nutzeranforderungen. Es wurde gezeigt, wie

die Datenqualität durch Konfiguration der Datenstromverarbeitung verbessert werden kann, so dass mehr Kontaktlinsen sicher klassifiziert werden können.

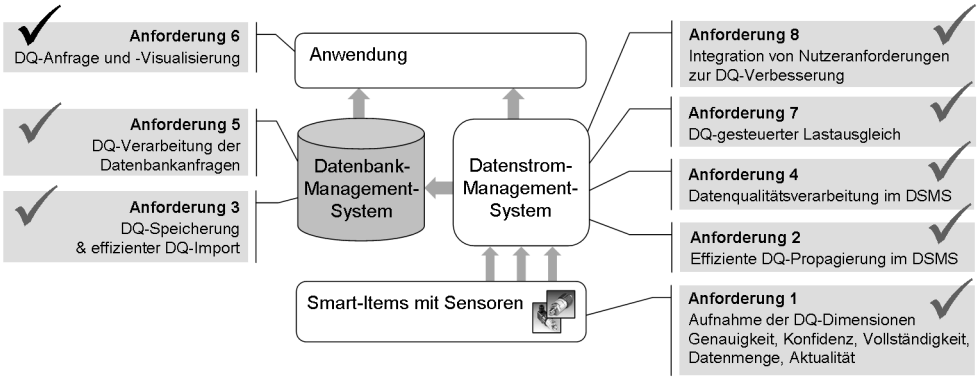


Abbildung 7.26.: Erfüllung der Anforderung 6

Schließlich wurde die Benutzeroberfläche des entwickelten Prototyps QPIPEZ vorgestellt. Modellierungs- und Visualisierungskomponente erlauben nutzerfreundliche Datenqualitätsanfragen und stellen die Qualität der Datenverarbeitungsergebnisse im Kontext der Sensordaten graphisch dar. Verschiedene Ansichten können gewählt werden, so dass eine umfangreiche Auswertung der Datenqualitätsinformationen ermöglicht und Anforderung 6 erfüllt wird.

8

Zusammenfassung und Ausblick

Die zunehmende Integration von Smart-Items in Unternehmensstrukturen erlaubt die Automatisierung von Produktions- und Geschäftsanwendungen. Sensoren liefern Messdaten zur Beschreibung der Eigenschaften eines Produktes, den Betriebszustand einer Maschine oder Informationen über deren Umwelt. Zum Beispiel können Sensorinformationen in betriebswirtschaftlichen Anwendungen zur vorausschauenden Wartung, zur Steuerung und Optimierung von Produktionsprozessen oder zur Überwachung der Produktqualität eingesetzt werden. Darüber hinaus spielen Sensoren in Wissenschaft und Forschung eine große Rolle.

Die Qualität von Sensormessdaten wird durch sensoreigene und externe Fehlerquellen, wie Umwelteinflüsse oder Sensorausfälle, beschränkt. Aufgrund des hohen Datenvolumens und der automatisierten Datenerfassung können Datenqualitätsprobleme in sensorgestützten Anwendungen nicht durch Datenbereinigung behoben werden. Vielmehr setzen sie sich in der Datenverarbeitung fort, die zudem zusätzliche Datenfehler einführen kann. Um Prozess- und Entscheidungsfehler auf Basis fehlerbehafteter Sensordaten zu verhindern, muss die Sensordatenqualität kontrolliert werden.

Kontinuierliche Sensormessungen werden in Sensordatenströmen gebündelt und in Datenstrom-Management-Systemen verarbeitet. Diese Systeme müssen erweitert werden, um Datenqualitätsinformationen aufzuzeichnen und dem Datenkonsumenten zur Bewertung der Sensordaten zur Verfügung zu stellen. Die Analyse bestehender Datenqualitätsmodelle ergab, dass keines den beschränkten Ressourcen typischer Smart-Item-Anwendungen entspricht. Deshalb wurde in Kapitel 4 das Datenqualitätsmodell DQMx zur effizienten Datenqualitätsverwaltung in Datenströmen vorgestellt. Datenqualitätsinformationen werden nicht für jedes einzelne Datenstromelement, sondern aggregiert in Datenqualitätsfenstern übertragen. Die Fenstergröße kann auf Basis des

Datenqualitätsfehlers sowie der Interessantheit des Datenstroms definiert und während der Laufzeit adaptiert werden. Des Weiteren werden Datenstrukturen sowie Methoden zur persistenten Qualitätsspeicherung in relationalen Datenbanken beschrieben. Das fensterbasierte Datenqualitätsmodell beantwortet die erste Forschungsfrage aus Abschnitt 1.3.

Um die Qualität von Datenverarbeitungsergebnissen zu bestimmen, müssen sämtliche Verarbeitungsschritte auf den Datenqualitätsinformationen nachvollzogen werden. Bisherige Ansätze liefern jedoch nur sehr eingeschränkte Methoden zur Bestimmung der Datenqualitätsfortpflanzung. Sie sind einerseits auf wenige Datenqualitätsdimensionen, andererseits auf einen geringen Operatorenumfang beschränkt. Deshalb wurde in Kapitel 5 eine umfassende Datenqualitätsalgebra zur Beantwortung der zweiten Forschungsfrage entwickelt. Untersucht wurden Operatoren der traditionellen Datenstromverarbeitung, der Signalanalyse sowie der numerischen Algebra, die speziell in sensorbasierten Systemen Anwendung finden.

Entspricht die berechnete Ergebnisdatenqualität nicht den Anforderungen des Nutzers, müssen Maßnahmen zur Datenqualitätsverbesserung ergriffen werden. Kapitel 6 widmet sich dieser Aufgabe. Der datenqualitätsgesteuerte Lastausgleich begegnet Überlastsituationen, indem Tupel minderer Qualität aus dem Datenstrom entfernt werden. Insbesondere wurden Strategien zur Verbesserung der Ergebnisqualität bei Aggregations- und Verbundanfragen geschaffen. Reicht diese Verbesserung nicht aus, kommt die qualitätsgesteuerte Optimierung der Datenstromverarbeitung zum Einsatz. Mit Hilfe heuristischer Optimierungsalgorithmen wird die optimale Konfiguration der Datenverarbeitungsoperatoren bestimmt, um Nutzeranforderungen zu integrieren und damit die Datenqualität der Verarbeitungsergebnisse zu verbessern. Kapitel 6 beantwortet damit die dritte Forschungsfrage.

Die prototypische Realisierung des DQMx stellt die Antwort auf die vierte Forschungsfrage. Sie umfasst ein Werkzeug zur nutzerfreundlichen, graphischen Komposition von Datenverarbeitungsgraphen sowie eine Komponente zur Visualisierung der Ergebnisse der Datenstromverarbeitung und der resultierenden Datenqualitätsinformationen.

Alle vorgestellten Konzepte und Methoden wurden anhand verschiedener Anwendungsszenarien validiert. Die durchgeführten Analysen sind in drei Teile gegliedert. Der erste Teil zeigt, dass die fensterbasierte Datenqualitätsübertragung - besonders unter Verwendung der automatischen Fenstergrößenadaption - sehr gute Abschätzungen der Datenqualität einzelner Tupel erlaubt. Der zweite Teil widmet sich der quantitativen und qualitativen Überprüfung der entwickelten Datenqualitätsalgebra. Die Auswirkungen der Verarbeitungsoperatoren auf die resultierende Datenqualität wurden am Beispiel der vorausschauenden Wartung untersucht. Es konnte gezeigt werden, dass die entwickelte Algebra eine sehr gute Abschätzung der Ergebnisdatenqualität liefert. Im dritten Teil wurden die Verfahren der Datenqualitätsverbesserung validiert. Im Vergleich mit bestehenden Verfahren kann der datenqualitätsgesteuerte Lastausgleich die Korrektheit von Aggregationen und die Vollständigkeit von Verbundmengen von Wetterdaten er-

höhen. Des Weiteren wird am Beispiel der Produktionskontrolle gezeigt, dass auch die Optimierung der Sensordatenverarbeitung zur Verbesserung der Ergebnisqualität führt.

Sowohl die Datenqualitätsmodellierung als auch die Datenqualitätsalgebra wurden generisch konzipiert, so dass sie um beliebige DQ-Dimensionen bzw. Operatoren der Datenverarbeitung erweitert werden können. Die entwickelten Konzepte sind dadurch auch außerhalb der Smart-Items-Domäne einsetzbar. Neue Anwendungsfelder sind zum Beispiel die Analyse von Börsenkursen, Netzwerklasten, Verkehrsaufkommen oder Geschäftszahlen wie Umsatz einer Filiale oder Absatzzahlen eines bestimmten Produktes.

Das Datenqualitätsmodell DQMx kann außerdem um Operatoren zur Modellierung von Datenübertragungsfehlern in verschiedenen Übertragungsmedien und -protokollen erweitert werden. Nur so kann die verteilte Infrastruktur typischer Smart-Item-Systeme vollständig abgebildet werden. Datenverluste müssen durch Anpassung der Vollständigkeit, verrauschte Datenübertragungen durch Einfügen zusätzlicher Genauigkeits- und Konfidenzfehler modelliert werden. Die Auswahl des spezifischen Datenübertragungskanals stellt dann einen neuen Konfigurationsparameter der Optimierung der Datenverarbeitung dar. Die Zuverlässigkeit der Übertragung muss mit Dienstqualitätsfaktoren, wie Latenzzeit und Datendurchsatz, abgewogen werden.

Die Optimierung der Datenverarbeitung verbessert die Datenqualität von Verarbeitungsergebnissen, indem alle beteiligten Operatoren optimal konfiguriert werden. Die Struktur des Verarbeitungsgraphen wird nicht verändert. In weiterführenden Arbeiten muss untersucht werden, wie die qualitätsgesteuerte Optimierung mit Regeln der traditionellen Anfrageoptimierung in Datenbank- und Datenstromsystemen verknüpft werden kann. Heuristiken (z.B. das frühe Ausführen von Projektionen und Selektionen) sowie statistische Informationen (z.B. Histogramme oder Selektivitätsfaktoren) müssen in die qualitätsgesteuerte Optimierung einbezogen werden. So kann neben der optimalen Parametrierung auch die optimale Struktur der Datenverarbeitung gefunden werden.

Neben der traditionellen Anfrageverarbeitung können Data-Mining-Verfahren auf Sensordaten angewendet werden, um Ähnlichkeiten oder Zusammenhänge zu erkennen. Auch hier müssen Datenqualitätsinformationen berücksichtigt werden. Sie können einerseits als Parameter des unsicheren Data-Mining verwendet werden, andererseits muss die Datenqualität der Mining-Ergebnisse bestimmt werden. Die relevanten Mining-Methoden müssen hinsichtlich ihres Einflusses auf die verschiedenen Datenqualitätsdimensionen untersucht werden, um eine Datenqualitätsalgebra des Data-Mining zu entwerfen.

Der im Rahmen der vorliegenden Arbeit entwickelte Prototyp QPIPEZ unterstützt die Visualisierung von numerischen Daten und Datenqualitätsinformationen in dynamischen Diagrammen. Dieser Visualisierungsansatz kann in zukünftigen Arbeiten durch Konzepte der visuellen Datenanalyse für verschiedene Anwendungskontexte und Datenqualitätsdimensionen erweitert und optimiert werden.

Zusammenfassend lässt sich feststellen, dass die zu Beginn aufgestellten Forschungsfragen in der vorliegenden Arbeit in vollem Maße beantwortet wurden. Das Datenqualitätsmodell DQMx vereinigt Strukturen zur effizienten Verwaltung von Datenqualitätsinformationen mit einer umfassenden Datenqualitätsalgebra. Des Weiteren wurden Methoden zur Datenqualitätsverbesserung entwickelt, die die Integration von Nutzeranforderungen ermöglichen. Die Datenqualitätsvisualisierung vervollständigt das entwickelte System des Datenqualitätsmanagements.

A

Anwendungsszenarien der Validierung

A.1. Vorausschauende Wartung eines Hydrauliksystems

Die vorausschauende Wartungsplanung wird anhand eines hydraulisch gesteuerten Baggerarms simuliert. Zur Steuerung der Armbewegung in alle drei Raumrichtungen und zur Bewegung der Baggerschaufel werden vier unabhängige Hydraulikzylindersysteme benötigt.

Abbildung A.1 zeigt das Schema der Sensordatenverarbeitung. Jedes Zylindersystem wird mit Hilfe der Drucksensoren in den Hoch- und Niederdruckkammern p_1, p_3, p_5, p_7 bzw. p_2, p_4, p_6, p_8 überprüft. Der Druckverlust wird als Differenz der Hoch- und Niederdruckmessung (z.B. $p_1 - p_2$) bestimmt. Eine durchschnittliche Druckdifferenz über $200bar$ lässt auf eine Blockierung, eine Differenz unter $40bar$ auf ein Dichtungsleck schließen, so dass eine Warnung ausgegeben und ein Wartungsauftrag ausgelöst werden muss. Außerdem wird der Druckverlustanstieg bzw. -abfall mit Hilfe der Anstiegsberechnung überwacht, um besonders kritische Situationen im Voraus erkennen zu können. Steigt oder sinkt der Druckverlust um mehr als $0,5bar$ pro Sekunde muss ebenfalls eine Warnung erfolgen.

Des Weiteren werden Sensoren zur Messung des Wassergehalts W des Hydrauliköls, der Öltemperatur T, der Viskosität V und der Partikelverschmutzung P verwendet, um das Alter des Öls zu bestimmen [DB07]. Dabei sind Temperaturen über 60°C sowie die maximale Partikelverschmutzung interessant. Anschließend werden die vier Messungen gewichtet, um unterschiedliche Wertebereiche auf das Intervall $[0; 1]$ zu normalisieren. Um die verbleibende Lebensdauer zu bestimmen, werden die Differenzen jedes Mess-

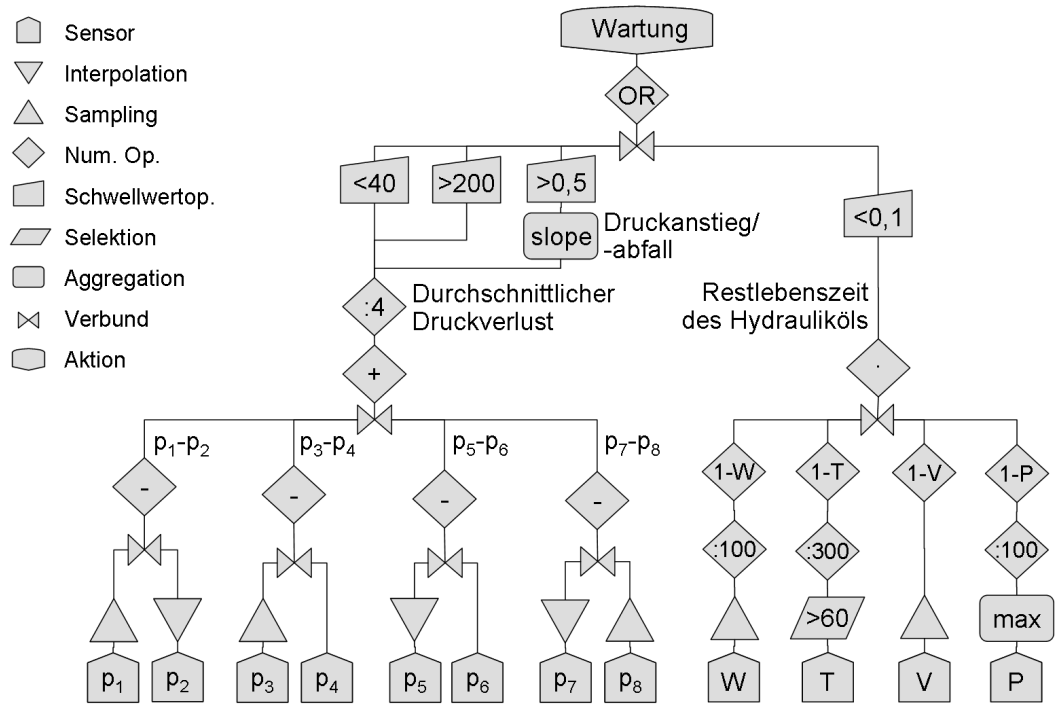


Abbildung A.1.: Datenverarbeitung der Hydrauliküberwachung

wertes zur vollen Lebensdauer $(1 - x)$ gebildet und anschließend multipliziert. Sinkt die verbleibende Lebenszeit unter 10% muss ein Ölwechsel eingeplant werden.

Anhand realer Messungen eines hydraulischen Baggerarms am Institut für Fluidtechnik der TU Dresden konnten Eigenschaften aller benötigter Sensordatenströme abgeleitet werden. Für jeden Messwert wurden Mittelwert und Varianz zu Beginn und am Ende der Messreihen bestimmt, die über mehrere Tage aufgezeichnet wurden (siehe Tabelle A.1). Die Druckmessungen werden in *bar*, der Wassergehalt in %, die Temperatur in °C und die Viskosität in Pascalsekunden ($Pa \cdot s$) angegeben. Die Partikelverschmutzung wird mit Hilfe der Reinheitsklassen durch Zählen der Partikel mit einem Durchmesser kleiner als $2\mu m, 5\mu m$ und $15\mu m$ bestimmt.

Darüber hinaus wurden die Güteklassen und die Maximalwertgrenze der verwendeten Sensoren festgehalten, die den systematischen Messfehler bestimmen (siehe Tabelle A.2). Zur Simulation von Sensorausfällen wurde die Ausfallrate, das heißt die durchschnittliche Anzahl fehlender Messwerte auf 100 Datentupel, aufgezeichnet. Auf dieser Basis konnten einerseits die „wahren“ Sensordatenströme, andererseits die verfälschten Datenströme, die realen Messungen entsprechen, simuliert werden (Vergleich Abschnitt 7.3).

	Mittelwert		Varianz	
Messwert	**Beginn**	**Ende**	**Beginn**	**Ende**
Hochdruck	165	12,93	60	6,90
Niederdruck	22	1,67	34	1,82
Wassergehalt	6	1,82	24,5	3,20
Öltemperatur	40	0,81	63	0,76
Viskosität	0,051	0,0014	0,082	0,0034
Partikelverschm.	12,33	4,19	16,41	4,25

Tabelle A.1.: Eigenschaften der Hydrauliksensoren

Messwert	**Güteklasse**	**Maximalwert**	**Sys. Messfehler**	**Ausfallrate**
Hochdruck	1	250	2,5	0,7
Niederdruck	0,5	100	0,5	0,6
Wassergehalt	5	100	5,0	1,6
Öltemperatur	1	100	1,0	0,2
Viskosität	2,5	0,1	0,0025	1,3
Partikelverschm.	—	20	0,5	2,8

Tabelle A.2.: Qualitätskriterien der Hydrauliksensoren

A.2. Qualitätskontrolle in der Kontaktlinsenproduktion

In [EA90] werden Messungen von 25 Kontaktlinsen zur Verfügung gestellt. Die Stärke der Kontaktlinsenmitte, des Linsenrandes sowie die Axialverschiebung wurden ermittelt. Tabelle A.3 zeigt einen Ausschnitt dieser Messungen (in Millimeter).

Die optimale Stärke der Linsendicke liegt bei $th_c = 0,4mm$. Der Rand einer Kontaktlinse sollte $th_e = 0,35mm$ betragen. Bei beiden Parametern sind Abweichungen von $\pm 0,01mm$ erlaubt. Die Axialverschiebung darf $ax = 0,0025mm$ nicht übersteigen. Um eine Annahme der Kontaktlinsenqualität zu simulieren, wurde ein zufälliger, unregelmäßiger Anstieg der Stärken der Linsenmitte sowie von vier Randmessungen modelliert. Darüber hinaus steigen die Varianzen der Sensormessungen. Tabelle A.4 fasst die Eigenschaften der simulierten Sensordatenströme der Kontaktlinsenkontrolle zusammen.

Linsennummer	Stärke d. Mitte	Stärke d. Randes	Axialverschiebung
1	0,3978	0,3454	0,0009
2	0,4019	0,3507	0,0007
3	0,4031	0,3478	0,0012
4	0,4044	0,3430	0,0011
5	0,3984	0,3519	0,0015
6	0,3972	0,3496	0,0018
7	0,3981	0,3478	0,0017
8	0,3947	0,3599	0,0022
9	0,4012	0,3472	0,0014
10	0,4043	0,3512	0,0018

Tabelle A.3.: Ausschnitt der realen Kontaktlinsendaten

	Mittelwert		Varianz	
Messwert	Beginn	Ende	Beginn	Ende
Stärke d. Mitte	0,4	$1,12 \cdot 10^{-5}$	0,437	$1,52 \cdot 10^{-5}$
Randstärke 1	0,35	$1,80 \cdot 10^{-5}$	0,362	$1,70 \cdot 10^{-5}$
Randstärke 2	0,35	$2,10 \cdot 10^{-5}$	0,380	$2,04 \cdot 10^{-5}$
Randstärke 3	0,35	$1,45 \cdot 10^{-5}$	0,352	$1,85 \cdot 10^{-5}$
Randstärke 4	0,35	$1,67 \cdot 10^{-5}$	0,379	$2,12 \cdot 10^{-5}$
Axialverschiebung	0,001	$1,56 \cdot 10^{-7}$	0,001	$1,565 \cdot 10^{-7}$

Tabelle A.4.: Eigenschaften der simulierten Sensordatenströme

A.3. Analyse von Wetterdaten

Der „Edited Synoptic Cloud Report“ [HWL08] umfasst Wetterdaten von Dezember 1981 bis November 1991, aufgezeichnet von Wetterstationen, die in der ganzen Welt zu Lande und auf Schiffen verteilt sind. Eine Vielzahl an Datensammlungen wurde zusammengetragen, vorverarbeitet und integriert, um Wetterdaten vor allem für die Analyse von Wolkenbildung und -verteilung zu gewinnen.

Das Archiv umfasst 240 Dateien, aufgeteilt in Monate und Land- oder Schiffswetterstationen. Die integrierte Wetterdatensammlung enthält 124 Millionen Datensätze von Landstationen und 15 Millionen Datensätze von Schiffen. Jeder Datensatz besteht aus 56 Zeichen. Tabelle A.5 beschreibt die Formatierung und Datenaufteilung. In Abbildung A.2 ist ein Ausschnitt der Wetterdaten vom 1. Juni 1990 dargestellt.

Meswert	Abkürzung	Byte	Minimum	Maximum
Jahr,Monat,Tag,Stunde	yr,mn,dy,hr	8	81120100	91113023
Helligkeit	IB	1	0	1
Breitengrad x100	LAT	5	-9000	9000
Längengrad x100	LON	5	0	36000
Nr. der Messstation	ID	5	01000	98999
Land/Meer	LO	1	1	2
Aktuelles Wetter	ww	2	-1	99
Gesamtwolkenbedeckung	N	1	0	8
Anteil niedriger Wolken	Nh	2	-1	8
Höhe niedriger Wolken	h	2	-1	9
Typus niedriger Wolken	CL	2	-1	11
Typus mittlerer Wolken	CM	2	-1	12
Typus hoher Wolken	CH	2	-1	9
Anteil mittlerer Wolken x100	AM	3	0	900
Anteil hoher Wolken x100	AH	3	0	900
Unüberd. Anteil mittlerer Wolken	UM	1	0	9
Unüberdeckt. Anteil hoher Wolken	UH	1	0	9
Wolkengeschwindigkeit	IC	2	0	9
Sonnenstand (Grad x10)	SA	4	-900	900
Relative Mondleuchtstärke x100	RI	4	-110	117

Tabelle A.5.: Formatierung der Wetterdatensätze

```
        I               L                   UU
yrmndyhrBLAT   LON   ID   OWWNNhh CLCMCHAM AH MHICSA  RI
900601031-662411053896111 21 1 5 5 0 8  0  000 0   0  -3
900601031-666614002896421718 6 7 511-180090020 7  12  -1
900601030-6759 6288895641 11 1 7 0 7 0100  010 0-135  -5
900601030-681229287890661-18 3 5 5 7-180090050 0-419   1
900601030-6857 7798895711157 2 5 5 2 890090099 0 -92  -4
900601030-6899 3958895321738 8 0 010-180090080 1-222  -4
900601030-703135765890011-11 0 9 0 0 8  010001 0-354  -2
900601030-778632538890341-10 0-1 0 0 0  0  000 0-338   0
900601061 825029767710821458 8 011-1-190090000 1 153  -1
900601061 8082 4745200341717 7 3 710-170090000 8 286  -2
900601061 8062 5805200461718 6 2 512-180090020 7 299  -1
900601061 8037 5292200491717 7 2 510-170090000 8 296  -2
900601061 8013 3675200261227 7 4 5-1-190090000 0 277  -2
```

Abbildung A.2.: Ausschnitt der verwendeten Wetterdaten

Die Wetterdaten werden in Aggregations- und Verbundanfragen verarbeitet, für die im Folgenden zwei Beispiele beschrieben werden. Abbildung A.3a zeigt den Anfragegraphen zur Berechnung der durchschnittlichen Wolkendichte über San Francisco (siehe Abschnitt 7.4.1). Zuerst wird eine Projektion der Attribute „Jahr,Monat,Tag,Stunde“, „Breitengrad“, „Breitengrad“ und „Gesamtwolkenbedeckung“ vorgenommen. Anschließend werden alle Wettermessungen des Raumes San Francisco selektiert. Zum Schluss wird die Wolkendichte berechnet und in die Zieldatenbank geschrieben.

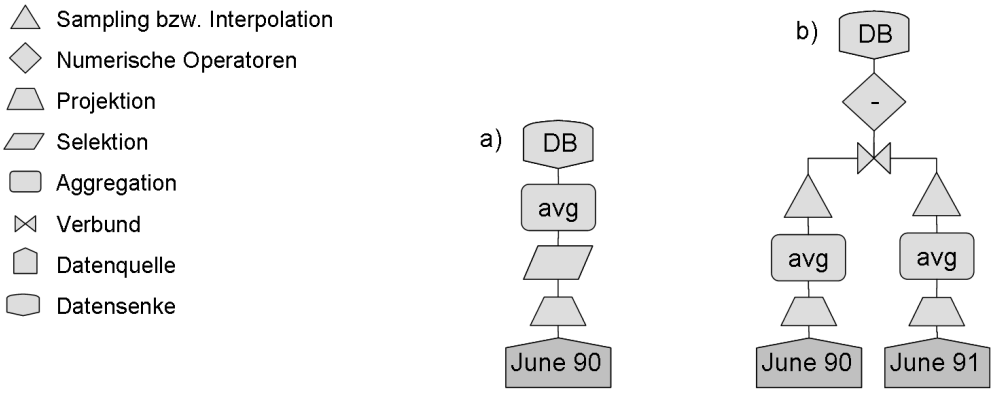


Abbildung A.3.: Verarbeitung der Wetterdaten

Abbildung A.3b zeigt ein Beispiel des qualitätsgesteuerten Lastausgleichs von Verbundanfragen. Wieder wird zuerst die relevante Attributmenge (z.B. „Jahr,Monat,Tag,Stunde“, „Nr. der Messstation“, „Aktuelles Wetter“ und „Gesamtwolkenbedeckung“) durch eine Projektion ausgelesen. Dann werden die Messungen jeder Wetterstation über je 24 Stunden gemittelt und auf Basis der Messzeitstempel verknüpft. Die Differenzen des aktuellen Wetters und der Wolkenbedeckung bestimmen die Wetterentwicklung und werden in der Zieldatenbank abgelegt.

Formelzeichen

D	Datenstrom
m	Länge des Datenstroms
j	Index der Datenstromtupel
A	Attribut
n	Anzahl der Attribute
i	Index der Attribute, Hilfsindex
t	Zeitstempel eines Datenstromtupels
x	Datenwert eines Attributs eines Datenstromtupels
$\hat{x}$	wahrer Wert der physikalischen Größe
Q	Menge der Datenqualitätsdimensionen
ϑ	Anzahl der propagierten DQ-Dimensionen
q	Spezifische DQ-Dimension
Δq	Datenqualitätsfehler
δq	geschätzter Datenqualitätsfehler
a	Genauigkeit
ϵ	Konfidenz
c	Vollständigkeit
d	Datenmenge
u	Aktualität
θ	Gesamtdatenqualität
w	Datenqualitätsfenster, Index zur Kennzeichnung der Fensterdatenqualität
κ	Anzahl der Datenqualitätsfenster
k	Index der Datenqualitätsfenster
ω	Fenstergröße
t_b	Startzeitpunkt des Datenqualitätsfensters
t_e	Endzeitpunkt des Datenqualitätsfensters
τ	Traditionelles Datenstromtupel
τ_q	Datenqualitätstupel
τ_c	Qualitätskontrolltupel

V	Datenstromvolumen
r	Datenstromrate
r_c	Qualitätskontrollrate
r_{sa}	Samplingrate
r_{in}	Interpolationsrate
r_{se}	Selektionsrate
r_{LS}	Load-Shedding-Rate
ϵ^+	Samplingfehler
p	Konfidenzwahrscheinlichkeit
ρ	(1-p/2)-Quantile der Normalverteilung
μ	Mittelwert
σ^2	Varianz
s^2	Stichprobenvarianz
Φ	Wahrscheinlichkeitsdichtefunktion der Normalverteilung
z	normalisierter geschätzter Datenqualitätsfehler
$EWSA$	Prozesskontrollfunktion
β	Empfindlichkeitsfaktor der Prozesskontrollfunktion
$s_{\delta q}$	Schwellwert des akzeptierten DQ-Fehlers
I	Interessantheit einer Datenstrompartition
s_I	Schwellwert der Interessantheit
I_{max}	maximale Interessantheit
X	Rohdatenmenge, Tupelmenge
Y	Ergebnismenge
F	Funktion der Datenverarbeitung
F^{DQ}	Funktion der Datenqualitätsverarbeitung
O	Operatorenmenge der Datenverarbeitung
O^{DQ}	Operatorenmenge der Datenqualitätsverarbeitung
o	Operator der Datenverarbeitung
o^{DQ}	Datenqualitätsoperator
Ω	Datenqualitätsalgebra
COV	Kovarianzmatrix
$cov(A, B)$	Kovarianz zweier Attribute A und B
ς	Pearson-Korrelationskoeffizienten
l	Gruppengröße
η	Synopsengröße
syn	Index zur Kennzeichnung der Synopsenqualität
α	Korrektheit
rec	Recall
int	Integrität

b	Datenqualitätsschranke
$depth$	Tiefe eines Verarbeitungsgraphen
dim	Dimension des Optimierungsproblems
ζ	Länge der zur Optimierung genutzten Datenstrompartition
f_{single}	monokriterielle Zielfunktion
f_{multi}	multikriterielle Zielfunktion
rel_dev	relative Fehlerabweichung

Abkürzungsverzeichnis

CQL	Continuous Query Language
CWM	Common Warehouse Metamodel
DBMS	Datenbank-Management-System
DDL	Data Definition Language
DML	Data Manipulation Language
DQ	Datenqualität
DQL	Data Query Language
DQLS	Datenqualitätsgesteuerter Lastausgleich
DQMx	Data Quality Model Extension
DSMS	Datenstrom-Management-System
DWH	Data Warehouse
ER	Entity Relationship
ETL	Extraction, Transformation, Loading
IP	Informationsprodukt
IS	Informationssystem
IT	Informationstechnologie
QPIPEZ	Quality PIPEs & VisualiZation
RFID	Radio Frequency Identification
SQL	Structured Query Language
TDQM	Total Data Quality Management
XML	eXtensible Markup Language

Abbildungsverzeichnis

Tabellenverzeichnis

Algorithmenverzeichnis

Literaturverzeichnis

[AA87] AGMON, NACHMAN und NIV AHITUV: *Assessing data reliability in an information system*. Journal on Management of Information Systems, 4(2):33–44, 1987. 21

[ABB+03] ARASU, ARVIND, BRIAN BABCOCK, SHIVNATH BABU, MAYUR DATAR, KEITH ITO, ITARU NISHIZAWA, JUSTIN ROSENSTEIN und JENNIFER WIDOM: *The STREAM Group. STREAM: The Stanford stream data manager*. IEEE Data Engineering Bulletin, 26(1), 2003. 16

[ABW06] ARASU, ARVIND, SHIVNATH BABU und JENNIFER WIDOM: *The CQL continuous query language: semantic foundations and query execution*. The VLDB Journal, 15(2):121–142, 2006. 16

[ACc+03a] ABADI, D., D. CARNEY, U. ÇETINTEMEL, M. CHERNIACK, C. CONVEY, C. ERWIN, E. GALVEZ, M. HATOUN, J. HWANG, A. MASKEY, A. RASIN, A. SINGER, M. STONEBRAKER, N. TATBUL, Y. XING, R.YAN und S. ZDONIK: *Aurora: A Data Stream Management System (Demonstration)*. In: *Proceedings of the 22nd ACM SIGMOD International Conference on Management of Data*, San Diego, CA, June 2003. 15

[ACc+03b] ABADI, D., D. CARNEY, U. ÇETINTEMEL, M. CHERNIACK, C. CONVEY, S. LEE, M. STONEBRAKER, N. TATBUL und S. ZDONIK: *Aurora: A New Model and Architecture for Data Stream Management*. Special Issue on Best Papers of VLDB 2002, 12(2):120–139, 2003. 15, 54, 55

[AGB08] ASKIRA GELMAN, IRIT und ANTHONY L. BARLETTA: *A "quick and dirty"website data quality indicator*. In: *Proceeding of the 2nd ACM workshop on Information credibility on the web*, Seiten 43–46, Napa Valley, California, USA, 2008. 51

[AN07] AHUJA, AMIT und YIU-KAI NG: *A Dynamic Attribute-Based Load Shedding Scheme for Data Stream Management Systems*. In: *Proceedings of the 2nd International Conference on Digital Telecommunications*, Seiten 30–42, Washington, DC, USA, 2007. 55

[Apa09] APACHE SOFTWARE FOUNDATION: *Apache Derby*, 2009. http://db.apache.org/derby, erfolgreicher Zugriff: April 2009. 150

[App04] APPLETON, K.: *Representing uncertainty in geovisualisation*. In: *Proceedings of*

the European Science Foundation conference on Geovisualization, 2004. 50

[Arn92] ARNOLD, STEPHEN E.: *Information manufacturing: the road to database quality*. Database, 15(5):32–39, 1992. 36

[AS64] ABRAMOWIZ, MILTON und IRENE A STEGUN.: *Handbook of Mathematical Functions*. National Bureau of Standards, 1964. 77

[AW04] ARASU, ARVIND und JENNIFER WIDOM: *Resource sharing in continuous sliding-window aggregates*. In: *Proceedings of the 30th International Conference on Very Large Data Bases*, Seiten 336–347, Toronto, Canada, 2004. 18

[Bai83] BAILEY, ROBERT W.: *Human Error in Computer Systems*. Prentice Hall PTR, Upper Saddle River, NJ, USA, 1983. 12

[Bas90] BASCH, REVA: *Measuring the quality of the data: Report on the fourth annual SCOUG retreat*. In: *Database Searcher*, Seiten 18–23, 1990. 21

[BBD+02] BABCOCK, BRIAN, SHIVNATH BABU, MAYUR DATAR, RAJEEV MOTWANI und JENNIFER WIDOM: *Models and issues in data stream systems*. In: *Proceedings of the 21st ACM SIGMOD-SIGACT-SIGART symposium on Principles Of Database Systems*, Seiten 1–16, Madison, Wisconsin, 2002. 15

[BCSW06] BALLOU, DONALD P., INDUSHOBHA N. CHENGALUR-SMITH und RICHARD Y. WANG: *Sample-Based Quality Estimation of Query Results in Relational Database Environments*. IEEE Transitions on Knowledge and Data Engineering, 18(5):639–650, 2006. 40, 42

[BDJ+05] BURDICK, DOUG, PRASAD M. DESHPANDE, T. S. JAYRAM, RAGHU RAMAKRISHNAN und SHIVAKUMAR VAITHYANATHAN: *OLAP over Uncertain and Imprecise Data*. In: *Proceedings of the 31st International Conference on Very Large Data Bases*, Seiten 970–981, Trondheim, Norway, 2005. 44

[BDJ+06] BURDICK, DOUG, PRASAD M. DESHPANDE, T. S. JAYRAM, RAGHU RAMAKRISHNAN und SHIVAKUMAR VAITHYANATHAN: *Efficient allocation algorithms for OLAP over imprecise data*. In: *Proceedings of the 32nd International Conference on Very Large Data Bases*, Seiten 391–402, Seoul, Korea, 2006. 44

[BDM04] BABCOCK, BRIAN, MAYUR DATAR und RAJEEV MOTWANI: *Load Shedding for Aggregation Queries over Data Streams*. In: *Proceedings of the 20th IEEE International Conference on Data Engineering*, Seiten 350–361, Los Alamitos, CA, USA, 2004. 54, 55, 121

[BG05a] BARATEIRO, JOSÉ und HELENA GALHARDAS: *A Survey of Data Quality Tools*. Datenbank-Spektrum, 14:15–21, 2005. 48

[BG05b] BEYDEDA, SAMI und VOLKER GRUHN: *Model-Driven Software Development*. Springer-Verlag New York, Inc., Secaucus, NJ, USA, 2005. 172

[BJ06] BERANDER, PATRIK und PER JÖNSSON: *A goal question metric based approach for efficient measurement framework definition*. In: *Proceedings of the 5th*

ACM/IEEE International Symposium on Empirical Software Engineering, Seiten 316–325, Rio de Janeiro, Brazil, 2006. 48

[BJ07] BRAZHNIK, OLGA und JOHN F. JONES: *Anatomy of data integration*. Journal of Biomedical Informatics, 40(3):252–269, 2007. 50

[BK03] BROWN, MARVIN L. und JOHN F. KROS: *The impact of missing data on data mining*. Seiten 174–198, 2003. 14

[BNQ06] BISWAS, JIT, FELIX NAUMANN und QIANG QIU: *Assessing the Completeness of Sensor Data*. In: *Proceedings of the 11th International Conference on Database Systems for Advanced Applications*, Seiten 717–732, 2006. 57, 89, 114

[BP98] BRIN, SERGEY und LAWRENCE PAGE: *The anatomy of a large-scale hypertextual Web search engine*. In: *WWW'98: Proceedings of the seventh international conference on World Wide Web*, Seiten 107–117, Brisbane, Australia, 1998. 52

[BP03] BALLOU, DONALD P. und HAROLD L. PAZER: *Modeling Completeness versus Consistency Tradeoffs in Information Decision Contexts*. IEEE Transitions on Knowledge and Data Engineering, 15(1):240–243, 2003. 40, 43

[BS06] BATINI, CARLO und MONICA SCANNAPIECO: *Data Quality: Concepts, Methodologies and Techniques (Data-Centric Systems and Applications)*. Springer-Verlag New York, Inc., Secaucus, NJ, USA, 2006. 43, 193

[BT99] BALLOU, DONALD P. und GIRI KUMAR TAYI: *Enhancing Data Quality in Data Warehouse Environments*. Communications of the ACM, 42(1):73–78, 1999. 46

[CC03] CHANDRASEKARAN, S. und O. COOPER: *Telegraph CQ: Continuous Dataflow Processing for an Uncertain World*. In: *Proceedings of the 1st Biennial Conference on Innovative Data Systems Research*, Asilomar, CA, 2003. 16

[CcC+02] CARNEY, DONALD, UGUR ÇETINTEMEL, MITCH CHERNIACK, CHRISTIAN CONVEY, SANGDON LEE, GREG SEIDMAN, MICHAEL STONEBRAKER, NESIME TATBUL und STANLEY B. ZDONIK: *Monitoring Streams - A New Class of Data Management Applications*. In: *Proceedings of the 29th International Conference on Very Large Data Bases*, Seiten 215–226, 2002. 15

[CGJ+02] CRANOR, CHUCK, YUAN GAO, THEODORE JOHNSON, VLAIDSLAV SHKAPENYUK und OLIVER SPATSCHECK: *Gigascope: high performance network monitoring with an SQL interface*. In: *Proceedings of the 21st ACM SIGMOD International Conference on Management of Data*, Seiten 623–623, Madison, Wisconsin, 2002. 16

[CHK+03] CAMMERT, MICHAEL, CHRISTOPH HEINZ, JÜRGEN KRÄMER, MARTIN SCHNEIDER und BERNHARD SEEGER: *A Status Report on XXL - a Software Infrastructure for Efficient Query Processing*. IEEE Data Engineering Bulletin, 26(2):12–18, 2003. 16

[CJSS03a] CRANOR, CHARLES D., THEODORE JOHNSON, OLIVER SPATSCHECK und VLADISLAV SHKAPENYUK: *The Gigascope Stream Database*. IEEE Data Engineering Bulletin, 26(1):27–32, 2003. 16

[CJSS03b] CRANOR, CHUCK, THEODORE JOHNSON, OLIVER SPATASCHEK und VLADISLAV SHKAPENYUK: *Gigascope: a stream database for network applications*. In: *Proceedings of the 22nd ACM SIGMOD International Conference onManagement of Data*, Seiten 647–651, San Diego, California, 2003. 16

[CKR07] CURTIN, JOHN, ROBERT J. KAUFFMAN und FREDERICK J. RIGGINS: *Making the MOST out of RFID technology: a research agenda for the study of the adoption, usage and impact of RFID*. Information Technology and Management, 8(2):87–110, 2007. 56

[CPPS01] CALERO, CORAL, MARIO PIATTINI, CAROLINA PASCUAL und MANUEL SERRANO: *Towards Data Warehouse Quality Metrics*. In: *Design and Management of Data Warehouses*, Seite 2, 2001. 48

[CRK00] CHAFFIN, MARK, TODD ROBINSON und BRIAN KNIGHT: *Professional SQL Server 2000 Dts*. Wrox Press Ltd., Birmingham, UK, UK, 2000. 49

[CSBP99] CHENGALUR-SMITH, INDUSHOBHA N., DONALD P. BALLOU und HAROLD L. PAZER: *The Impact of Data Quality Information on Decision Making: An Exploratory Analysis*. IEEE Transitions on Knowledge and Data Engineering, 11(6):853–864, 1999. 32

[CW01] CUI, YINGWEI und JENNIFER WIDOM: *Lineage Tracing for General Data Warehouse Transformations*. The Journal on Very Large Data Bases, Seiten 471–480, 2001. 38, 43

[CZW98] CHEN, YING, QIANG ZHU und NENGBIN WANG: *Query processing with quality control in the World Wide Web*. World Wide Web, 1(4):241–255, 1998. 21, 51

[D'A95] D'AGOSTINI, G.: *Probability and Measurement Uncertainty in Physics - a Bayesian Primer*. 1995. 14

[Dat02] DATA WAREHOUSING INSTITUTE: *Data Quality and the Bottom Line: Achieving Business Success through a Commitment to High Quality Data*, 2002. http://www.dw-institute.com/research/ display.asp?id6064#RS. 3

[DB07] DIETER, M. und F. BAUER: *Hydac Condition Monitoring - from Sensor to System (german)*. O+P - Journal for Fluid Dynamics, 2007. 10, 181

[DeM82] DEMARCO, T.: *Software-Projektmanagement*. Yourdon Inc, Prentice Hall, München, 1982. 4

[DJMS02] DASU, TAMRAPARNI, THEODORE JOHNSON, S. MUTHUKRISHNAN und VLADISLAV SHKAPENYUK: *Mining database structure; or, how to build a data quality browser*. In: *Proceedings of the 21st ACM SIGMOD International Con-*

ference on Management of Data, Seiten 240–251, Madison, Wisconsin, 2002. 49

[DKO+06] DAI, BING TIAN, NICK KOUDAS, BENG CHIN OOI, DIVESH SRIVASTAVA und SURESH VENKATASUBRAMANIAN: *Column Heterogeneity as a Measure of Data Quality*. In: *CleanDB*, 2006. 50

[DNGM07] DANIEL, T. E., R. M. NEWMAN, E. I. GAURA und S. N. MOUNT: *Complex query processing in wireless sensor networks*. In: *Proceedings of the 2nd ACM Workshop on Performance Monitoring and Measurement of Heterogeneous Wireless and Wired Networks*, Seiten 53–60, Chania, Crete Island, Greece, 2007. 53

[EA90] EDGEMAN, RICK L. und SUSAN B. ATHEY: *Digidot Plots for Process Surveillance*. Quality Progress, 23(5):66–68, 1990. 166, 183

[EIV07] ELMAGARMID, AHMED K., PANAGIOTIS G. IPEIROTIS und VASSILIOS S. VERYKIOS: *Duplicate Record Detection: A Survey*. IEEE Transitions on Knowledge and Data Engineering, 19(1):1–16, 2007. 47

[ES07] EVEN, ADIR und GANESAN SHANKARANARAYANAN: *Utility-driven assessment of data quality*. SIGMIS Database, 38(2):75–93, 2007. 20, 32

[Fin06] FINKENZELLER, KLAUS: *RFID-Handbuch*. Hanser Fachbuchverlag, 4 Auflage, 2006. 56

[FK01] FISHER, CRAIG W. und BRUCE R. KINGMA: *Criticality of data quality as exemplified in two disasters*. Information Management, 39(2):109–116, 2001. 3

[FN09] FÜHRING, PAUL und FELIX NAUMANN: *Viqtor Projekt-Homepage*, 2009. http://www.hpi.uni-potsdam.de/naumann/projekte/viqtor, erfolgreicher Zugriff: April 2009. 50

[Geo72] GEOFFRION, A. M.: *Generalized Benders decomposition*. Journal of Optimization Theory and Applications, 10(4):237–260, 1972. 47

[GFS+01] GALHARDAS, HELENA, DANIELA FLORESCU, DENNIS SHASHA, ERIC SIMON und CRISTIAN-AUGUSTIN SAITA: *Improving Data Cleaning Quality Using a Data Lineage Facility*. In: *Design and Management of Data Warehouses*, Seite 3, 2001. 38, 43

[GFSS00a] GALHARDAS, HELENA, DANIELA FLORESCU, DENNIS SHASHA und ERIC SIMON: *AJAX: an extensible data cleaning tool*. SIGMOD Records, 29(2):590, 2000. 49

[GFSS00b] GALHARDAS, HELENA, DANIELA FLORESCU, DENNIS SHASHA und ERIC SIMON: *Declaratively Cleaning your Data using AJAX*. Journal Bases de Donnees Avancees, 2000. 49

[GH01a] GRIMMER, UDO und HOLGER HINRICHS: *Datenqualitätsmanagement mit Data-Mining-Unterstützung*. HMD - Praxis Wirtschaftsinform., 222, 2001. 47

[GH01b] GRIMMER, UDO und HOLGER HINRICHS: *A Methodological Approach to Data Quality Management Supported by Data Mining*. In: *Proceedings of the 6th International Conference on Information Quality*, Seiten 217–232, 2001. 47

[GLC05] GAO, MINGXIA, CHUNNIAN LIU und FURONG CHEN: *An Ontology Search Engine Based on Semantic Analysis*. In: *ICITA '05: Proceedings of the Third International Conference on Information Technology and Applications (ICITA'05) Volume 2*, Seiten 256–259, 2005. 52

[GO03] GOLAB, LUKASZ und M. TAMER ÖZSU: *Issues in data stream management*. SIGMOD Records, 32(2):5–14, 2003. 15

[Gol89] GOLDBERG, DAVID E.: *Genetic Algorithms in Search, Optimization, and Machine Learning*. Addison-Wesley Professional, 1989. 144

[Gra05] GRABE, MICHAEL: *Measurement Uncertainties in Science and Technology*. Springer Verlag, Berlin, 2005. 14

[GS05] GRIETHE, HENRIETTE und HEIDRUN SCHUMANN: *Visualizing Uncertainty for Improved Decision Making*. In: *Proceedings of the 4th International Conference on Perspectives in Business Informatics Research*, Skövde, Sweden, 2005. 50

[HA01] HINRICHS, HOLGER und THOMAS ADEN: *An ISO 9001: 2000 Compliant Quality Management System for Data Integration in Data Warehouse Systems*. In: *Design and Management of Data Warehouses*, Seite 1, 2001. 49

[Haa97] HAAS, PETER J.: *Large-Sample and Deterministic Confidence Intervals for Online Aggregation*. In: *Proceedings of the 9th International Conference on Scientific and Statistical Database Management*, Seiten 51–63, Olympia, WA, USA, 1997. 96, 157

[HEK05] HARTUNG, JOACHIM, BÄRBEL ELPELT und KARL-HEINZ KLÖSENER: *Statistik: Lehr- und Handbuch der angewandten Statistik*. Oldenbourg, 2005. 106

[Hin01] HINRICHS, HOLGER: *Datenqualitätsmanagement in Data Warehouse-Umgebungen*. In: *9th GI Fachtagung: Datenbanksysteme in Büro, Technik und Wissenschaft*, Seiten 187–206, 2001. 48

[Hin09] HINRICHS, HOLGER: *CLIQ - Intelligent Data Quality Management*, 2009. citeseer.ist.psu.edu/320504.html, erfolgreicher Zugriff: April 2009. 49

[HS98] HERNÁNDEZ, MAURICIO A. und SALVATORE J. STOLFO: *Real-world Data is Dirty: Data Cleansing and The Merge/Purge Problem*. Data Mining Knowledge Discovery, 2(1):9–37, 1998. 47

[HWL08] HAHN, C. J., S. G. WARREN und J. LONDON: *Edited synoptic cloud reports from ships and land stations over the globe, 1982-1991*, 2008. http://cdiac.esd.ornl.gov/ftp/ndp026b, erfolgreicher Zugriff: April 2009. 11, 162, 184

[Hyd09] HYDAC INTERNATIONAL: *Druckmeßumformer HDA 4400*, 2009.

http://www.hydac.de/de-de/produkte.html, erfolgreicher Zugriff: April 2009. 13

[IOB83] IVES, BLAKE, MARGRETHE H. OLSON und JACK J. BAROUDI: *The measurement of user information satisfaction*. Communications of the ACM, 26(10):785–793, 1983. 32

[JIM00] JONATHAN I. MALETIC, ANDRIAN MARCUS: *Data Cleansing: Beyond Integrity Analysis*. In: *Proceedings of the 5th International Conference on Information Quality*, Seiten 200–209, Massachusetts Institute of Technology, Boston, MA, USA, 2000. 46

[JIW95] JANG, YEONA, ALEXANDER T. ISHII und RICHARD Y. WANG: *A qualitative approach to automatic data quality judgment*. Journal of Organization Computation, 5(2):101–121, 1995. 23

[JJQV98] JARKE, MATTHIAS, MANFRED A. JEUSFELD, CHRISTOPH QUIX und PANOS VASSILIADIS: *Architecture and Quality in Data Warehouses*. In: *Proceedings of the 10th International Conference on Advanced Information Systems Engineering*, Seiten 93–113, London, UK, 1998. 48

[Jur88] JURAN, JOSEPH M.: *Juran's quality control handbook*. McGraw-Hill, 4. ed Auflage, 1988. 20, 46

[JV97] JARKE, MATTHIAS und YANNIS VASSILIOU: *Data Warehouse Quality: A Review of the DWQ Project*. In: *Proceedings of the 2nd International Conference on Information Quality*, Seiten 299–313, 1997. 22, 48

[KCC+03] KRISHNAMURTHY, SAILESH, SIRISH CHANDRASEKARAN, OWEN COOPER, AMOL DESHPANDE, MICHAEL J. FRANKLIN, JOSEPH M. HELLERSTEIN, WEI HONG, SAMUEL MADDEN, FREDERICK REISS und MEHUL A. SHAH: *TelegraphCQ: An Architectural Status Report*. IEEE Data Engineering Bulletin, 26(1):11–18, 2003. 16

[KGH05] KÜBART, JOACHIM, UDO GRIMMER und JOCHEN HIPP: *Regelbasierte Ausreißersuche zur Datenqualitätsanalyse*. Datenbank-Spektrum, 14:22–28, 2005. 47

[KJ05] KIENCKE, U. und H. JÄKEL: *Signale und Systeme*. Oldenbourg Verlag, 2005. 81, 95, 100

[KKPP98] KAPLAN, DAVID, RAMAYYA KRISHNAN, REMA PADMAN und JAMES PETERS: *Assessing data quality in accounting information systems*. Communications of the ACM, 41(2):72–78, 1998. 45

[KNV02] KANG, J., J. F. NAUGHTON und S. D. VIGLAS: *Evaluating Window Joins Over Unbounded Streams*. In: *Proceedings of the 28th International Conference on Very Large Data Bases*, Seiten 341–352, Hong Kong, China, 2002. 18, 55, 130

[Kri78] KRIEBEL, C. H.: *Evaluating the Quality of Information Systems*. In: *Proceedings*

of the BIFOA Symposium, Seiten 114–128, Bensberg/Colonge, 1978. 21

[Kri79] KRIEBEL, C. H.: *Evaluating the Quality of Information Systems*. In: *Design and Implementation of Computer-Based Information Systems*, Seiten 29–43, The Netherlands, 1979. Sijthoff and Noordhoff International Publishers bV. 21

[KS04] KRAEMER, JÜRGEN und BERNHARD SEEGER: *PIPES - A Public Infrastructure for Processing and Exploring Streams*. In: WEIKUM, GERHARD, ARND CHRISTIAN KOENIG und STEFAN DESSLOCH (Herausgeber): *Proceedings of the 9th ACM SIGMOD International Conference on Management of Data*, Seiten 925–926, 2004. 16, 150

[KSW02] KAHN, BEVERLY K., DIANE M. STRONG und RICHARD Y. WANG: *Information quality benchmarks: product and service performance*. Communications of the ACM, 45(4):184–192, 2002. 45

[Lau86] LAUDON, KENNETH C.: *Data quality and due process in large interorganizational record systems*. Communications of the ACM, 29(1):4–11, 1986. 52

[LC02] LIU, LIPING und LAUREN CHI: *Evolutional Data Quality: A Theory-Specific View*. In: *Proceedings of the 7th International Conference on Information Quality*, Seiten 292–304, 2002. 23, 193

[Leh03] LEHNER, WOLFGANG: *Datenbanktechnologie für Data-Warehouse-Systeme : Konzepte und Methoden*. 1. Aufl Auflage, 2003. 44

[Len04] LENZERINI, MAURIZIO: *Quality-aware peer-to-peer data integration*. In: *Proceedings of the 1st International Workshop on Information Quality in Information Systems*, Seite 1, 2004. 53

[LGJ03] LUEBBERS, DOMINIK, UDO GRIMMER und MATTHIAS JARKE: *Systematic development of data mining-based data quality tools*. In: *Proceedings of the 29th International Conference on Very Large Data Bases*, Seiten 548–559, Berlin, Germany, 2003. VLDB Endowment. 47

[LLLK99] LEE, MONG-LI, TOK WANG LING, HONGJUN LU und YEE TENG KO: *Cleansing Data for Mining and Warehousing*. In: *Proceedings of the 10th International Workshop on Database and Expert Systems Applications*, Seiten 751–760, Florence, Italy, 1999. 27, 47

[LN00] LESER, ULF und FELIX NAUMANN: *Query Planning with Information Quality Bounds*. In: *Proceedings of the 4th International Conference on Flexible Query Answering Systems*, Seiten 85–94, Warsaw, Poland, 2000. 51

[LSKW02] LEE, YANG W., DIANE M. STRONG, BEVERLY K. KAHN und RICHARD Y. WANG: *AIMQ: a methodology for information quality assessment*. Information Management, 40(2):133–146, 2002. 45

[LU90] LIEPINS, GUNAR E. und V. R. R. UPPULURI (Herausgeber): *Data quality control theory and pragmatics*. Marcel Dekker, Inc., New York, USA, 1990. 46

[LZZM07] LONGBO, ZHANG, LI ZHANHUAI, WANG ZHENYOU und YU MIN: *Semantic Load Shedding for Sliding Window Join-Aggregation Queries over Data Streams*. In: *Proceedings of the International Conference on Convergence Information Technology*, Seiten 2152–2155, 2007. 55

[Man07] MANGOLD, CHRISTOPH: *A survey and classification of semantic search approaches*. International Journal of Metadata, Semantics and Ontologies, 2(1):23–34, 2007. 52

[MDJ+97] MICHELIS, GIORGIO DE, ERIC DUBOIS, MATTHIAS JARKE, FLORIAN MATTHES, JOHN MYLOPOULOS, MIKE PAPAZOGLOU, KLAUS POHL, JOACHIM SCHMIDT, CARSON WOO und ERIC YU: *Cooperative Information Systems: A Manifesto*. In: PAPAZOGLOU, MIKE P. und GUNTHER SCHLAGETER (Herausgeber): *Cooperative Information System: Trends and Directions*. Academic Press, 1997. 44

[ME05] MISSIER, PAOLO und SUZANNE EMBURY: *Provider issues in quality-constrained data provisioning*. In: *Proceedings of the 2nd International Workshop on Information Quality in Information Systems*, Seiten 5–15, Baltimore, Maryland, 2005. 33

[MEG+06] MISSIER, PAOLO, SUZANNE EMBURY, MARK GREENWOOD, ALUN PREECE und BINLING JIN: *Quality views: capturing and exploiting the user perspective on data quality*. In: *Proceedings of the 32nd International Conference on Very Large Data Bases*, Seiten 977–988, Seoul, Korea, 2006. 32

[MFHH05] MADDEN, SAMUEL R., MICHAEL J. FRANKLIN, JOSEPH M. HELLERSTEIN und WEI HONG: *TinyDB: an acquisitional query processing system for sensor networks*. ACM Transitions on Database Systems, 30(1):122–173, 2005. 53

[Mic94] MICHALEWICZ, ZBIGNIEW: *Genetic Algorithms Plus Data Structures Equals Evolution Programs*. Springer-Verlag New York, Inc., Secaucus, NJ, USA, 1994. 144

[MMMR96] MENON, A., K. MEHROTRA, C. K. MOHAN und S. RANKA: *Characterization of a Class of Sigmoid Functions with Applications to Neural Networks*. Neural Networks, 9(5):819–835, 1996. 126

[MMS04] MURRENHOFF, H., T. MEINDORF und C. STAMMEN: *Online Condition Monitoring in Fluid Power Technology*. In: *IFK*, 2004. 10

[Moo06] MOON, YANG-SAE: *Efficient Stream Sequence Matching Algorithms for Handheld Devices on Time-series Stream Data*. In: *Proceedings of the 24th IASTED International Conference on Database and Applications*, Seiten 44–49, Innsbruck, Austria, 2006. 18

[Mor82] MOREY, RICHARD C.: *Estimating and improving the quality of information in a MIS*. Communications of the ACM, 25(5):337–342, 1982. 51

[MR97] MOTRO, A. und I. RAKOV: *Not All Answers are Equally Good: Estimating the*

Quality of Database Answers. Seiten 1–21, 1997. 39, 42

[MR98] MOTRO, AMIHAI und IGOR RAKOV: *Estimating the Quality of Databases*. Lecture Notes in Computer Science, 1495:298–307, 1998. 39, 42

[MRV00] MIHAILA, GEORGE A., LOUIQA RASCHID und MARIA-ESTHER VIDAL: *Using Quality of Data Metadata for Source Selection and Ranking*. In: *WebDB (Informal Proceedings)*, Seiten 93–98, 2000. 51

[MSO01] MYRTVEIT, INGUNN, ERIK STENSRUD und ULF H. OLSSON: *Analyzing Data Sets with Missing Data: An Empirical Evaluation of Imputation Methods and Likelihood-Based Methods*. IEEE Transitions Software Engineering, 27(11):999–1013, 2001. 14

[Mut05] MUTHUKRISHNAN, S.: *Data streams: algorithms and applications*. Foundation and Trends in Theoretic Computer Science, 1(2):117–236, 2005. 15

[Nau02] NAUMANN, FELIX: *Quality-driven query answering for integrated information systems*. Springer-Verlag New York, Inc., New York, NY, USA, 2002. 22, 51

[Nau07] NAUMANN, FELIX: *Datenqualität*. Informatik Spektrum, 30(1):27–31, 2007. 32

[Nau09] NAUMANN, FELIX: *HiQiIQ - High-Quality Information Querying*, 2009. http://www.hiqiq.de, erfolgreicher Zugriff: April 2009. 51

[NFL04] NAUMANN, FELIX, JOHANN-CHRISTOPH FREYTAG und ULF LESER: *Completeness of Integrated Information Sources*. Information Systems, 29(7):583–615, 2004. 40, 43

[NLF99] NAUMANN, FELIX, ULF LESER und JOHANN CHRISTOPH FREYTAG: *Quality-driven Integration of Heterogenous Information Systems*. In: *Proceedings of the 25th International Conference on Very Large Data Bases*, Seiten 447–458, San Francisco, CA, USA, 1999. Morgan Kaufmann Publishers Inc. 50

[Orr98] ORR, KEN: *Data Quality and Systems Theory*. Communications of the ACM, 41(2):66–71, 1998. 33

[OZH06] OJEWOLE, ADEGOKE, QIANG ZHU und WEN-CHI HOU: *Window join approximation over data streams with importance semantics*. In: *Proceedings of the 15th ACM International Conference on Information and Knowledge Management*, Seiten 112–121, Arlington, Virginia, USA, 2006. 55

[Pap06] PAPULA, L.: *Mathematische Formelsammlung fuer Ingenieure und Naturwissenschaftler (german)*. Vieweg Verlag, 2006. 14

[Par06] PARSSIAN, AMIR: *Managerial decision support with knowledge of accuracy and completeness of the relational aggregate functions*. Decision Support Systems, 42(3):1494–1502, 2006. 39, 42

[Pie04] PIERCE, ELIZABETH M.: *Assessing data quality with control matrices*. Commu-

nications of the ACM, 47(2):82–86, 2004. 36

[PM01] POOLE, JOHN und DAVID MELLOR: *Common Warehouse Metamodel: An Introduction to the Standard for Data Warehouse Integration*. John Wiley & Sons, Inc., New York, NY, USA, 2001. 64

[PMH95] PEARSON, J. MICHAEL, CYNTHIA S. MCCAHON und ROSS T. HIGHTOWER: *Total quality management: are information systems managers ready?* Information Management, 29(5):251–263, 1995. 33

[Pri04] PRICEWATERHOUSECOOPERS: *Data Quality Survey 2004*, 2004. http://www.pwc.com, erfolgreicher Zugriff: April 2009. 3

[PSJ99] PARSSIAN, AMIR, SUMIT SARKAR und VARGHESE S. JACOB: *Assessing data quality for information products*. In: *Proceedings of the 20th International Conference on Information Systems*, Seiten 428–433, Atlanta, GA, USA, 1999. Association for Information Systems. 38, 42

[PSJ02] PARSSIAN, AMIR, SUMIT SARKAR und VARGHESE S. JACOB: *Assessing Information Quality for the Composite Relational Operation Join*. In: *Proceedings of the 7th International Conference on Information Quality*, Seiten 225–237, 2002. 38, 42

[PSJ04] PARSSIAN, AMIR, SUMIT SARKAR und VARGHESE S. JACOB: *Assessing Data Quality for Information Products: Impact of Selection, Projection, and Cartesian Product*. Management Science, 50(7):967–982, 2004. 38, 42

[PSZ89] PATEL, N. R., R. L. SMITH und Z. B. ZABINSKY: *Pure adaptive search in Monte Carlo optimization*. Mathematical Programing, 43(3):317–328, 1989. 168

[QSB97] QI, HAIRONG, WESLEY E. SNYDER und GRIFF L. BILBRO: *Comparison of Mean Field Annealing and Multiresolution Analysis in Missing Data Estimation*. In: *Proceedings of the 3rd Asian Conference on Computer Vision*, Seiten 722–729, London, UK, 1997. Springer-Verlag. 14

[RD00] RAHM, ERHARD und HONG HAI DO: *Data Cleaning: Problems and Current Approaches*. IEEE Data Engineering Bulletin, 23(4):3–13, 2000. 22, 23, 193

[Red97] REDMAN, THOMAS C.: *Data Quality for the Information Age*. Artech House, Inc., Norwood, MA, USA, 1997. Foreword By-A. Blanton Godfrey. 21, 44

[Red98] REDMAN, THOMAS C.: *The impact of poor data quality on the typical enterprise*. ACM Communications, 41(2):79–82, 1998. 42

[RGT07] RÖHM, UWE, MOHAMED MEDHAT GABER und QUINCY TSE: *Enabling resource-awareness for in-network data processing in wireless sensor networks*. In: *Proceedings of the 19th International Conference on Australasian Database*, Seiten 107–114, Darlinghurst, Australia, Australia, 2007. Australian Computer Society, Inc. 53

[RH01] RAMAN, VIJAYSHANKAR und JOSEPH M. HELLERSTEIN: *Potter's Wheel: An Interactive Data Cleaning System*. In: *The VLDB Journal*, Seiten 381–390, 2001. 49

[RH05] REISS, FREDERICK und JOSEPH M. HELLERSTEIN: *Data Triage: An Adaptive Architecture for Load Shedding in TelegraphCQ*. In: *Proceedings of the 21st International Conference on Data Engineering*, Seiten 155–156, 2005. 55

[RL07] REN, QINGCHUN und QILIAN LIANG: *Energy and quality aware query processing in wireless sensor database systems*. Information Science, 177(10):2188–2205, 2007. 53

[RLSG07] RIERA-LEDESMA, JORGE und JUAN-JOSÉ SALAZAR-GONZÁLEZ: *A branch-and-cut algorithm for the continuous error localization problem in data cleaning*. Computational Operations Research, 34(9):2790–2804, 2007. 47

[RW95] REDDY, M. P. und RICHARD Y. WANG: *Estimating Data Accuracy in a Federated Database Environment*. In: *CISMOD*, Seiten 115–134, 1995. 44

[SAP09] SAP AG: *SAP BusinessObjects Portfolio - Connecting People, Information, and Businesses*, 2009. http://www.businessobjects.com/, erfolgreicher Zugriff: April 2009. 49

[SB04] SCANNAPIECO, MONICA und CARLO BATINI: *Completeness in the Relational Model: a Comprehensive Framework*. In: *Proceedings of the 9th International Conference on Information Quality*, Seiten 333–345, Cambridge, MA, USA, 2004. 37, 39, 42

[SBL04] SCHMIDT, SVEN, HENRIKE BERTHOLD und WOLFGANG LEHNER: *QStream: deterministic querying of data streams*. In: *Proceedings of the 30th International Conference onVery Large Data Bases*, Seiten 1365–1368, Toronto, Canada, 2004. 16

[SC06] SHANKARANARAYANAN, GANESAN und YU CAI: *Supporting data quality management in decision-making*. Decision Support Systems, 42(1):302–317, 2006. 36

[SEG05] SULO, RAJMONDA, STEPHEN EICK und ROBERT GROSSMAN: *DaVis: A tool for Visualizing Data Quality*. In: *Proceedings of the 11th IEEE Symposium on Information Visualization*, 2005. 50

[Sei08] SEIDLER, KATJA: *Datenqualität in relationalen Datenbankmanagementsystemen*. Belegarbeit, Technische Unitversität, Dresden, Germany, 2008. 72

[SFL05] SCHMIDT, SVEN, MARC FIEDLER und WOLFGANG LEHNER: *Source-aware Join Strategies of Sensor Data Streams*. In: *Proceedings of the 17th International Conference on Scientific and Statistical Database Management*, Seiten 123–132, Berkeley, CA, US, 2005. Lawrence Berkeley Laboratory. 19, 110

[Sha07] SHAW, W. T.: *Refinement of the Normal Quantile: Simple improvements to the*

Beasley Springer Moro method of simulating the Normal Distribution and a comparison with Acklams method and Wichuras AS241; King's College working paper, 2007. 126

[SJFW06] SARMA, ANISH DAS, SHAWN R. JEFFERY, MICHAEL J. FRANKLIN und JENNIFER WIDOM: *Estimating Data Stream Quality for Object-Detection Applications*. In: *Proceedings of the 3rd International Workshop on Information Quality in Information Systems*, Chicago, USA, 2006. 56

[SLSL05] SCHMIDT, SVEN, THOMAS LEGLER, SEBASTIAN SCHÄR und WOLFGANG LEHNER: *Robust real-time query processing with QStream*. In: *Proceedings of the 31st International Conference on Very Large Data Bases*, Seiten 1299–1301, Trondheim, Norway, 2005. 16

[SLW97] STRONG, DIANE M., YANG W. LEE und RICHARD Y. WANG: *Data Quality in Context*. Communications of the ACM, 40(5):103–110, 1997. 22

[SM07] SUHL, LEENA und TAIEB MELLOULI: *Optimierungssysteme: Modelle, Verfahren, Software, Anwendungen (Springer-Lehrbuch)*. Springer-Verlag New York, Inc., Secaucus, NJ, USA, 2007. 140

[Sri06] SRIVASTAVA, DIVESH: *The Bellman Data Quality Browser*. In: *CleanDB*, 2006. 49

[SS07] SONG, QINBAO und MARTIN SHEPPERD: *Missing Data Imputation Techniques*. International Journal on Business Intellice Data Mining, 2(3):261–291, 2007. 14

[Str08] STROPP, HERIBERT: *Physik für Studenten der Natur- und Ingenieurwissenschaften*. Hanser Fachbuch; Auflage: 14., aktualisierte Auflage (17. Januar 2008), 2008. 12

[Sva88] SVANKS, M. I.: *Integrity analysis: methods for automating data quality assurance*. Inf. Softw. Technol., 30(10):595–605, 1988. 46

[SVM+04a] SCANNAPIECO, MONICA, ANTONINO VIRGILLITO, CARLO MARCHETTI, MASSIMO MECELLA und ROBERTO BALDONI: *The architecture: a platform for exchanging and improving data quality in cooperative information systems*. Information Systems, 29(7):551–582, 2004. 52

[SVM+04b] SCANNAPIECO, MONICA, ANTONINO VIRGILLITO, CARLO MARCHETTI, MASSIMO MECELLA und ROBERTO BALDONI: *The DaQuinCIS architecture: a platform for exchanging and improving data quality in cooperative information systems*. Information Systems, 29(7):551–582, 2004. 35

[SW98] STOREY, V. C. und R. Y. WANG: *An Analysis of Quality Requirements in Database Design*. In: *Proceedings of the 3rd International Conference on Information Quality*, Seiten 64–87, 1998. 34, 62

[SW01] STOREY, V. C. und R. Y. WANG: *Extending the ER Model to Represent Data*

Quality Requirements. In: *Data Quality, Advances in Database Systems, Volume 23*, Seiten 37–48. Verlag Springer US, 2001. 34

[SW04] SRIVASTAVA, UTKARSH und JENNIFER WIDOM: *Memory-limited execution of windowed stream joins*. In: *Proceedings of the 30th International Conference on Very Large Data Bases*, Seiten 324–335, Toronto, Canada, 2004. 18, 55, 130

[SWZ00] SHANKARANARAYANAN, GANESAN, RICHARD Y. WANG und MOSTAPHA ZIAD: *IP-MAP: Representing the Manufacture of an Information Product*. In: *Proceedings of the 5th International Conference on Information Quality*, Seiten 1–16, 2000. 35

[TB98] TAYI, GIRI KUMAR und DONALD P. BALLOU: *Examining data quality*. Communications of the ACM, 41(2):54–57, 1998. 33

[TBH+04] TATBUL, NESIME, MARK BULLER, REED HOYT, STEVE MULLEN und STANLEY B. ZDONIK: *Confidence-based data management for personal area sensor networks*. In: *DMNS*, Seiten 24–31, 2004. 58

[TcZ+03] TATBUL, NESIME, UGUR ÇETINTEMEL, STANLEY B. ZDONIK, MITCH CHERNIACK und MICHAEL STONEBRAKER: *Load Shedding in a Data Stream Manager*. In: *Proceedings of the 29th International Conference on Very Large Data Bases*, Seiten 309–320, Berlin, Germany, 2003. 54

[TLPY06] TU, YI-CHENG, SONG LIU, SUNIL PRABHAKAR und BIN YAO: *Load shedding in stream databases: a control-based approach*. In: *Proceedings of the 32nd International Conference on Very Large Data Bases*, Seiten 787–798, Seoul, Korea, 2006. 54

[Tri09] TRILLIUM SOFTWARE: *Trillium Software Discovery*, 2009. http://www.trilliumsoftware.com/home/products/data-profiling/data-discovery.aspx, erfolgreicher Zugriff: April 2009. 48

[TZ06] TATBUL, NESIME und STAN ZDONIK: *Window-aware Load Shedding for Aggregation Queries over Data Streams*. In: *Proceedings of the 32nd International Conference on Very Large Data Bases*, Seiten 799–810, 2006. 54, 55

[UKN+99] UMAR, AMJAD, GEORGE KARABATIS, LINDA NESS, BRUCE HOROWITZ und AHMED ELMAGARDMID: *Enterprise Data Quality: A Pragmatic Approach*. Information Systems Frontiers, 1(3):279–301, 1999. 45

[VB99] VAN SOLINGEN, RINI und EGON BERGHOUT: *Goal/Question/Metric Method*. McGraw-Hill Education, 1999. 48

[VdP02] VIKTOR, HERNA L. und NIEK F. DU PLOOY: *Assessing and improving the quality of knowledge discovery data*. Seiten 198–205, 2002. 47

[Wan98] WANG, RICHARD Y.: *A product perspective on total data quality management*. Communications of the ACM, 41(2):58–65, 1998. 33

[Wei99] WEIKUM, GERHARD: *Towards Guaranteed Quality and Dependability of Infor-*

mation Services. In: *8th GI Fachtagung: Datenbanksysteme in Buero, Technik und Wissenschaft*, Seiten 379–409, 1999. 21

[Wiz09] WIZSOFT COMPANY: *WizRule*, 2009. http://www.wizsoft.com/, erfolgreicher Zugriff: April 2009. 49

[WM90] WANG, Y. RICHARD (YNG-YUH RICHARD) und STUART E. MADNICK: *A polygen model for heterogeneous database systems : the source tagging perspective*. Working papers 3119-90. CIS (Series) (Sl), Massachusetts Institute of Technology (MIT), Sloan School of Management, 1990. 35, 38, 42

[WS96] WANG, RICHARD Y. und DIANE M. STRONG: *Beyond accuracy: What Data Quality Means to Data Consumers*. Journal of Management Information Systems, 12(4):5–33, 1996. 22, 197

[WSF95] WANG, RICHARD Y., VEDA C. STOREY und CHRISTOPHER P. FIRTH: *A Framework for Analysis of Data Quality Research*. IEEE Transitions on Knowledge and Data Engineering, 7(4):623–640, 1995. 34, 37, 42, 62

[WW96] WAND, YAIR und RICHARD Y. WANG: *Anchoring Data Quality Dimensions in Ontological Foundations*. Communications of the ACM, 39(11):86–95, 1996. 45

[WZL01] WANG, R., W. ZIAD und Y. LEE: *Developing a Data Quality Algebra*. In: *Data Quality, Advances in Database Systems, Volume 23*, Seiten 63–77. Verlag Springer US, 2001. 38, 42

[Zel09] ZELL, ANDREAS: *JavaEva: A Java based framework for Evolutionary Algorithms*, 2009. http://www.ra.cs.uni-tuebingen.de/software/EvA2/, erfolgreicher Zugriff: April 2009. 150