CARL VON OSSIETZKY universität OLDENBURG

Fakultät II – Informatik, Wirtschafts- und Rechtswissenschaften
Department für Informatik

Qualitätssensitive Datenstromverarbeitung zur Erstellung von dynamischen Kontextmodellen

Dissertation zur Erlangung des Grades eines
Doktors der Naturwissenschaften

vorgelegt von

Dipl.-Inform. Christian Kuka

Gutachter:

Prof. Dr. Daniela Nicklas, Otto-Friedrich-Universität Bamberg
Prof. Dr. Klaus David, Universität Kassel

Oldenburg, 20. April 2015

Datum der Disputation: 02. März 2015

für Kristina
und meine Eltern Joachim und Monika

Zusammenfassung

Sensoren sind die Basis um in vielen Bereichen allgegenwärtige Anwendungen zu realisieren. So werden etwa bei fahrerlosen Transportsystemen Distanzsensoren, wie Lidar-, Radar- und Ultraschallsensoren eingesetzt um Distanzen zu messen und Hindernisse frühzeitig zu erkennen. Bei Offshore-Operationen dienen Lokalisierungssensoren dazu, den Aufenthaltsort von Personen oder Gütern bei Verladearbeiten zu bestimmen und so mögliche kritischen Situationen zu melden.

Um allerdings einen Nutzen aus mehreren und vor allem unterschiedlichen Sensoren zu ziehen und ein möglichst zeitnahes Bild der Umwelt zu erhalten, das alle anwendungsrelevanten Informationen enthält, bedarf es einer anwendungsgerechten Verknüpfung und Aggregation der Sensordaten durch eine effiziente strombasierte Verarbeitung. Ein solches Bild wird in dieser Arbeit als ein dynamisches Kontextmodell bezeichnet, da auf Basis der Informationen in einem solchen dynamischen Kontextmodell Anwendungen auf ihren aktuellen Kontext reagieren und Operationsentscheidungen treffen können. Hierfür wird in dieser Arbeit das Konzept von Datenstrommanagementsystemen (DSMS) für die Verarbeitung von Sensordaten verwendet. Ein DSMS bietet zum einen die Möglichkeit, aktive Datenquellen effizient und strombasiert zu verarbeiten, auf der anderen Seite aber auch eine Schnittstelle um unterschiedliche, der Anwendung angepasste, Verarbeitungen von Sensordaten durch eine Anfragesprache zu formulieren.

Im Zuge dieser Arbeit wird dabei das Kontextmodell, in Anlehnung an die JDL Fusionsarchitektur, in drei Ebenen unterteilt, die Signalebene, die Merkmalebene und die Objektebene. Jede dieser Ebenen bietet dabei Kontextinformationen mit der jeweiligen semantischen Höhe an und kann daher in unterschiedlichen Szenarien verwendet werden. Die Informationen in den einzelnen Ebenen werden durch die Verarbeitung in einem DSMS bestimmt, wodurch eine Abbildung der Kontextebenen auf die Verarbeitungsoperatoren in einem Datenstrommanagementsystem stattfindet.

Durch die Verwendung von Sensordaten in solchen Anwendungen entsteht allerdings das Problem, dass die von Sensoren wahrgenommen Eigenschaften der Umwelt nicht frei von Fehlern sind. Gründe hierfür sind zum einen äußere Einflüsse, das verwendete Messprinzip, sowie die allgemeine Unschärfe bei der Messung von Sensoren, die sich negativ auf die wahrgenommenen Eigenschaften von Sensoren auswirken und zu Qualitätsverlusten führen. Daher muss die Frage beantwortet werden, wie eine qualitätssensitive Verarbeitung von Sensordaten in einem DSMS in einer allgemeinen und weitestgehend automatisierten Weise durchgeführt werden kann.

Datenstrommanagementsysteme bieten bisher keine Möglichkeit, die Qualität der Sensorwahrnehmungen automatisch auf Basis von Quelleninformationen an die Verarbeitungsergebnisse anzuheften. Stattdessen gehen sie vielmehr davon aus, dass die Daten die sie verarbeiten von exakter Natur sind. Um dieser Tatsache entgegenzuwirken wird die Bereitstellung von Kontextinformation in diesem Ansatz mit zusätzlichen Qualitätsinfor-

mationen angereichert, so dass Anwendungen zusätzliches Wissen über die Informationen in der jeweiligen Kontextmodellebene erhalten. Hierzu werden im Rahmen dieser Arbeit Verfahren entwickelt, um auf Basis einer Ontologie vom Anwender formulierte Anfragen durch zusätzliche Quellen zu erweitern. Die verwendete Ontologie beschreibt dabei die Beziehungen zwischen Sensoren, ihren Wahrnehmungen und ihren aktuellen Qualitäten. Die zusätzlichen Quellen werden in diesem Ansatz dazu verwendet, die aktuellen Qualitäten der einzelnen Sensorwahrnehmungen kontinuierlich zu bestimmen und mit der ursprünglichen Verarbeitungsanfrage zu verknüpfen. Des Weiteren wird auf Basis der aktuellen Messwerte das stochastische Modell der Daten bestimmt, welches die Exaktheit der Sensorwahrnehmungen wiedergibt.

Um diese zusätzlichen Informationen in einem DSMS nutzen und verarbeiten zu können, werden in einem zusätzlichen Schritt die existierenden Operatoren und Datenstrukturen eines DSMS dahingehend erweitert, dass diese Qualitäten abgebildet, genutzt und zwischen Operatoren ausgetauscht werden können. Dies führt zu einer probabilistischen Verarbeitung von Datenelementen in einem DSMS. Bei der hier entwickelten probabilistischen Datenstromverarbeitung werden daher Sensorwahrnehmungen nicht mehr länger nur als deterministische Werte dargestellt, sondern in Form von (multivariaten) Wahrscheinlichkeitsverteilungen repräsentiert, welche die Qualität der einzelnen Datenelemente wiedergeben und Korrelationen zwischen Datenelementen einer Messung zulassen. Bei der Erweiterung wird darauf geachtet, dass die ursprüngliche Anfragesprache zur Verarbeitungsbeschreibung der Sensordaten unverändert bleibt.

Der beschriebene Ansatz wurde in dem Datenstrommanagementsystem Odysseus prototypische implementiert und in den beiden Forschungsprojekten SALSA und SOOP angewandt. Das Projekt SALSA beschäftigte sich dabei mit der sicheren Umgebungswahrnehmung für fahrerlose Transportfahrzeuge im Außenbereich, währen das SOOP Projekt maßgeblich der Frage nach der Sicherheit bei Offshore-Operationen nachging. In beiden Projekten wurden Teile des implementierten Ansatzes hinsichtlich seiner Eignung evaluiert.

Abstract

Sensors are the basis to realize pervasive applications in many areas. In the context of autonomous transport vehicles for example, distance sensors, such as Lidar, Radar and ultrasonic sensors are used to measure distances and to identify obstacles early on. In other scenarios such as offshore operations, ultra wideband sensors and global positioning systems are used to display the current position of people and cargos to determine and to warn the participants in case of critical situations.

However, in order to benefit from multiple and especially diverse sensors and to obtain a timely picture of the environment that contains all application-relevant information, it requires an application oriented linking and aggregation of sensor data by an efficient flow-based processing. Such a picture is referred as a dynamic context model in this work. Based on the information in such a dynamic context model applications are able to make context-aware decisions. For this purpose, the concept of data stream management systems (DSMS) is used for the processing of sensor data in this work. A DSMS offers the opportunity to process active data sources in an efficient and flow-based manner. On the other side such systems provide an interface to formulate an application adopted processing of sensor data through a query language.

In the course of this work the context model is divided into three levels; the signal level, the feature level and the object level based on the JDL sensor data fusion architecture. Each of these levels provides context information with the respective semantic level and can therefore be used in different scenarios. The information in each layer are determined by the processing in a DSMS. Thereby causing a mapping between the context levels and the processing operators in a data stream management system.

By the use of sensor data in such applications, however, there arises a problem that the perceived properties of the environment by the sensors are not free from errors. The reasons for this are, firstly, external influences, the used measuring principle, as well as the general blur in the measurement of sensors which have a negative impact on the perceived properties of sensors and lead to loss of quality. Therefore, the question needs to be answered, if a quality-aware processing of sensor data in a DSMS can be realized in a generic and automated manner.

Data stream management systems do not provide the possibility to automatically annotate the quality on the basis of source information to the processing results until now. Instead, they rather assume that the data they process are of exact nature. To counteract this fact, in this approach the processing results are enriched with additional quality information so that applications obtain additional knowledge about the information in the specific context model level. For this purpose, a method is developed on the basis of an ontology to expand a user requests formulated processing query by additional sources. The ontology used here describes the relationships between sensors, their perceptions and their current conditions. The additional sources are used to determine the actual qualities

of the individual sensor perceptions continuously and link it to the original processing request. Further, a stochastic model is estimated based on the current measurements to describe the accuracy of the sensor perceptions.

In order to use this additional information in a DSMS and perform a deterministic processing, the existing operators and data structures of a DSMS are extended such that these qualities are represented, used, and exchanged between operators. This leds to a probabilistic processing of data elements in a DSMS. In the hereby developed probabilistic data stream processing the sensor perceptions are no longer represented only as deterministic values, but represented in the form of (multivariate) probability distributions, which reflect the quality of the individual data elements and allow correlations between data elements of a measurement. In the extension it is ensured that the actual query language for the formulation of the processing of sensor data will remain unchanged.

The approach described in this work has been prototypical implemented in the data stream management system Odysseus and applied in the two research projects SALSA and SOOP. The SALSA project has investigated the secure environment recognition for driverless transport vehicles. The SOOP project focused on the observation of safety in offshore operations. In both projects, parts of the implemented approach were evaluated to determine its suitability.

Inhalt

1 Einleitung

In vielen Bereichen, in denen Anwendungen mit ihrer Umgebung interagieren und auf Ereignisse in ihrer Umwelt reagieren, werden Sensoren eingesetzt um diese Umwelt wahrzunehmen. So werden etwa Distanzsensoren, wie Laserscanner und Radar, im Automotivbereich eingesetzt um Hindernisse und andere Gefahren frühzeitig zu erkennen und situationsgerecht zu reagieren. Bei der Unterstützung von Verladeoperationen in der Offshore-Logistik werden Lokalisierungssensoren eingesetzt um den Aufenthaltsort von Personen oder Gütern zu bestimmen und so mögliche kritischen Situationen zu vermeiden.

1.1 Motivation

Sensoren können eine Vielzahl von Aspekten ihrer Umgebung wahrnehmen und lassen sich so in unterschiedlichster Form einsetzen. Um allerdings ein möglichst zeitnahes und vollständiges Bild der Umwelt zu erhalten, bedarf es immer auch einer anwendungsgerechten Verknüpfung und Aggregation der Sensordaten. Ein solches Bild spiegelt dabei den aktuellen Kontext einer Anwendung wieder, weshalb hier von einem Kontextmodell gesprochen werden kann. Da sich die Informationen in dem Kontextmodell durch jede neue Messung der verwendeten Sensorik kontinuierlich verändern können, spricht man hier von einem dynamischen Kontextmodell.

Die verwendete Sensorik unterliegt in solchen Anwendungen allerdings unterschiedlichen Umwelteinflüssen, welche sich negativ auf die Wahrnehmung der Sensorik auswirken können. Damit sich eine Anwendung auf die Informationen in einem dynamischen Kontextmodell stützen und Operationsentscheidungen treffen kann, bedarf es daher nicht nur der reinen Sensorwahrnehmung, sondern auch immer der Qualität dieser Wahrnehmungen. Die Qualität gibt an, in wie weit die Wahrnehmungen der Sensorik und die daraus folgenden Schlüsse über den aktuellen Kontext frei von möglichen Fehlern sind.

1.2 Szenario

Um die anwendungsgerechte Verknüpfung und Aggregation unter der Berücksichtigung der Wahrnehmungsqualität der verwendeten Sensorik zu untersuchen, ist diese Arbeit an zwei typische Anwendungsfälle angelehnt. Das erste Szenario ist dabei aus dem Transportbereich und behandelt die Erweiterung der Wahrnehmung von fahrerlosen Transportsystemen durch externe Sensorik. Das zweite Szenario stammt aus dem maritimen Bereich und fokussiert sich auf die sensorgestützte Überwachung von Verladeoperationen zur Vermeidung von Personenschäden.

1.2.1 Fahrerlose Transportsysteme

Die Verwendung von fahrerlosen Transportsystemen ist in zutrittsbeschränkten Gebieten weit verbreitet. Zu diesen zutrittsbeschränkten Gebieten zählen etwa die Intralogistik, in autonomen Geschäftsfluren und in beschränkten Bahnen, sowie der Passagiertransport durch automatische Züge an Flugterminals. Jedoch existieren gerade bei der Verwendung im teil-öffentlichen und öffentlichen Bereich Sicherheitsbeschränkungen die eine sehr niedrige Operationsgeschwindigkeit vorgeben. Dies ist vor allem der Tatsache geschuldet, dass die derzeitigen fahrerlosen Transportsysteme nur einen sehr begrenzten Bereich ihrer Umgebung durch ihre eigene, am Fahrzeug angebrachte, Sensorik überwachen und wahrnehmen können. Dies ist insbesondere deshalb hinderlich, da fahrerlose Transportsysteme in größeren öffentlichen Arealen dadurch nicht effizient betrieben werden können.

Da allerdings immer mehr und mehr Sensoren in der Umwelt verfügbar sind, liegt die Idee nahe, diese Sensoren zu verknüpfen und zu aggregieren. Auf diese Weise lassen sich statische und dynamische Objekte, die außerhalb des Erfassungsbereichs der Sensorik des fahrerlosen Transportsystems sind, aus unterschiedlichen Blickwinkeln detektieren. Das so entstandene zusätzliche Wissen über die Umgebung des Transportsystems kann so als ein globales Kontextmodell dem fahrerlosen Transportsystem zur Verfügung gestellt werden und als Basis für zukünftige Operationsentscheidungen dienen. Hierbei können sich die Wahrnehmungen der unterschiedlichen Sensoren allerdings unterscheiden, da der Detektionsbereich der Sensoren etwa durch Objekte oder Umwelteinflüsse beeinträchtigt wird oder die Messungen zu unterschiedlichen Zeitpunkten erfolgt. Aus diesem Grund ist es wichtig die Sensoren entsprechend korrekt zu fusionieren um ein Gesamtbild zu erhalten.

1.2.2 Offshore-Operationen

Bei Offshore-Operationen, wie etwa der Konstruktion von Offshore-Windrädern, kommt es immer wieder bei Verladearbeiten zu Unfällen, weil sich die beteiligte Mitarbeiter in Gefahrenbereichen, wie etwa unter einer schwebenden Last, aufhalten. Zur Verbesserung der Sicherheit von Offshore-Operationen wird daher in aktuellen Forschungsprojekten über den Einsatz von Sensoren nachgedacht, um die Position von Mitarbeitern und Gütern zu erfassen und kritische Situationen frühzeitig zu erkennen und entsprechend zu reagieren. Allerdings ist auf Grund der Umweltbedingungen auf hoher See eine akkurate Messung durch die verwendete Sensorik nicht immer möglich. Dies wirkt sich wiederum negativ auf die Verarbeitungsergebnisse zur Situationserkennung in Form von Widersprüchen und Fehlerkennung aus. Verfügt jedoch das Kontextmodell über Qualitätsinformationen zu den enthaltenen Kontextinformationen können Widersprüche und Fehlerkennung in der Situationserkennung entsprechend kenntlich gemacht und teilweise durch die Wahl der Informationen mit der höchsten Qualität aufgelöst oder durch eine dritte Instanz, wie etwa einem Mitarbeiter, überprüft werden.

1.3 Problemstellung

Die beiden zuvor beschriebenen Anwendungen werfen mehrere Probleme auf. Erstens sind die aktuellen Lösungen zur Sensordatenverarbeitung meist an eine bestimmte Klasse von Sensoren gebunden, da die Algorithmen ein bestimmtes Datenformat benötigen. Zweitens ist das Wissen über die Charakteristiken der Sensoren indirekt in der Applikation verankert und eine spätere Erweiterung oder eine Integration neuer Sensoren würde eine erneute Implementierung zumindest einiger Teile der Anwendung erfordern. Drittens muss die Behandlung von äußeren Einflüssen, wenn sie innerhalb der Anwendung betrachtet werden, ein fester Bestandteil der Applikation sein. Dies bedeutet, dass jede Modifikation oder Erweiterung der Anwendung diese Informationen verwalten und mit verarbeiten muss, was in der Regel sehr fehleranfällig ist.

Die Probleme bisheriger Ansätze lassen sich dabei in die folgenden Teilprobleme gliedern:

1. Probleme der Flexibilität von sensordatenverarbeitenden Systemen: Viele Systeme, die in den betrachteten Anwendungen verwendet werden, können nur mit einer Klasse von Sensoren oder sogar nur mit Sensoren eines einzigen Herstellers arbeiten. Eine Kombination aus Messungen unterschiedlicher Sensoren um ein einheitliches Lagebild zu erstellen, welches im Rahmen dieser Arbeit als Kontextmodell bezeichnet wird, ist daher problematisch und scheitert meist an unterschiedlichen Datenformaten und Protokollen.

2. Probleme der Integration von Sensordatenqualitäten: Viele Systeme betrachten lediglich den Messwert eines Sensors, jedoch nicht die Qualität der Messung, wenn diese nicht direkt von dem Sensor selbst geliefert wird. Eine Integration von Qualitätsinformationen bedeutet daher einen zusätzlichen Implementierungsaufwand pro Sensor.

3. Probleme der Verarbeitung von Qualitätsinformationen: Selbst eine nachträgliche Integration von Qualitätsinformation in ein System bedeutet noch nicht, dass diese Qualitätsinformationen bei der Verarbeitung berücksichtigt und durch das komplette System korrekt propagiert werden.

Zur kontinuierlichen Verknüpfung und Aggregation von Daten aus unterschiedlichen aktiven Quellen haben sich in den letzten Jahren Datenstrommanagementsysteme etabliert. Diese Systeme ermöglichen eine flexible und effiziente Verarbeitung gerade auch von hochfrequenten Daten von Sensoren und erlauben die Verarbeitung über eine Anfragesprache zu steuern.

1.3.1 Forschungsfrage

Ein Datenstrommanagementsystem als Basistechnologie zur Verarbeitung von Sensormessdaten könnte in den betrachteten Anwendungsszenarien das erste Teilproblem lösen. Allerdings erlauben diese Systeme noch keine qualitätssensitive Verarbeitung zur Erstellung eines dynamischen Kontextmodells und können somit auch nicht die letzten beiden Teilprobleme lösen. Die, in dieser Arbeit zu beantwortende Forschungsfrage lautet daher:

Wie kann eine qualitätssensitive Verarbeitung von Sensordaten in einem Datenstrommanagementsystem in einer allgemeinen und weitestgehend automatisierten Weise durchgeführt werden?

1.4 Beiträge

Die grundlegende Idee dieser Arbeit wurde in [Kuk12] erstmalig beschrieben. Sie besteht darin, Sensordaten durch die Verwendung eines Datenstrommanagementsystems kontinuierlich zu verarbeiten um ein dynamisches Kontextmodell für eine Anwendung zu erstellen und dabei mit Qualitätsinformationen anzureichern. Damit soll eine qualitätssensitive Datenstromverarbeitung ermöglicht werden. Um dies zu erreichen gilt es die drei folgenden Teilziele zu fokussieren.

Erstellung von dynamischen Kontextmodellen

Zunächst müssen die für eine Anwendung benötigten Informationen zu einem einheitlichen Kontextmodell fusioniert werden. Hierzu wird das Kontextmodell in Ebenen unterschiedlicher Semantik unterteilt, um so die Daten je nach Informationsgehalt in das Kontextmodell zu integrieren. Dieses Kontextmodell ist dabei dynamisch, da die Informationen kontinuierlich durch neue Messungen der Sensoren aktualisiert werden.

Bestimmung von Qualitäten

Zur qualitätssensitiven Verarbeitung muss die Qualität einer Messung bestimmt werden. Hierzu werden zunächst der Begriff der Qualität und verschiedene Qualitätsdimensionen erläutert. Anschließend wird gezeigt, wie Beziehungen zwischen Umwelteinflüssen und Sensoren modelliert und genutzt werden um automatisch Verarbeitungsschritte zu generieren, die die aktuelle Qualität bestimmen können. Des Weiteren werden Verfahren zur kontinuierlichen direkten Qualitätsbestimmung auf Basis von Sensormessungen vorgestellt und aufgezeigt, wie diese in ein Datenstrommanagementsystem integriert werden können.

Qualitätssensitive Sensordatenverarbeitung

Unter diesem Punkt versteht sich die kontinuierliche Verarbeitung von qualitätsangereicherten Sensordaten durch semantisch klar definierte Verarbeitungsoperatoren. Hierzu wird zunächst ein Datenmodell entwickelt, welches die benötigten Qualitätsinformationen repräsentieren kann. Anschließend wird auf Basis dieses Modells die, für die Verarbeitung notwendigen, Operatoren auf Grundlage der relationalen Algebra definiert.

1.5 Aufbau der Arbeit

Zunächst werden im Kapitel 2 die Grundsteine zum Verständnis der Sensordatenverarbeitung und Fusion gelegt, sowie in das Themengebiet der Datenstrommanagementsysteme und die Konzepte der kontinuierlichen Datenverarbeitung eingeführt. Im anschließenden Kapitel 3 wird aufgezeigt, wie mit den beschriebenen Technologien dynamische Kontextmodelle erzeugt werden und in Anwendungen genutzt werden können. Die Ergebnisse dieses Kapitels wurden in [KGS+12], sowie in [EFG+12] veröffentlicht. Für die Unterstützung von qualitätssensitiven Anwendungen werden in Kapitel 4 Ansätze vorgestellt, um kontinuierliche Datenströme mit Qualitäten anzureichern, sowie Beziehungen zwischen Sensoren und ihren Messwerten zu repräsentieren und in Anwendungen zu nutzen. Teilergebnisse dieser Arbeiten finden sich in [KN12] und [KN14b]. In Folge dieser Arbeiten wird in Kapitel 5 eine Möglichkeit der Verarbeitung von qualitätsannotierten Daten präsentiert. Ein Teil der Ergebnisse aus diesem Kapitel wurden in [KN14c] vorgestellt und in [KN14a] demonstriert. In Kapitel 6 wird aufgezeigt, wie die entwickelten Verfahren in einem Prototypen zur Verarbeitung und Bereitstellung eines dynamischen Kontextmodells realisiert wurden. Hierzu wurde ein existierendes Datenstrommanagementsystem durch die zusätzlichen Datenmodelle und Verarbeitungsverfahren erweitert. In Kapitel 7 werden Ergebnisse der Evaluation von Teilen des Prototyps hinsichtlich der einzelnen Konzepte und der Anforderungen der Anwendungsszenarien vorgestellt. Die Evaluation wurde dabei sowohl auf synthetischen Daten, wie auch auf realen Sensordaten aus den Anwendungsszenarien durchgeführt. Abschließend werden in Kapitel 8 noch einmal die Beiträge dieser Arbeit Revue passiert und aufgezeigt an welchen Stellen noch offene Fragen oder Verbesserungsvorschläge existieren, die in dieser Arbeit nicht behandelt wurden. Ein Ausblick zeigt zudem weitere, im Rahmen dieser Arbeit entstandene, Forschungsfragen auf.

Ältere Arbeiten des Autors, die sich zum Teil mit der kontinuierlichen Verarbeitung von Datenströmen und ihrer Repräsentation befassen, finden sich in [BKW+09, BKNB11, FBK+11] und [KBNB11].

2 Grundlagen

Dieses Kapitel befasst sich mit den Grundlagen der Umgebungswahrnehmung durch Sensoren und der anschließenden Aufbereitung und Aggregation von Sensordaten durch Sensorverarbeitungssysteme. Ziel ist zunächst wichtige Grundlagen für die Erstellung von Kontextmodellen zu schaffen, welche in den beiden genannten Anwendungsszenarien benötigt werden, damit die beteiligten Systeme auf ihre Umgebung reagieren können. Hierzu wird zunächst ein Überblick über existierende Sensoren gegeben, welche in Anwendungen zur Umgebungswahrnehmung und Lagebilderstellung zum Einsatz kommen. Im darauf folgenden Abschnitt wird auf den Begriff der Sensorfusion eingegangen und am Beispiel des Joint Directors of Laboratories (JDL) Datenfusionsprozessmodells die unterschiedlichen Stufen der Fusion erläutert.

Um eine solche Sensorfusion zu realisieren wird im Weiteren auf die Technologie der kontinuierlichen Datenstromverarbeitung eingegangen und die dort verwendeten Datenmodelle und die zugrundeliegende temporale relationale Algebra erläutert. Abschließend wird gezeigt, wie mit den Operatoren der temporalen relationalen Algebra die einzelnen Stufen des Datenfusionsprozessmodells realisiert werden können und so ein Datenstrommanagementsystem zur Sensordatenfusion eingesetzt werden kann.

2.1 Sensoren

In den zuvor aufgeführten Szenarien interagieren die Anwendungen direkt mir ihrer Umgebung ohne hierzu zuvor die Eingabe von Daten durch einen Anwender zu erhalten. Damit dieses möglich ist, müssen die Anwendungen direkt ihre Umgebung wahrnehmen können. Zu diesem Zweck werden Sensoren verwendet, die bestimmte Eigenschaften ihrer Umgebung erfassen. Unter einem Sensor versteht man dabei ein technisches Bauteil, das bestimmte physikalische oder chemische Eigenschaften seiner Umwelt qualitativ oder in Messgrößen quantitativ erfassen kann [HSK13]. Ein Sensor kann aber auch eine Softwarekomponente sein, die Messwerte über ein empirisches erlerntes oder physikalisches Modell berechnen kann.

Ein Sensor kann entweder autonom seine Umwelt kontinuierlich erfassen oder dies über eine explizite Anfrage tun. Dabei kann ein Sensor auch aus mehreren internen Sensoren bestehen, die unterschiedliche oder gleiche Eigenschaften der Umwelt erfassen. Die so entstandenen Messwerte werden dabei über ein Kommunikationsprotokoll an einen oder mehrere Interessenten übermittelt oder von diesen über das gleiche oder ein anderes Kommunikationsprotokoll abgefragt.

Im Folgenden werden die typischen Sensoren beschrieben, welche in Anwendungsszenarien zur Umgebungswahrnehmung und Lagebilderstellung zum Einsatz kommen. Hierbei wird im Rahmen dieser Arbeit zwischen aktiven und passiven Sensoren unterschieden.

Die Unterscheidung basiert dabei nicht auf der internen Verarbeitung des Sensors, sondern auf der Interaktion mit seiner Umwelt.

2.1.1 Aktive Sensoren

Aktive Sensoren verändern aktiv ihre Umgebung durch die Emittierung eines Signals und messen anschließend die Veränderung des Signals oder die Laufzeit bis zu einer Reflektion durch die Umgebung. Dies erlaubt aktiven Sensoren unteranderem Geschwindigkeiten und Bewegungsrichtungen von Objekten abzuschätzen. Durch die aktive Veränderung ihrer Umgebung können aktive Sensoren die Wahrnehmung von anderen Sensoren beeinflussen.

Radarsensoren Radarsensoren emittieren elektromagnetische Wellen um die Entfernung, Richtung und Geschwindigkeit von reflektierenden Objekten zu bestimmen. Dabei lassen sich Radarsensoren in Pulsradar, Strichradar und gepulste Strichradarsensoren unterscheiden. Beim Pulsradar werden kurze Impulse emittiert. Bei Dauerstrichradaren wird dauerhaft ein Signal gesendet und empfangen und dabei die Sendefrequenz im Gegensatz zum Pulsradar variiert. Das gepulste Dauerstrichradar stellt eine Kombination beider Verfahren dar. Zu den Stärken von Radarsensoren zählt die Möglichkeit durch Hindernisse hindurch zu sehen, weshalb sie auch keine direkte Sichtverbindung benötigen. Radarsensoren finden sich sowohl im Automotivbereich zur Detektion von Objekten, aber auch im Schifffahrtsbetrieb.

Ultraschallsensoren Über die Erzeugung und den Empfang von Ultraschallwellen erlauben Ultraschallsensoren die Messung der Entfernung zu Objekte. Zu den Stärken von Ultraschallsensoren zählt die Multi-Echo-Verarbeitung, welche es ermöglicht Objekte über kleinere Hindernisse hinweg zu detektieren. Zu den Schwächen zählen allerdings Signallaufschwankungen aufgrund von Temperatur und Luftfeuchtigkeit und die Notwendigkeit einer direkten Sichtverbindung.

Lidarsensoren Bei Lidarsensoren oder Laserscannern werden Laserimpulse über eine Vorrichtung ausgestrahlt und die Laufzeit bis zur Reflexion gemessen. Über einen internen rotierenden Spiegel kann auf diese Weise zudem ein zweidimensionales Abbild der Umgebung erstellt werden. In Abbildung 2.1 ist ein solcher Laserscanner der Firma SICK abgebildet. Der Laserscanner bietet die Möglichkeit seine Umgebung in einem Radius von $270\,^\circ$ und einer Auflösung von bis zu $0.25\,^\circ$ in einer Frequenz von 25Hz abzutasten und diese Messung als Punktwolke bestehend aus Winkel, Distanz und Remission zu übermitteln.

Laserscanner gehören zu den typischen Sensoren um Distanzen zu möglichen Hindernissen oder anderen Objekten zu messen. Häufige Anwendungen sind dabei autonome oder teilautonome Fahrzeuge. In [TCD+05] wurde zudem gezeigt, wie mit Laserscannern auch Objekte verfolgt werden können. Durch die Notwendigkeit einer direkten

Abbildung 2.1: *Laserscanner der Firma SICK*

Sichtverbindung sind Laserscanner allerdings anfällig gegenüber Schnee, Regen und Nebel.

2.1.2 Passive Sensoren

Passive Sensoren detektieren und messen verschiedene Eigenschaften ihrer Umgebung ohne diese dabei zu beeinflussen und ermöglichen es so Signaturen von Objekten zu erkennen.

Thermische Sensoren Thermische Sensoren, wie etwa Wärmebildkameras, messen die Wärmestrahlung von Objekten. Diese Art der Sensoren bietet eine sehr hohe Selektivität und Empfindlichkeit auf hohe Distanz, benötigt dafür aber eine direkte Sichtverbindung zu dem zu detektierenden Objekt.

Seismische Sensoren Seismische Sensoren bieten die Möglichkeit der Detektion von seismischen Objektsignaturen auf eine hohe Reichweite und benötigen dabei keine direkte Sichtverbindung. Zu ihren Schwächen zählen allerdings Signalschwankungen aufgrund von Bodenbeschaffenheiten, eine mäßig hohe Abtastrate, sowie eine erhöhte Speicher- und Zeitkomplexität der Signalverarbeitung.

Optische Sensoren Optische Sensoren dienen vor allem der Erkennung von Objekten. Zu den Stärken von optischen Sensoren zählt die hohe Reichweite. Hierbei kann unterschieden werden zwischen Mono- und Stereo-Kameras. Während Mono-Kameras meist dazu verwendet werden, Objekte aus Mustern zu erkennen, bieten Stereo-Kameras die Möglichkeit auch Entfernungen zu Objekten zu schätzen. Zu den Schwächen von optischen Sensoren zählen die Notwendigkeit einer direkten Sichtverbindung und die Speicher- und Zeitkomplexität der Signalverarbeitung. Bildgebende Verfahren sind zudem stark von dem Umgebungslicht abhängig und nur bedingt im Außenbereich einsetzbar.

Eigenschaften	Radarsensor	Ultraschallsensoren	Laserscanner	Thermische Sensoren	Seismische Sensoren	Mono-Kameras	Stereo-Kameras	Positionssensoren
Robustheit gegenüber Umwelteinflüssen	⊕⊕	⊖	⊕	⊖	⊖	⊖⊖	⊖⊖	⊕
Kosten	⊖	⊕⊕	⊕	⊖⊖	⊕⊕	⊕	⊖	⊕⊕
Reichweite	⊕⊕	⊕	⊕⊕	⊖	⊖	⊖⊖	⊕	
Ausmaß	⊕⊕	⊖⊖	⊕	⊕	⊖⊖	⊖	⊕⊕	⊖⊖
Klassifikation	⊕⊕	⊖⊖	⊖	⊖⊖	⊖⊖	⊕	⊕⊕	⊖⊖
Sichtverbindung	Nein	Ja	Ja	Ja	Nein	Ja	Ja	Nein
Signalverarbeitung	⊕	⊕⊕	⊕⊕	⊖⊖	⊖	⊕	⊖⊖	⊕⊕
Datenrate	40 Hz	15 Hz	25 Hz	25 Hz	14 Hz	25 Hz	25 Hz	5 Hz

Tabelle 2.1: *Charakteristiken von Sensortypen*

Positionssensoren Positionssensoren wie GPS oder D-GPS stützen sich auf die Signallaufzeit zu Referenzknoten um ein Objekt in einem zweidimensionalen oder dreidimensionalen Raum zu positionieren. Hierbei benötigen sie immer eine Signalverbindung zu den jeweiligen Referenzknoten, die entweder satellitengestützt sind oder innerhalb des Einsatzgebietes positioniert werden müssen.

2.1.3 Zusammenfassung

In Tabelle 2.1 sind noch einmal die einzelnen Typen von Sensoren gegenübergestellt. Während Laserscanner in vielen Szenarien zur Erstellung von Kontextmodellen hinreichende Eigenschaften bieten, sind Radarsensoren gerade in Bezug auf die Reichweite zu bevorzugen. Ultraschallsensoren dagegen bieten nur eine geringe Reichweite. Auf Grund ihrer Kosten im Vergleich zu Radarsensoren und Laserscanner sind Ultraschallsensoren für die zuverlässige Detektion von Objekten auf geringe Distanzen zu bevorzugen. Bildverarbei-

tende Sensoren dagegen sind relativ günstig und können verwendet werden um Objektausmaße zu bestimmen. Allerdings sind diese Sensoren dafür nicht sehr robust gegenüber Umwelteinflüssen, wie etwa direkter Sonneneinstrahlung. Die zusätzliche Signalverarbeitung ist zudem zeit- und speicherintensiv. Seismische Sensoren sind sehr günstig, erlauben aber nur die Detektion von Objekten und ihre Klassifikation anhand von bekannten Objektsignaturen. Thermische Sensoren erlauben die Detektion von weit entfernten Objekten anhand ihrer Wärmeabstrahlung, sind aber in ihrer Anschaffung vergleichsweise teuer. Auch hier ist die Signalverarbeitung, wie bereits bei den bildverarbeitenden Sensoren, sehr zeit- und speicherintensiv. Positionssensoren erlauben eine exakte Positionierung von Objekten, allerdings muss hierzu jedes Objekt mit einem solchen Sensor ausgestattet sein, was wiederum hohe Kosten verursacht. Des Weiteren verfügen Positionssensoren meist über eine, im Vergleich zu den anderen vorgestellten Sensortypen, niedrige Aktualisierungsfrequenz.

Neben den genannten Sensoren existieren natürlich noch eine Vielzahl anderer Sensorarten, welche für die Umgebungswahrnehmung eingesetzt werden können. Hierzu zählen beispielsweise Temperatur, Luftdruck und Beschleunigungssensoren über die Umwelteinflüsse und somit der aktuelle Arbeitsbereich von den beschriebenen Sensoren bestimmt werden kann.

2.2 Sensorfusion

Ziel der Sensorfusion ist es die Sensorwahrnehmungen der zuvor vorgestellten Sensoren zu kombinieren, um auf diese Weise höherwertige Information über den zu beobachteten Prozess zu erhalten. In [Wal99] wird die Datenfusion dabei wie folgt definiert:

> *„Data fusion is a formal framework in which are expressed means and tools for the alliance of data originating from different sources. It aims at obtaining information of greater quality; the exact definition of 'greater quality' will depend up on the application.“*

Die Datenfusion zielt also darauf ab, aus Daten Informationen von höherer Qualität zu erhalten. Die Definition von höherer Qualität ist dabei hochgradig anwendungsabhängig und kann erst durch die Anforderungen der Anwendung an die jeweiligen fusionierten Informationen definiert werden.

In den letzten Jahren wurde eine Vielzahl von Fusionsmodellen und Anwendungen entwickelt. Ein aktuelle Auflistung vorhandener Fusionstechniken wird in [KKKR11] gegeben. Eines der bekanntesten Fusionsmodelle ist dabei das JDL-Datenfusionsprozessmodell [HL97].

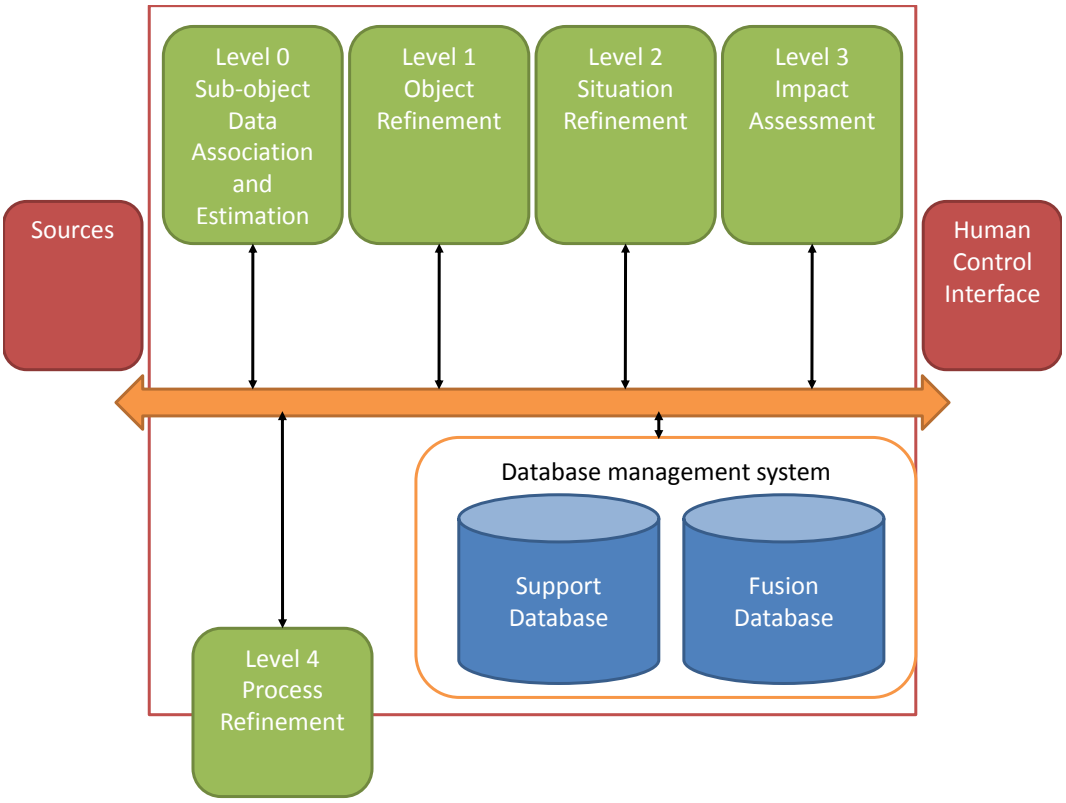


Abbildung 2.2: *Das JDL-Datenfusionsmodell nach Hall et al. [HL97]*

2.2.1 Joint Directories of Laboratory Datenfusionsprozessmodell

Das JDL-Datenfusionsprozessmodell (vgl. Abb. 2.2) ist ein konzeptionelles Modell, welches die Prozesse, Funktionen, Technologiekategorien und anwendungsspezifischen Techniken zur Datenfusion identifiziert. Ursprünglich bestand das JDL-Datenfusionsprozessmodell aus den Stufen 0-3 [W+88] und wurde nachträglich durch die Stufen 4 und 5 erweitert. Eine letzte Erweiterung des Modells wurde in [LBR+04] vorgenommen. In [HL97] beschreiben die Autoren die Aufgaben der einzelnen Stufen wie folgt:

Stufe 0 Quellenvorverarbeitung (Process Assignment) Ein erster Prozess ordnet die Daten den jeweils geeigneten Verfahren zu und führt eine Vorauswahl der Daten durch. Die Quellevorverarbeitung reduziert dabei die Last der Datenfusion durch die Zuweisung der Daten an die entsprechenden Prozesse. Die Quellenvorverarbeitung zwingt auch den Datenfusionsprozess sich auf die relevantesten Daten in der aktuellen Situation zu konzentrieren.

Stufe 1 Objektverfeinerung (Object Refinement) In dieser Stufe werden Standort-, parametrische, sowie Identitätsinformationen kombiniert um eine verfeinerte Darstellung

von einzelnen Objekten zu erhalten. Die Verarbeitung in Stufe 1 führt hierbei vier Hauptfunktionen aus:

1. Transformation von Sensordaten in einen einheitlichen Satz von Einheiten und Koordinaten,
2. Verfeinerung von zeitlicher Abschätzung von Objektpositionen, Kinematik und Attributen,
3. Zuordnung von Daten zu Objekten um statistische Schätzverfahren anwenden zu können,
4. Verfeinerung der Schätzung der Identität eines Objekts.

Stufe 2 Situationsverfeinerung (Situation Refinement) Die Verarbeitung in Stufe 2 entwickelt eine Beschreibung der aktuellen Beziehungen zwischen Objekten und Ereignissen im Kontext ihrer Umgebung. Verteilung von einzelnen Objekten (welche in Stufe 1 definiert wurden) werden untersucht, um sie zu sinnvollen Einheiten/Systemen zu aggregieren. Darüber hinaus konzentriert sich die Situationsverfeinerung auf relationale Informationen (d.h., körperliche Näherung, Kommunikation, sowie kausale, zeitliche und andere Beziehungen), um die Bedeutung einer Sammlung von Entitäten zu bestimmen. Situationsverfeinerung befasst sich mit der Interpretation von Daten, analog dazu, wie ein Mensch die Bedeutung der Sensordaten interpretieren könnte. Beide, formale und heuristische, Techniken werden verwendet, um, in einem bedingten Sinn, die Verarbeitungsergebnisse von Stufe 1 zu untersuchen.

Stufe 3 Gefährdungsverfeinerung (Impact Assessment) Die Verarbeitung in Stufe 3 projiziert die aktuelle Situation in die Zukunft, um Rückschlüsse über kritische Situationen zu ziehen. Die Verarbeitung entwickelt dabei alternative Hypothesen über die Strategien von Objekten und die Wirkung von unsicherem Wissen über die Umwelt.

Stufe 4 Prozessverfeinerung (Process Refinement) Die Verarbeitung in Stufe 4 kann nach den Autoren als *Metaprozess* verstanden werden, der andere Fusionsprozesse überwacht. Die Verarbeitung in Stufe 4 führt dabei die folgenden vier Hauptfunktionen aus:

1. Überwachung der Datenfusionsprozessleistung um Informationen über die Echtzeitsteuerung und langfristige Performance zu liefern,
2. Identifizierung, welche Informationen benötigt werden, um das Ergebnis der Fusion (Schlüsse, Positionen, Identitäten etc.) zu verbessern,
3. Bestimmung der quellenspezifischen Anforderungen, um relevante Informationen zu sammeln (d.h., welcher Sensortyp, welcher spezifische Sensor, welche Datenbank), und
4. Steuerung der Quellen, um die jeweilige Aufgabe der Fusion zu erfüllen.

Die letztere Funktion kann außerhalb des Bereichs der Datenfusion sein. Daher wird die Verarbeitung in Stufe 4 teilweise innerhalb und teilweise außerhalb des Datenfusionsprozesses gezeigt.

Stufe 5 Wahrnehmungsverfeinerung (Cognitive Refinement) Die Stufe 5 kam in der Revision des JDL Datenfusionsprozessmodells hinzu. Die Mensch-Maschine-Interaktion nimmt dabei Eingaben aus der höchsten Ebene des Fusionsprozesses entgegen.

Zusätzliche zu den beschriebenen Stufen des Datenfusionsprozesses existieren noch die Mensch-Maschine Schnittstelle und die Datenquellen zur Ein- und Ausgabe und Konfiguration, sowie die persistente Datenhaltung für unterstützende Information.

2.3 Datenstrommanagementsysteme

Datenstrommanagementsysteme bieten die Möglichkeit, Daten aus einer aktiven Quelle zu verarbeiten und dabei kontinuierlich Ergebnisse auf eine initial formulierte Anfrage zu liefern. Dieses Konzept der Verarbeitung hat sich in den letzten Jahren, wie in Abbildung 2.3 dargestellt, von reinen Forschungsprototypen wie etwa Aurora [ACc+03], PIPES [KS04], STREAM [ABB+03], oder Odysseus [AGG+12]) mehr und mehr zu kommerziellen Produkten wie etwa Esper[1], IBM System S [GAW+08] oder Amazon Kinesis[2] entwickelt.

Erste Verwendung von Datenstrommanagementsystemen zur Verfolgung von Objekten finden sich in dem STREAM Projekt, in welchem Datenstromelemente die Trajektorie eines beweglichen Objekts darstellen. In [PS04] werden neue Arten von Fenstertypen vorgestellt um den Sensordatenstrom in partielle Ströme für die Verarbeitung von Objekten zu zerteilen. Andere Arbeiten fokussieren die ortsbasierten Anfragen [HJ04] und orts-zeitbasierten Anfragen [GHM+07] innerhalb eines Datenstrommanagementsystems. Letztlich wurde auch das Nexus/Projekt um die Möglichkeit der Verarbeitung von Kontextdatenströmen erweitert [CEB+09].

2.3.1 Datenmodelle

In einer Vielzahl der genannten Datenstrommanagementsysteme wird das relationale Modell [Dat82] verwendet. Dies liegt zum einen daran, dass die Operatoren der relationalen Algebra eine gut definierte Semantik besitzen und die Verarbeitungsprinzipien aus dem Bereich der relationalen Datenbanken gut erforscht sind. Die Anwendung der Operatoren der relationalen Algebra auf einen Datenstrom in Form eines temporalen relationalen Operators wird durch die Schnappschussreduzierbarkeit (vgl. [Krä07]) ermöglicht. Die Schnappschussreduzierbarkeit sagt aus, dass wenn eine Operation durch einen temporalen relationalen Operator auf einen Datenstrom angewendet wird und anschließend ein

[1] http://www.espertech.com
[2] http://aws.amazon.com/kinesis

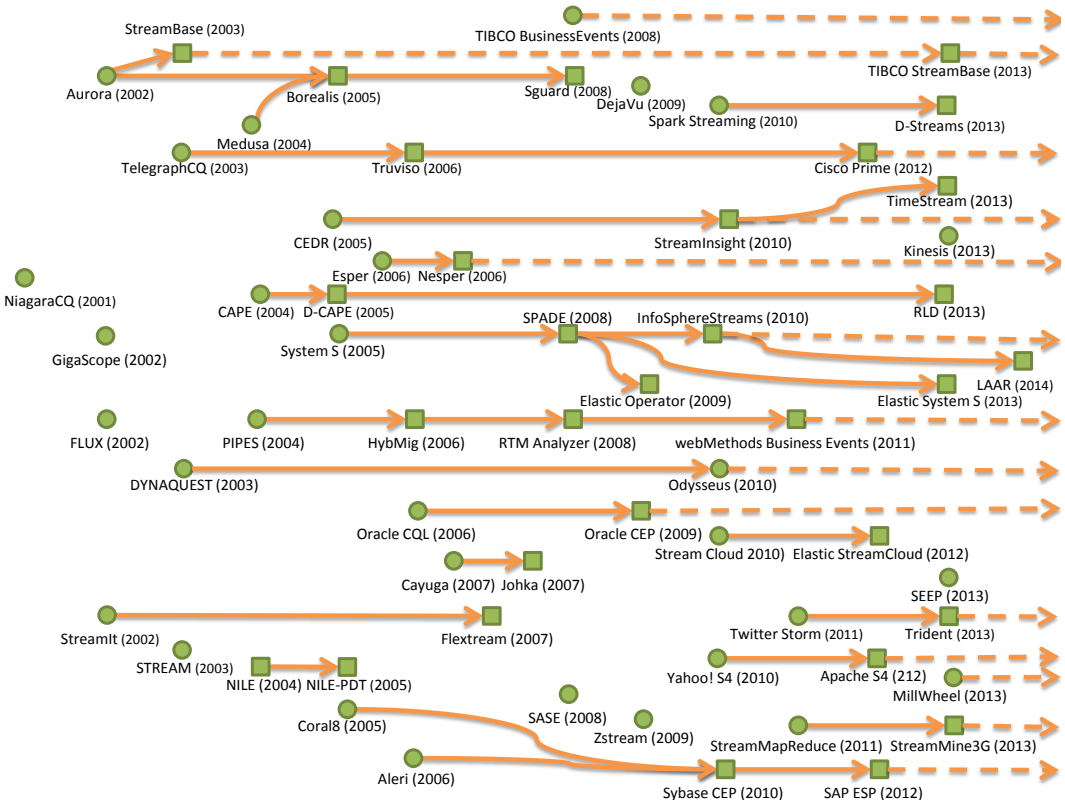


Abbildung 2.3: *Entwicklungsgeschichte der Datenstrommanagementsysteme nach Heinze et al. [HAQJ14]*

Schnappschuss für einen Zeitpunkt t berechnet wird, der Inhalt dieses Schnappschusses äquivalent ist zu einer vorherigen Schnapsschussberechnung und anschließender Anwendung eines nicht-temporalen relational Operators auf den Inhalt dieses Schnappschusses zum Zeitpunkt t. Ist diese Eigenschaft zu jedem Zeitpunkt gegeben, können die temporalen relationalen Operatoren als schnappschussreduzierbar bezeichnet werden.

Zusätzlich zu dem relationalen Model existieren noch weitere Datenmodelle wie etwa XML [BKF+07] und JSON welche die Verarbeitung von verschachtelten Daten zulassen. Des Weiteren wurden Operatoren und Datenmodell zur Verarbeitung von SPARQL-Anfragen auf RDF-Datenströmen [GGKL07, BBC+09] und Datenmodelle für die Verarbeitung von dynamischen Graphen [FKM+05] entwickelt.

Im Rahmen dieser Arbeit wird das von [Krä07] definierte Datenstrommodell verwendet. Die Operatoralgebra wird dabei auf Basis von logischen Datenströmen definiert. Sei hierzu $\mathbb{T} = (T; \leq)$ eine diskrete Zeitdomäne mit einer totalen Ordnung $\leq$ und T die Menge aller Zeitstempel. Ein logischer Datenstrom (Abb. 2.4) kann dann wie folgt definiert werden.

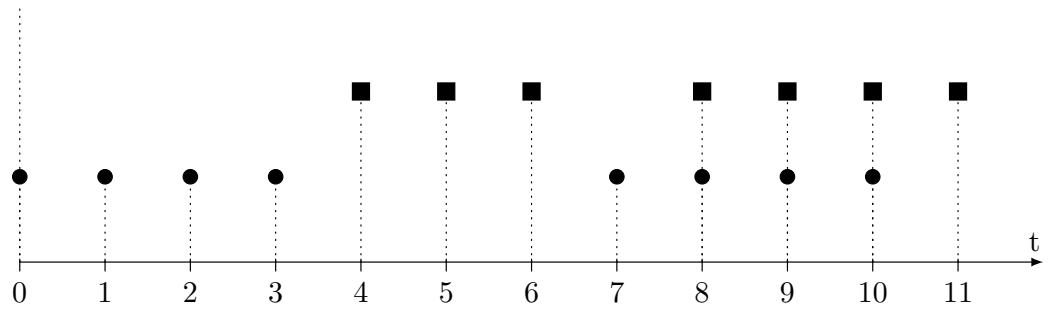


Abbildung 2.4: *Darstellung eines logischen Datenstroms*

Definition 1 (Logischer Datenstrom). Ein logischer Datenstrom S^l mit Schema $\mathcal{T}$ ist eine potenziell unendliche Multimenge von Elementen (e, t, n) für die gilt:

$$S^l := \{(e, t, n) | e \in \Omega_{\mathcal{T}} \wedge t \in T \wedge n \in \mathbb{N}_{>0}\} \tag{2.1}$$

Hierbei ist $e \in \Omega_{\mathcal{T}}$ ein Tupel aus der Menge aller Nutzdaten Ω mit Schema $\mathcal{T}$, $t \in T$ der assoziierte Zeitstempel und $n \in \mathbb{N}_{>0}$ gibt die Multiplizität des Tupels an. Ein Stromelement hat dabei die folgende Bedeutung: Ein Tupel e ist zum Zeitpunkt t gültig und existiert n mal zu diesem Zeitpunkt.

Ein logischer Datenstrom S^l genügt dabei den folgenden Bedingungen:

$$\forall (e, t, n), (\hat{e}, \hat{t}, \hat{n}) \in S^l : (e = \hat{e} \wedge t = \hat{t}) \implies (n = \hat{n}) \tag{2.2}$$

Die Bedingung verhindert, dass ein logischer Datenstrom mehrere Elemente mit gleichem Tupel und gleichem Zeitstempel enthält. Sei weiterhin $\mathbb{S}^l$ die Menge aller logischen Datenströme und $\mathbb{S}^l_{\mathcal{T}} \subset \mathbb{S}^l$ die Menge aller logischen Datenströme mit Schema $\mathcal{T}$.

Da die Darstellung aller gültigen Elementinstanzen zu jedem Zeitpunkt, wie sie in Abbildung 2.4 dargestellt ist, bei einer konkreten Implementierung ineffizient wäre, existieren unterschiedliche Ansätze um die Gültigkeit eines Elements über eine Zeitspanne zu repräsentieren. Zu den bekanntesten Ansätzen zählen der Positiv-Negativ-Ansatz [GHM+07] und der Intervall-Ansatz [Krä07]:

- Der Positiv-Negativ-Ansatz nutzt Negativ-Tupel (oder auch Lösch-Tupel) in der Verarbeitung, um den Ablauf der Gültigkeit eines Datentupels bekannt zu geben. Folgeoperatoren können entsprechend ihrer Logik Tupel aus ihrem internen Speicher entfernen und eine neue Ausgabe erzeugen, wenn sie ein identisches negatives Tupel erhalten.
- Der Intervall-Ansatz nutzt Gültigkeitsintervalle für jedes Datentupel. Ein Tupel trägt hierbei jeweils den Startzeitpunkt t_S und den Endzeitpunkt t_E seiner Gültigkeit. Diese Zeitpunkte bilden das halboffene Gültigkeitsintervall $[t_S, t_E)$. Auf diese Weise kann

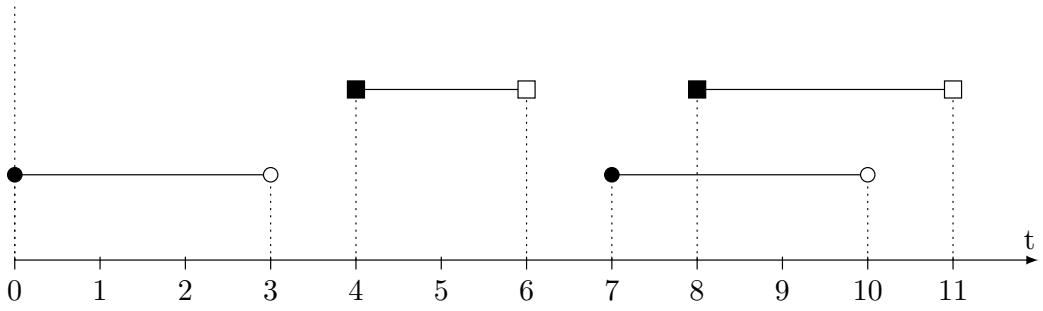


Abbildung 2.5: *Darstellung eines physischen Datenstroms*

ein Datensatz innerhalb eines Operators für ein vordefiniertes Zeitintervall als gültig betrachtet werden. Nach Ablauf dieses Intervalls kann das Datenelement aus dem Speicher des Operators entfernt werden.

Im weiteren Verlauf diese Arbeit wird der Intervall-Ansatz verwendet, um so einen konkreten physischen Datenstrom, wie in Abbildung 2.5, darzustellen. Ein physischer Datenstrom, basierend auf [Krä07], ist wie folgt definiert.

Definition 2 (Physischer Datenstrom). Sei $\mathbb{I} := \{[t_S, t_E) \in T \times T | t_S < t_E\}$ die Menge aller Zeitintervalle. Ein physischer Datenstrom S^p ist ein Paar $S^p = (M, \leq_{t_S,t_E})$. M ist dabei eine potenziell unendliche Sequenz von Tupeln $(e, [t_S, t_E))$ mit $e \in \Omega_{\mathcal{T}}$ und $[t_S, t_E) \in \mathbb{I}$.

$$S^p := \{(e, [t_S, t_E)) | e \in \Omega_{\mathcal{T}} \wedge [t_S, t_E) \in \mathbb{I}\} \tag{2.3}$$

Weiterhin besitzen alle Elemente aus M das gleiche Schema $\mathcal{T}$ und sind lexikalisch geordnet durch die Ordnungsrelation $\leq_{t_S,t_E}$ über M, so dass alle Tupel $(e, [t_S, t_E))$ nach ihren Zeitstempeln geordnet sind. Ein physisches Stromelement $(e, [t_S, t_E))$ mit Nutzdaten e ist innerhalb des Zeitintervalls $[t_S, t_E)$ gültig. $\mathbb{S}^p$ definiert des Weiteren die Menge aller physischen Datenströme.

2.3.2 Umwandlung zwischen physischem und logischem Strom

Durch den von [Krä07] definierten Operator φ kann ein physischer Datenstrom in einen logischen Datenstrom umgewandelt werden. Dabei sei $\varphi^{p \to l} := S^p \to S^l$ eine Abbildungsfunktion, die einen physischen auf einen logischen Datenstrom abbildet:

$$\begin{aligned} \varphi^{p \to l}(S^p) := & \{(e, t, n) \in \Omega_{\mathcal{T}} \times T \times \mathbb{N}_{>0} | \\ & n = |\{(e, [t_S, t_E)) \in S^p | t \in [t_S, t_E)\}| \wedge n \in \mathbb{N}_{>0}\} \end{aligned} \tag{2.4}$$

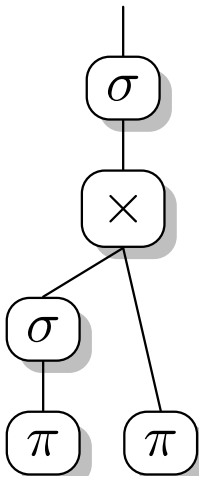


Abbildung 2.6: *Gerichteter Operatorgraph aus logischen temporalen relationalen Operatoren*

Auf diese Weise werden die, in einem Zeitintervall gültigen, Elemente wieder auf diskrete Zeitpunkte abgebildet und über die Variable n ihre Multiplizität wiedergegeben. Durch diesen Operatoren ist es möglich, die Semantik der Operatoren auf logischer Ebene zu definieren, während die konkrete Implementierung auf physischer Ebene durchgeführt werden kann.

2.3.3 Logische Operatoralgebra

Basierend auf der Definition des Umwandlungsoperators können die temporalen relationalen Operatoren eines Datenstrommanagementsystems auf logischer Ebene definiert werden. Diese Operatoren können dabei zu einem gerichteten logischen Operatorgraphen (vgl. Abb. 2.6) verknüpft werden, wobei die Knoten die logischen Operatoren repräsentieren und die gerichteten Kanten den Datenfluss darstellen. Für diese Arbeit fokussiert sich der folgende Abschnitt auf die, innerhalb der Sensorfusion, notwendigen Operatoren. Die temporal relationale Algebra umfasst die Operator zur Selektion, Projektion und Abbildung, sowie den Verbund und die Aggregation von Elementen aus einem logischen Datenstrom. Des Weiteren umfasst sie die Mengenoperationen, welche aber im Folgenden nicht näher betrachtet werden. Eine erweiterte Definition aller temporalen relationalen Operatoren findet sich in [Krä07].

2.3.3.1 Selektionsoperator

Der Selektionsoperator dient zur Filterung von Elemente, die nicht einem gegebenen Selektionskriterium genügen. Das Selektionskriterium selbst kann dabei auf die Nutzdaten

in dem jeweiligen Tupel zugreifen und auswerten. Für die Verarbeitung innerhalb einer Sensorfusion kann der Selektionsoperator daher für die Filterung von Ausreißern genutzt werden oder im Allgemeinen zur Reduktion der Systemlast herangezogen werden. Der temporale relationale Selektionsoperator kann daher wie im Folgenden definiert werden.

Definition 3 (Selektion (σ)). Eine Selektion $\sigma : \mathbb{S}^l_{\mathcal{T}} \times \mathbb{P}_{\mathcal{T}} \rightarrow \mathbb{S}^l_{\mathcal{T}}$ bildet alle Elemente eines logischen Datenstroms auf einen neuen logischen Datenstrom ab, die ein Selektionskriterium p erfüllen. Das Selektionskriterium $p \in \mathbb{P}_{\mathcal{T}}$ ist dabei eine Funktion $p : \Omega_{\mathcal{T}} \rightarrow \{true, false\}$ aus der Menge aller Selektionskriterien $\mathbb{P}_{\mathcal{T}}$ für ein Tupel mit Schema $\mathcal{T}$.

$$\sigma_p(S) := \{(e,t,n) \in S | p(e)\} \tag{2.5}$$

2.3.3.2 Abbildungsoperator

Der Abbildungsoperator hat die Aufgabe, Nutzdaten in einem Tupel durch die Anwendung einer Abbildungsfunktion zu verändern. Im Rahmen einer Sensorfusion kann dieser Operator eingehende Sensordaten beispielsweise auf eine systemweite einheitliche Einheit abbilden oder in ein einheitliches Koordinatensystem transformieren. Der Abbildungsoperator für einen logischen Datenstrom kann wie folgt definiert werden.

Definition 4 (Abbildung (μ)). Der Abbildungsoperator $\mu_f : \mathbb{S}^l_{\mathcal{T}} \times \mathbb{F}_{map} \rightarrow \mathbb{S}^l_{\hat{\mathcal{T}}}$ bildet den logischen Datenstrom $S^l_{\mathcal{T}}$ mit Schema $\mathcal{T}$ durch Anwendung einer Abbildungsfunktion f auf den logischen Datenstrom $S^l_{\hat{\mathcal{T}}}$ mit Schema $\hat{\mathcal{T}}$ ab. Sei hierzu $\mathbb{F}_{map}$ die Menge aller Abbildungsfunktionen, die ein Tupel mit Schema $\mathcal{T}$ auf ein Tupel mit Schema $\hat{\mathcal{T}}$ abbilden und $f \in \mathbb{F}_{map}$.

$$\begin{aligned} \mu_f(S) := \{(\hat{e},t,\hat{n}) | \exists X \subseteq S : X \neq \emptyset \\ \wedge X = \{(e,t,n) \in S | f(e) = \hat{e}\} \\ \wedge \hat{n} = \sum_{(e,t,n) \in X} n\} \end{aligned} \tag{2.6}$$

2.3.3.3 Projektionsoperator

Der Projektionsoperator hat die Aufgabe, Attribute innerhalb eines Tupels auf eine Teilmenge von Attribute zu reduzieren. Bei der Sensorfusion lässt sich so eine Fokussierung auf eine bestimmte Menge von Attributen realisieren. Die Definition des logischen Projektionsoperators lautet daher wie folgt.

Definition 5 (Projektion (π)). Der Projektionsoperator $\pi_{\hat{\mathcal{T}}} : \mathbb{S}^l_{\mathcal{T}} \rightarrow \mathbb{S}^l_{\hat{\mathcal{T}}}$ bildet den logischen Datenstrom $S^l_{\mathcal{T}}$ mit Schema $\mathcal{T}$ auf den logischen Datenstrom $S^l_{\hat{\mathcal{T}}}$ mit Schema $\hat{\mathcal{T}}$ ab.

$$\begin{aligned}\pi_{\hat{\mathcal{T}}}(S) := \{(\hat{e}, t, \hat{n}) | \exists X \subseteq S : X \neq \emptyset \\ \wedge X = \{(e, t, n) \in S | e_{\hat{\mathcal{T}}} = \hat{e}\} \\ \wedge \hat{n} = \sum_{(e,t,n) \in X} n\}\end{aligned} \tag{2.7}$$

2.3.3.4 Verbundoperator

Der Verbundoperator erlaubt es, Elemente aus zwei logischen Datenströmen, die zum selben Zeitpunkt gültig sind, zu einem neuen logischen Datenstrom zu verbinden. Dieser Operator ist dabei für den temporal korrekten Verbund von Datenströmen aus unterschiedlichen Quellen innerhalb der Sensorfusion zuständig. Der logische Verbundoperator lässt sich wie folgt definieren.

Definition 6 (Verbund ($\times$)). Der Verbund $\times : \mathbb{S}^l_{\mathcal{T}_1} \times \mathbb{S}^l_{\mathcal{T}_2} \rightarrow \mathbb{S}^l_{\hat{\mathcal{T}}}$ von zwei Strömen kombiniert alle Elemente aus beiden Strömen mit Schema $\mathcal{T}_1$ und Schema $\mathcal{T}_2$, die zum selben Zeitpunkt gültig sind, zu einem neuen logischen Datenstrom mit Schema $\hat{\mathcal{T}}$. Für die Kombination wird die Funktion $\circ : \Omega_{\mathcal{T}_1} \times \Omega_{\mathcal{T}_2} \rightarrow \Omega_{\hat{\mathcal{T}}}$ verwendet, welche die beiden Tupel konkatiniert.

$$\times(S_1, S_2) := \{(\circ(e_1, e_2), t, n_1 n_2) | (e_1, t, n_1) \in S_1 \wedge (e_2, t, n_2) \in S_2\} \tag{2.8}$$

2.3.3.5 Aggregationsoperator

Der Aggregationsoperator dient der Verdichtung von Nutzdaten durch die Anwendung einer Aggregationsfunktion auf eine Menge von Elementen. Innerhalb der Sensorfusion ist eine solche Verarbeitung notwendig, um Informationen über eine Menge von Sensormessungen zu bestimmen. Konkret lässt sich dieses Verhalten in Form eines logischen Aggregationsoperators wie folgt definieren.

Definition 7 (Aggregation (α)). Der Aggregationsoperator $\alpha : \mathbb{S}^l_{\mathcal{T}} \times \mathbb{F}_{agg} \rightarrow \mathbb{S}^l_{\hat{\mathcal{T}}}$ berechnet eine gegebene Aggregationsfunktion f_{agg} über die nicht-temporale Multimenge von Elementen aus dem logischen Datenstrom $\mathbb{S}^l_{\mathcal{T}}$ mit Schema $\mathcal{T}$, die zu einem Zeitpunkt gültig sind. Sei hierzu $\mathbb{F}_{agg}$ die Menge aller Aggregationsfunktionen. Eine Aggregationsfunktion $f_{agg} \in \mathbb{F}_{agg}$ mit $f_{agg} : \mathcal{P}(\Omega_{\mathcal{T}} \times \mathbb{N}_{>0}) \rightarrow \Omega_{\hat{\mathcal{T}}}$, wobei $\mathcal{P}$ die Potenzmenge symbolisiert, berechnet das Aggregat mit Schema $\hat{\mathcal{T}}$ über eine Menge von Elementen (e, n) .

$$\begin{aligned}\alpha_{f_{agg}}(S) := \{(agg, t, 1) | \exists X \subseteq S : X \neq \emptyset \\ \wedge X = \{(e, n) | (e, t, n) \in S\} \\ \wedge agg = f_{agg}(X)\}\end{aligned} \tag{2.9}$$

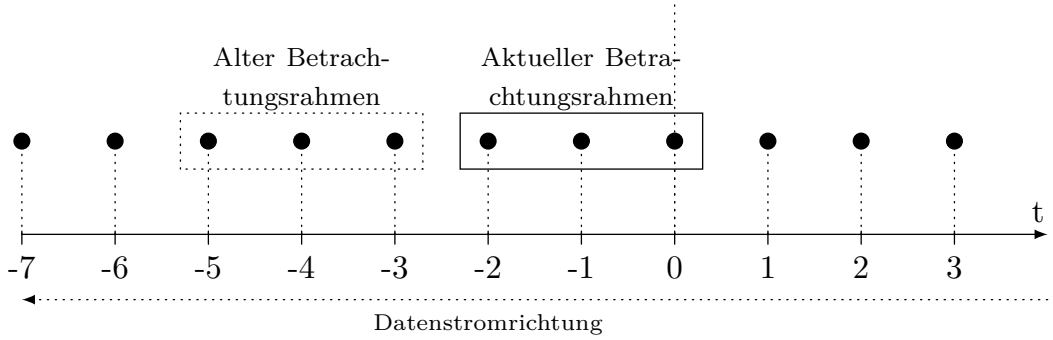


Abbildung 2.7: *Fensteroperator auf einem kontinuierlichen Datenstrom mit einem Betrachtungsrahmen der Größe 3*

2.3.4 Fensteroperatoren

Da ein Datenstrom eine möglicherweise unendliche Sequenz von Elementen darstellt, können blockierende oder statusbehaftet Operatoren nicht direkt auf einem Datenstrom ausgeführt werden, da diese erst ein Ergebnis liefern können, wenn sie ihre komplette Eingabe gesehen haben [BBD+02, MLT+05]. Zu diesen blockierenden Operatoren zählen die Aggregation, die Gruppierungen, sowie die Mengenoperatoren. Zu den statusbehafteten Operatoren zählt der Verbundoperator.

Eine Lösung hierzu ist die Partitionierung des Datenstroms in Teilmengen, sogenannte Fenster. Jedes Fenster besteht aus einer endlichen Menge von Tupeln, so dass blockierende oder statusbehaftete Operatoren auf dieser Menge ausgeführt werden können. Die verbreitetsten Arten von Fenstern sind Zeit- und Elementfenster.

Bei beiden Fensterarten kann zwischen gleitenden, springenden und taumelnden Fenstern unterschieden werden.

- Gleitende Fenster verschieben den Betrachtungsrahmen immer um genau eine Einheit.
- Springe Fenster verschieben den Betrachtungsrahmen immer um die Fenstergröße, so dass ein Tupel immer nur in einem Betrachtungsrahmen gültig ist.
- Taumelnde Fenster stellen einen Kompromiss aus den beiden zuvor genannten Fenstertypen dar, wodurch nur ein Teil der Tupel in mehreren Betrachtungsrahmen gültig sind.

Ein Zeitfenster definiert den Betrachtungsrahmen über eine Zeitspanne und kann wie folgt definiert werden.

Definition 8 (Zeitfenster (ω)). Das Zeitfenster $\omega_w : \mathbb{S}^l_{\mathcal{T}} \times \mathbb{N}_{>0} \rightarrow \mathbb{S}^l_{\mathcal{T}}$ bildet zu jedem Zeitpunkt den logischen Datenstrom $S^l_{\mathcal{T}}$ mit Schema $\mathcal{T}$ auf den logischen Datenstrom $S^l_{\mathcal{T}}$ mit Zeitspanne $w \in \mathbb{N}_{>0}$ ab.

$$\begin{aligned}\omega_w(S) := \{(e, \hat{t}, \hat{n}) | \exists X \subseteq S : X \neq \emptyset \\ \wedge X = \{(e, t, n) \in S | \max(\hat{t} - w + 1, 0) \leq t \leq \hat{t}\} \\ \wedge \hat{n} = \sum_{(e,t,n) \in X} n\}\end{aligned} \tag{2.10}$$

Ähnlich dem Zeitfenster bildet auch das Elementfenster einen logischen Datenstrom auf einen neuen logischen Datenstrom ab, wobei bei einem Elementfenster jeweils immer die letzten w Datensätze als gültig betrachtet werden. Wird ein neuer Datensatz konsumiert und innerhalb des Elementfensters befinden sich bereits w Datensätze, so wird der älteste Datensatz innerhalb des Elementfensters entfernt.

2.3.5 Verarbeitung

Bei der Verarbeitung in einem Datenstrommanagementsystem kann zwischen einer zeitgetriebenen Verarbeitung und einer elementgetriebenen Verarbeitung unterschieden werden. Bei der zeitgetriebenen Verarbeitung wird von dem System in Zyklen die Verarbeitung von Daten durch die Operatoren angestoßen. Hierbei kann das System selbst die Geschwindigkeit der Verarbeitung steuern und Daten von Quellen abrufen.

Bei der elementgetriebenen Verarbeitung wird die Verarbeitung mittels der Operatoren durch die Elemente im Strom angestoßen. Hierzu ist es notwendig, dass die Quellen aktiv Daten in das System übermitteln. Hierdurch wird die Geschwindigkeit der Verarbeitung durch die Quellen selbst gesteuert.

2.3.6 Anfragesprachen

Zur Definition der Verarbeitung von Daten muss der Anfragegraph (siehe Abbildung 2.6) innerhalb des Systems in Form einer Anfrage beschrieben werden. Die Formulierung einer Anfrage in einem Datenstrommanagementsystem geschieht dabei je nach Ausprägung des Systems über eine deklarative oder prozedurale Anfragesprache.

2.3.6.1 Deklarative Anfragesprachen

Im Bereich der Datenstrommanagementsysteme werden deklarative Anfragesprachen verwendet um eine Anlehnung an die SQL-Anfragesprache zu schaffen, welche bei relationalen Datenbanken verwendet wird. Der Vorteil durch die Verwendung einer auf SQL-basierenden deklarativen Sprache besteht in der Optimierungsmöglichkeit, da die Anzahl

der zur Verfügung stehende Operatoren in der deklarativen Sprache durch die Sprache selbst beschränkt ist und die zur Verfügung stehenden Operatoren eine durch die relationale Algebra klar definierte Semantik aufweisen.

```
SELECT l.x AS x,l.y AS y
FROM LocationSensorA AS l[5 SECONDS ADVANCE 2 SECONDS TIME],
     LocationSensorB AS r[5 SECONDS ADVANCE 2 SECONDS TIME]
WHERE l.x=r.x AND l.y=l.y
```

Quelltext 2.1: *Beispiel einer deklarativen Anfrage in der Continuous Query Language (CQL)*

In der Beispielanfrage 2.1 werden zwei Datenströme verbunden, wenn die beiden Attribute x und y in beiden Datenströmen den gleichen Wert aufweisen und die Elemente einen überlappenden Zeitintervall haben. Hierzu werden jeweils Zeitfenster über beide Ströme definiert, die die Gültigkeit eines Stromelements festlegen. In diesem Beispiel hat der Betrachtungsrahmen eine Größe von 5 Sekunden und springt jeweils um 2 Sekunden. Dies bedeutet, dass sich der Inhalt der Fenster jeweils zu 3 Sekunden überlappt.

Deklarative Anfragesprachen werden unter anderem in PIPES [KS04], STREAM [ABB+03] und in Esper verwendet.

2.3.6.2 Prozedural Anfragesprachen

Bei der prozeduralen Anfragesprache werden Operatoren zur Verarbeitung von Daten direkt durch die Anfrage miteinander verbunden. Da hierbei die Anzahl der möglichen Operatoren nicht durch die Anfragesprache selbst begrenzt wird, können prinzipiell beliebig viele verschiedene Operatoren miteinander verknüpft werden. Eine Beispielanfrage in der

```
accessA = WINDOW({size=[5,'SECONDS'], advance=[2,'SECONDS'],
    type='time'}, LocationSensorA)
accessB = WINDOW({size=[5,'SECONDS'], advance=[2,'SECONDS'],
    type='time'}, LocationSensorB)
product = JOIN(accessA, accessB)
match = SELECT({predicate='r.x=l.x AND l.y=r.x'}, product)
out = PROJECT({attributes = ['l.x', 'l.y']}, match)
```

Quelltext 2.2: *Beispiel einer prozeduralen Anfrage in der Procedural Query Language (PQL)*

Anfragesprache PQL aus dem Datenstrommanagementsystem Odysseus [AGG+12] findet sich in Beispielanfrage 2.2. Hierbei wird jeder Operator einzeln mit dem Ausgabestrom eines anderen vorherigen Operators verknüpft.

Die Anfrage ist dabei äquivalent zu der Anfrage in Beispielanfrage 2.1. In der prozeduralen Anfragesprache PQL werden jedoch die Operatoren einzeln miteinander verknüpft, so dass zunächst die Fenster über die beiden Ströme definiert werden und anschließend über den *Join*-Operator die Ströme verbunden werden. Der *Select*-Operator filtert anschließend die Elemente heraus, die dem Selektionskriterium, welches über den Parameter *predicate* definiert wird, nicht genügen. Der anschließende *Project*-Operator bildet danach die benötigten Attribute auf den Ausgabedatenstrom ab. Hierbei ist zu beachten, dass dies nur die Beschreibung der Verarbeitung darstellt und nicht zwangsläufig die konkrete Implementierung. Auch kann sich die Reihenfolge der Auswertungen der Operatoren von System zu System unterscheiden. Ein Beispiel hierfür ist die Kombination aus *Join*-Operator und *Select*-Operator, welche in den meisten Systemen zu einem einzigen physischen Operator zusammengeführt werden.

Eine prozedurale Anfragesprache erlaubt es zudem, zusätzlich zu den temporal relationalen Operatoren, weitere Operatoren innerhalb der Verarbeitung zu nutzen. Eine Optimierung der Anfrage auf Basis der Optimierungsregeln der relationalen Algebra ist dann allerdings nur in Teilen des Anfragegraphen möglich, in dem temporal relationale Operatoren verwendet werden.

Prozedurale Anfragesprachen werden unter anderem in Aurora [ACc+03] und System S [GAW+08] verwendet.

2.3.7 Abbildung des JDL Datenfusionsprozessmodells auf die Verarbeitung in einem Datenstrommanagementsystem

Das Ziel der einzelnen Ebenen des JDL-Datenfusionsprozessmodells ist es die Semantik der Ergebnisse einer Sensorfusion klar voneinander abzugrenzen. Diese einzelnen Schritte der Verarbeitung können dabei, wie in Abbildung 2.8 dargestellt, auf die Verarbeitung innerhalb eines Datenstrommanagementsystems abgebildet werden. Hierbei stellt die unterste Stufe, die Stufe 0, die Anpassung der Sensordaten und die Aufbereitung für eine Fusion dar. Dieses kann dabei durch die Projektions- und Selektionsoperatoren der temporal relationalen Algebra ausgedrückt werden.

In den Stufen 1-2 werden Objekte gebildet und dabei Daten fusioniert. Hierbei fungieren wieder Selektion und Projektion zur Verarbeitung der Daten und Mengenoperatoren und Verbundoperatoren zur Fusion von Sensorwahrnehmungen aus mehreren Quellen. Zusätzlich können Sensorwahrnehmungen aus komplexen Sensorsystemen als Objektdatenstrom oder Situationsdatenstrom, also Datenströme, die bereits die geforderte Semantik aufweisen, in das System einfließen.

Die Stufe 3 projiziert die aktuelle Situation in die Zukunft um Rückschlüsse auf kritische Situationen zu ziehen. Auch diese Stufe lässt sich, wie bereits in [Bol11] gezeigt wurde, durch die Verwendung eines Datenstrommanagementsystems realisieren, indem eine

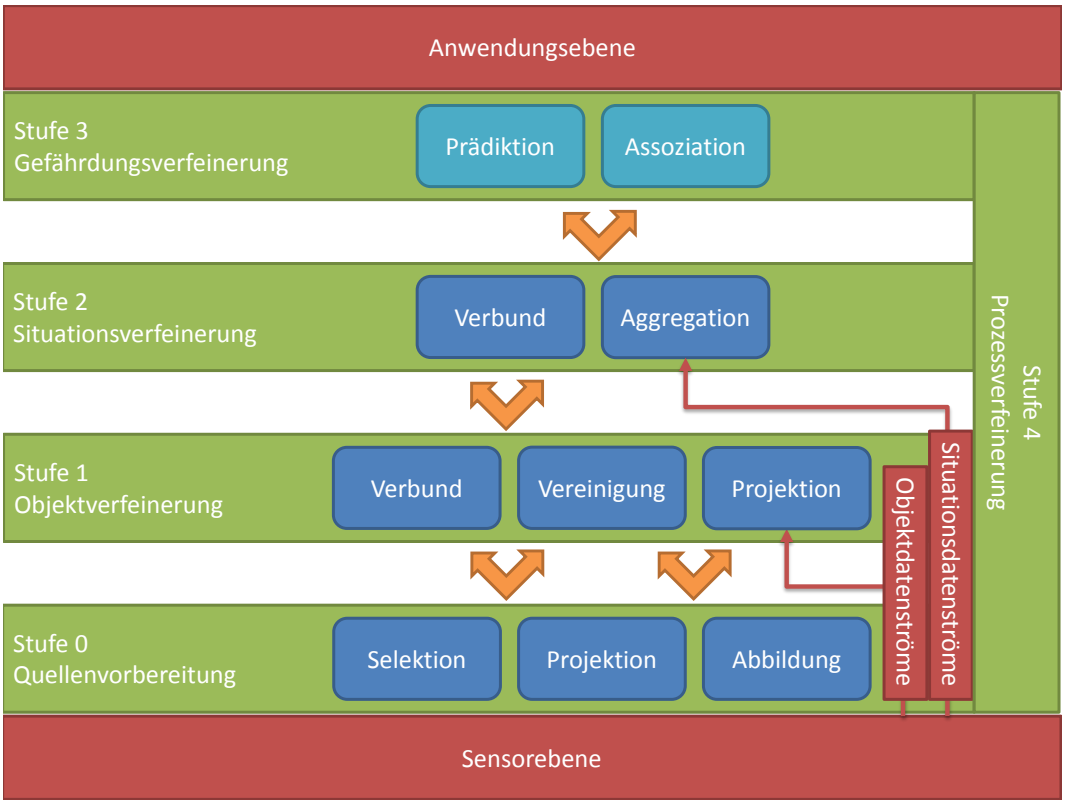


Abbildung 2.8: *Abbildung einzelner Anfragen auf die Ebenen des JDL-Datenfusionsmodells*

zweite Zeitebene zusätzlich zur Stromzeit für die Prädiktion von Zuständen verwendet wird. Die Verarbeitung findet dabei in speziellen Operatoren zur Prädiktion und Assoziation statt.

Die Stufe 4 des JDL-Datenfusionsprozessmodells überwacht den Fusionsprozess und liefert Informationen über die aktuelle Leistung. Diese Art der Überwachung ist bereits fester Bestandteil der meisten Datenstrommanagementsysteme und dient in erster Linie der optimalen Auslastung des Systems, sowie der Zuteilung von Verarbeitungsanfragen im Falle einer verteilten Systemarchitektur. Aus diesem Grund verläuft diese Stufe der Fusion vertikal zu den Verarbeitungsstufen des Datenstrommanagementsystems.

2.4 Zusammenfassung

In diesem Kapitel wurden die notwendigen Grundsteine für diese Arbeit gelegt. Hierzu wurde zunächst ein Überblick über die in den betrachteten Anwendungsszenarien verwendeten Sensoren gegeben, sowie ihre Charakteristiken hinsichtlich ihrer Wahrnehmung und Einsatzmöglichkeiten aufgezeigt.

Anschließend wurde auf das Thema Sensorfusion eingegangen und grundlegende Definitionen eingeführt, welche die Sensorfusion beschreiben. Als bekanntestes Modell zur Sensorfusion wurde das JDL-Datenfusionsprozessmodell erläutert, welches die einzelnen Schritte der Sensorfusion voneinander abgrenzt und die Semantik der, aus den Daten gewonnenen Informationen in den einzelnen Stufen der Fusion darlegt.

In Folge dessen wurde in die Basistechnologie der kontinuierlichen Datenstromverarbeitung eingeführt. Hierzu wurde das verwendete Datenmodell, sowie die darauf aufbauende temporal relationale Operatoralgebra vorgestellt. Die temporal relationale Operatoralgebra ist insofern wichtig, als dass sie im Laufe dieser Arbeit verwendet wird um dynamische Kontextmodelle zu erstellen und später dazu genutzt wird Messungen von Sensoren mit Qualitäten anzureichern und zu verarbeiten. Des Weiteren wurde aufgezeigt, wie in den meisten Datenstrommanagementsystemen Verarbeitungsanfragen formuliert werden. Hierbei wurde zwischen der deklarativen Anfragesprache und der prozeduralen Anfragesprache unterschieden und jeweils existierende Systeme genannt.

Abschließend wurde gezeigt, wie mit Hilfe der temporal relationalen Operatoren die einzelnen Stufen der Sensorfusion realisiert werden können bzw. existierende Arbeiten aufgezeigt, die zur Realisierung herangezogen werden können, um eine Sensorfusion mit Hilfe eines Datenstrommanagementsystems zu ermöglichen.

3 Dynamische Kontextmodelle

In diesem Kapitel werden die wichtigsten Grundsteine gelegt, damit Anwendungen auf Basis von Kontextmodellen auf die aktuelle Lage reagieren können. Hierzu wird zunächst der Begriff des *Kontextes* definiert und verwandte Arbeiten aufgezeigt. Anschließend wird auf Basis des Sensorfusionsprozessmodells die verschiedenen Kontextebenen erläutert und ihre Anwendung auf die zwei betrachteten Anwendungsszenarien gezeigt.

3.1 Einführung

Bevor wir beginnen ein Kontextmodell für Anwendungen zu erstellen, stellt sich zunächst die Frage, was überhaupt ein Kontext ist. Nach [Dey01] lässt sich ein Kontext wie folgt definieren:

> *„Context is any information that can be used to characterize the situation of an entity. An entity is a person, place, or object that is considered relevant to the interaction between a user and an application, including the user and applications themselves.“*

Das wichtigste Wort in dieser Definition eines Kontextes ist „relevant“. Relevant bedeutet hier, dass ein Kontextmodell nicht alle verfügbaren Informationen enthalten muss, sondern nur Informationen über Personen, Orte und Objekte, welche für die Interaktion zwischen einem Nutzer und der Anwendung von belangen sind und deren Situation charakterisieren können. Dies unterscheidet einen Kontext auch von einer Situation, die nach [YDM12] als eine externe semantische Interpretation der Sensordaten definiert ist. Interpretation bedeutet hier, dass Situationen den Sensordaten eine bestimmte Bedeutung zuordnen. Extern bedeutet, dass die Interpretation aus der Perspektive von Anwendungen stattfindet und nicht aus der Perspektive der Sensoren. Semantisch sagt aus, dass die Interpretation den Sensordaten auf der Grundlage von Strukturen und Beziehungen innerhalb der gleichen Art von Sensordaten und zwischen verschiedenen Arten von Sensordaten eine Bedeutung zuweist.

Auf Basis der Definition eines Kontextes folgt nach [Dey01] auch die nachstehende Definition eines kontextsensitiven Systems:

> *„A system is context-aware if it uses context to provide relevant information and/or services to the user, where relevancy depends on the user's task.“*

Dies bedeutet allerdings auch, dass die Informationen, welche in einem konkreten Kontextmodell enthalten sein müssen, im höchsten Masse anwendungsabhängig sind und nicht auf eine generische Menge von Informationen beschränkt werden kann. Als Kontextinformationen gelten beispielsweise im Rahmen von Autonomik-Anwendungen die Positi-

on, Richtung und Geschwindigkeit von beweglichen Objekten oder die Distanz zwischen mehreren Objekten bei der Überwachung von Arbeitsumgebungen.

3.2 Verwandte Arbeiten

Obwohl eine Vielzahl von Frameworks und Middleware-Systemen vorgestellt wurden um Kontextsensitivität und die Verwaltung von Kontextmodellen zu unterstützen (wie etwa Care [ABC+04], Nexus [GBH+05], das CML Framework [Hen03], Gaia [RHC+02], ACoMS [HIR08], der Context Broker [CFJ04], oder MAIS [CCMP06], um nur ein paar zu nennen), ist keines dieser Ansätze darauf ausgelegt, mit hochfrequenten dynamischen Kontextmodellen umzugehen, wie sie bei der Verarbeitung von Sensoren in sicherheitskritischen Anwendungen entstehen.

Die Frequenz der verwendeten Sensoren in den genannten Arbeiten liegen im Sekundenbereich und die Frequenz, mit der sich ihre Anwendungen anpassen sind typischerweise noch um einiges langsamer. Dies trifft auch auf die Arbeiten von [BMR07] zu, in der das Care-Framework für kontinuierliche Datenstromanwendungen verwendet wird. Hier ist zwar die Anwendung hochdynamisch, die Verwaltung des Kontextmodells allerdings nicht. In den hier betrachteten Anwendungsdomänen liegen die typischen Aktualisierungsfrequenzen im Bereich von 15 oder 25 Hz und da Anwendungen in diesem Bereich häufig sicherheitskritisch sind, muss eine Anwendung in der Lage sein, im Bereich von Millisekunden zu reagieren.

Allerdings existieren auch viele Ansätze, die in der Lage sind große verteilte Kontextmodelle, historische und zukünftige Kontexte und abstrahierte semantische Argumentation und Benachrichtigungsfunktionen bereitzustellen. Die hier entwickelten Ansätze betrachten dabei ein wesentlich einfacheres Kontextmodell, das typischerweise nur kürzlich verarbeitete Informationen über beispielsweise Hindernisse in der Umgebung verfügt. Daher ist der hier zu entwickelnde Ansatz komplementär zur existierenden Kontextverwaltung: Er verwaltet ein limitiertes, hochfrequentes, lokales Kontextmodell für eine kleine Menge von lokalen Anwendungen.

Da Sensorwahrnehmungen meist nur ein diskretisiertes Bild von kontinuierlichen Phänomenen in der wahren Welt liefern, ist es zumeist notwendig die so entstehende Lücke zwischen Messung durch Daten zu schließen, die den aktuellen Zustand der Welt möglichst exakt wiedergeben. Dieses kann auf Basis von Vorhersagen geschehen. Hierbei kann zwischen kurzzeitiger Vorhersage und langzeitiger Vorhersage unterschieden werden. Bei der kurzzeitigen Vorhersage wird ein kontinuierliches Phänomen zwischen zwei Sensorwahrnehmungen interpoliert. Dies dient besonders im Falle der Objektverfolgung dazu, zwei Merkmale aus unterschiedlichen Sensorwahrnehmungen miteinander zu assoziieren. Da in diesen Szenarien meist unterschiedliche Typen von Objekten verfolgt werden, werden meistens unterschiedliche Dynamikmodell und Vorhersagefunktionen verwendet [ZDM03, DMC00].

Die Langzeitvorhersage wird meist in Anwendungen wie der Kollisionsvermeidung und der Verkehrsüberwachung verwendet und arbeitet daher auf etwas längeren Zeitspannen (mehrere Sekunden oder noch länger). Die Autoren von [IWM+09] nutzen etwa die Vorhersage zur Interpolation von Messwerten in drahtlosen Sensornetzwerken zur Reduktion von Datenübertragungen bei der Kommunikation, um auf diese Weise den Energieverbrauch zu senken und die Lebensdauer zu erhöhen. In [TFPL04] verwenden die Autoren Vorhersagefunktionen auf Basis von historischen Bewegungsdaten um Anwendungen in der Verkehrsüberwachung zu unterstützen.

In all diesen Fällen werden dynamische Modelle benötigt um Vorhersagen zu treffen, wie etwa die Position eines beweglichen Objekts in der Zukunft. Viele Arbeiten beschäftigen sich mit der Entwicklung von Dynamikmodellen für spezifische Szenarien, wie etwa Fahrzeuge auf einer Schnellstraße, Fußgänger, Fahrradfahrer usw.. Eine Übersicht über Dynamikmodelle findet sich in [SFD02]. In [Bol11] wurde gezeigt, wie unterschiedliche Dynamikmodelle in einem Datenstrommanagementsystem verwendet werden können. Allerdings betrachtet dieser Ansatz nur die Verarbeitung von Positionsdaten und ist nicht generisch auf die verschiedenen Informationsbedürfnisse von kontextsensitiven Anwendungen ausgerichtet.

3.3 Kontextmodellebenen

In Kapitel 2 wurde die Fusion von Sensordaten und die JDL-Architektur als Referenzarchitektur zur Sensorfusion erläutert. Hierbei wurde unterschieden zwischen Signalen, Merkmalen und Objekten als Fusionsergebnis der einzelnen Ebenen in der JDL-Architektur. Auf die gleiche Art und Weise kann ebenfalls ein dynamisches Kontextmodell aufgebaut werden, in dem je nach Semantik der Daten und dem Informationsgehalt die Kontextinformationen in die drei Ebenen Signalebene, Merkmalebene und Objektebene gegliedert werden. Die Informationen in den einzelnen Ebenen können dabei jeweils aus der darunter liegenden Ebene mit Informationen aus zusätzlichen Quellen, welche bereits eine semantisch gleichwertige Information bereitstellen, verdichtet werden. Die drei Ebenen eines dynamischen Kontextmodells sind in Abbildung 3.1 exemplarisch dargestellt und werden im Folgenden näher erläutert.

3.3.1 Signalebene

Die Signalebene bildet die unterste Schicht des Kontextmodells ab. Aufgabe der Signalebene ist es Sensorwahrnehmungen direkt als Informationen bereit zu stellen, die in den Anwendungen häufig und vor allem schnell benötigt werden, d.h. die Informationen in dieser Kontextebene müssen eine hohe Zeitnähe und Aktualität aufweisen damit sie innerhalb einer Anwendung genutzt werden können. Bei der Verarbeitung von Sensorinformationen findet daher keine Fusion mit anderen Sensorarten statt, da im Allgemeinen unter-

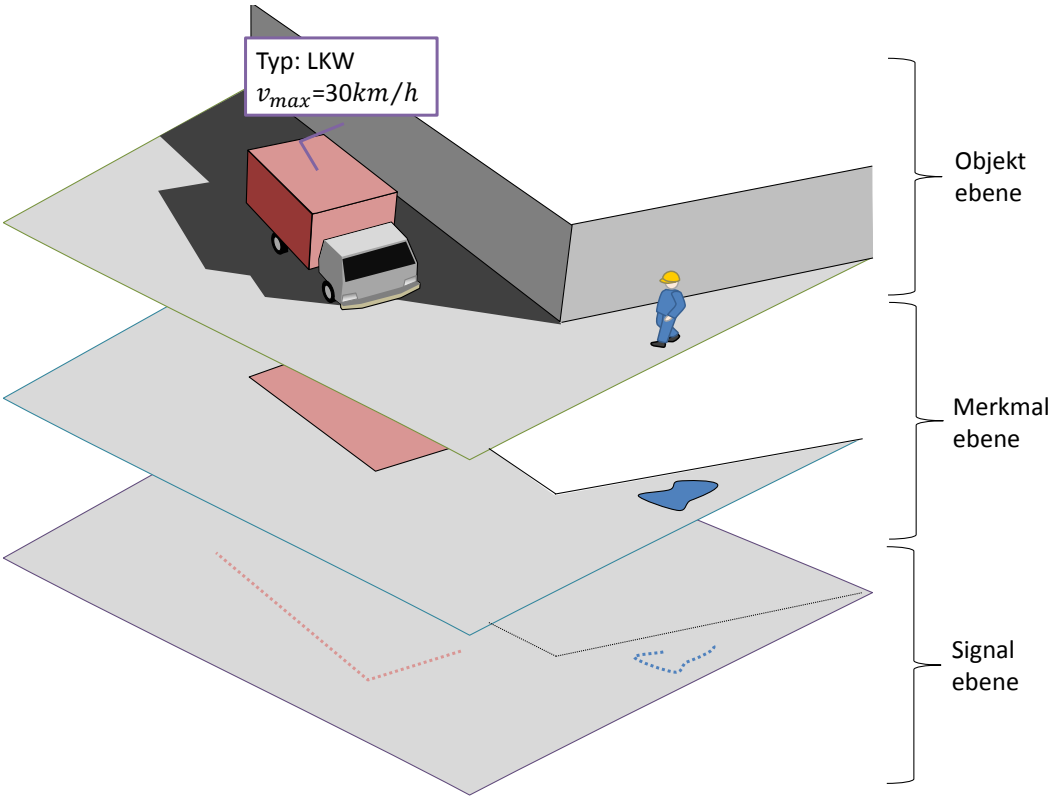


Abbildung 3.1: *Ebenen eines dynamischen Kontextmodells*

schiedliche Sensoren ihre Messdaten nicht synchron übermitteln und somit Sensordaten zurückgehalten werden müssten um sie mit anderen Sensordaten zu fusionieren, bevor sie in dieser Kontextebene zur Verfügung stehen. Im Gegensatz dazu können die Sensordaten zuvor transformiert und gefiltert werden, also von statuslosen Operatoren verarbeitet werden, bevor sie in dieser Kontextmodellebene bereit stehen. Dies ist beispielsweise notwendig wenn Sensordaten zuvor in ein globales Koordinatensystem transformiert werden müssen oder um Ausreißer in Sensormessungen frühzeitig zu filtern. Informationen in dieser Kontextebene stellen also somit die Verarbeitungsergebnisse der 0-ten Ebene des JDL-Fusionsprozesses dar. Ein Beispiel für solche Informationen sind Distanzmessungen bei beweglichen Objekten, die nicht weiter mit Informationen verdichtet werden müssen und nur für kurze Zeit als valide angesehen werden können. Ein weiteres Beispiel sind die Detektion von Flüssigkeiten oder Gasen, welche direkt innerhalb einer Anwendung einen Alarm auslösen müssen.

3.3.2 Merkmalebene

Aufbauend auf der Signalebene steht die Merkmalebene. Bei der Merkmalebene werden die Sensorwahrnehmungen von mehreren Sensoren aus der Signalebene zu Merkmalen verdichtet. Auf diese Weise lassen sich Informationen von unterschiedlichen Blickwinkeln zu einem Gesamtbild über die Umwelt kombinieren. Diese Ebene bietet zudem die Möglichkeit, auf Basis von Messungen unterschiedlicher Sensoren, auch neue Informationen zu generieren, die sich aus dem Gesamtbild der Fusion der Sensordaten ergeben. Zu den Verarbeitungsoperatoren zählen also zusätzlich zu den statuslosen Operatoren auch statusbehaftete Operatoren, wie der Verbund und die Vereinigung. Hierzu müssen die Informationen aus der Signalebene allerdings zeitlich überlappen, wodurch eine Synchronisation der Ströme notwendig ist und somit eine Verzögerung entstehen kann. Ein Beispiel für ein solches Merkmal und die dabei gewonnene Zusatzinformation sind Kanten von Objekten, die von mehreren Seiten durch Sensoren erfasst werden, und das dabei entstandene Wissen über die Ausmaße eines Objekts.

3.3.3 Objektebene

Die Informationen in der zuvor beschriebenen Merkmalebene und das dabei entstehende Gesamtbild verfügt noch über kein Wissen über die Semantik der erkannten Merkmale oder, im Falle von beweglichen Objekten, über ihre Dynamik. Hierzu bedarf es weiterer Informationen, welche in der Objektebene zusammenfließen. Die Objektebene bildet dabei die, für die Anwendung relevanten, Informationen von Objekten aus der realen Welt ab. Hierzu werden innerhalb der Objektebene wahrgenommenen Merkmale mit Hilfe von Wissen über die Anwendung mit ihrer Semantik verknüpft und zu Objekten verdichtet. Dies kann je nach Anwendungen rein auf Basis der Merkmalebene und direktem oder indirektem Anwendungswissen geschehen oder aber auch unter Verwendung zusätzlicher Quellen, wie etwa Zusatzinformationen über die Objekte aus einer Datenbank oder anderen Informationsdiensten. Auf diese Weise wird etwa aus einem erkannten Quader ein fahrendes Fahrzeug mit einem objektspezifischen Dynamikmodell.

3.3.4 Zusammenfassung

In Tabelle 3.1 findet sich ein Vergleich der einzelnen Ebenen. Hierbei gilt zu beachten, dass die Bewertungen der Geschwindigkeit mit der Informationen aktualisiert werden relativ zueinander stehen und hierbei immer noch eine Verarbeitung im Millisekundenbereich gemeint ist. Die Geschwindigkeit ist hierbei maßgeblich durch die Geschwindigkeit der verwendeten Quellen limitiert. Während bei der Signalebene keine Fusion mit anderen Sensoren stattfindet, wird in der Merkmalebene und in der Objektebene eine Fusion mehrerer Quellen benötigt um ein Gesamtbild und so ein höheren Informationsgehalt zu erhalten.

	Signalebene	Merkmalebene	Objektebene
Geschwindigkeit	hoch	mittel	niedrig
Informationsgehalt	niedrig	mittel	hoch
Fusion	Keine Fusion	Fusion mit anderen Sensordaten	Fusion mit anderen Sensordaten und historischen Daten

Tabelle 3.1: *Charakteristiken der Kontextmodellebenen*

Eine weitere Ebene, auf welche hier nicht weiter eingegangen wurde, ist die Situationsebene, welche die Beziehungen zwischen Objekten abbildet. Diese Abgrenzung geschieht deswegen, weil solche Informationen für die betrachteten Anwendungen nicht relevant sind.

3.4 Anwendung von Kontextmodellen

Dynamische Kontextmodelle spielen in einer Vielzahl von Anwendungen eine Rolle. Zu Beginn wurden bereits die beiden Anwendungen fahrerlose Transportsysteme und Offshore-Operationen kurz vorgestellt. Im Folgenden werden nun die beiden Anwendungen genauer betrachtet und hinsichtlich ihrer relevanten Kontextinformationen untersucht.

3.4.1 Fahrerlose Transportsysteme

Bei einem fahrerlosen Transportsystem handelt es sich um ein Fahrzeug, welches auf Basis seiner Sensorik und unter Zuhilfenahme von externen Sensoren autonom manövriert, beschleunigt und in kritischen Situationen abbremst oder ausweicht. Fahrerlose Transportsysteme werden heute in einer Vielzahl von Anwendungen eingesetzt, angefangen von dem klassischen Transport in Warenhäusern bis hin zur Unterstützung in Krankenhäusern. Da diese dabei auch unter anderem in Bereichen eingesetzt werden, in denen auch Personen arbeiten, ist es wichtig, dass hierbei keine Personenschäden entstehen. Hierfür benötigen diese Fahrzeuge ein aktuelles Kontextmodell auf dem sie ihre zukünftigen Operationen planen können. Zu den wichtigsten Operationen im Bereich der fahrerlosen Transportsysteme gehören das Ausweichen von Hindernissen und der rechtzeitige Nothalt, wenn kein Ausweichen möglich ist. Die, für diese Operationen notwendigen Informationen in einem Kontextmodell können dabei wie im Folgenden auf die drei vorgestellten Ebenen verteilt werden. Auf der Signalebene liegen Informationen über Distanzen zu möglichen Hindernisse in der Umgebung vor. Diese Distanzmessungen werden dabei von Distanzsensoren wie etwa Lidarsensoren oder Radarsensoren geliefert. Die Informationen auf der Signalebene können somit direkt verwendet werden um mögliche sichtbare

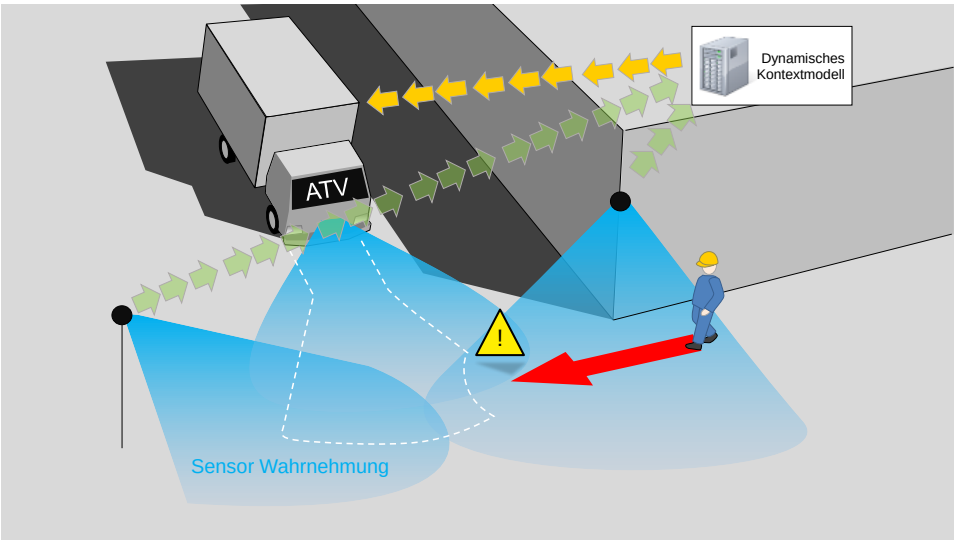


Abbildung 3.2: *Fahrerlose Transportsysteme im teil-öffentlichen Bereich*

Hindernisse auf der aktuellen Trajektorie zu detektieren. Somit kann die Signalebene des Kontextmodells direkt verwendet werden um die Geschwindigkeit des Fahrzeugs zu steuern und einen Nothalt auszulösen. Auf der Merkmalebene werden diese Hindernisse mit Hilfe von Sensoren in der unmittelbaren Umgebung des Fahrzeugs, wie es in dem konkreten Anwendungsszenario in Abbildung 3.2 dargestellt ist, zu Merkmalen fusioniert, um so die Ausmaße eines möglichen Hindernisses zu erfassen. Auf Basis dieser Informationen in dem dynamischen Kontextmodell kann ein fahrerloses Transportsystem Ausweichmanöver bei statischen Objekten planen und durchführen. Statische Objekte deswegen, da bei dynamischen Objekten die Richtung und die Geschwindigkeit auf der Merkmalebene nicht bekannt ist und somit eine Planung der Trajektorie nicht genügend relevante Information hat um eine sichere Trajektorie zu bestimmen. Die Objektebene wiederum enthält Informationen über die Art der Hindernisse und ihre Dynamik. Auf Basis der Informationen aus der Merkmalebene und historischen Beobachtungen können sowohl die Richtung, sowie die Geschwindigkeit von Objekten bestimmt werden. Durch diese zusätzlichen Informationen können anschließend fahrerlose Transportsysteme ihre Trajektorien auf die Dynamik der Objekte abstimmen und so effizienter eine Ausweichroute bestimmen.

In Tabelle 3.2 sind die beschriebenen Operationen noch einmal mit den für diese Aufgaben notwendigen Ebenen aufgezeigt.

Im Folgenden soll nun gezeigt werden, wie die Verarbeitung von Sensoren zur Erstellung eines exemplarischen Kontextmodells für fahrerlose Transportsysteme in ein Datenstrommanagementsystem integriert werden kann. Zur Detektion von Hindernissen innerhalb der Signalebene genügen dabei Transformations- und Filterfunktion, welche die Sensordaten in das lokale Koordinatensystem transformieren und die Distanz zwischen dem

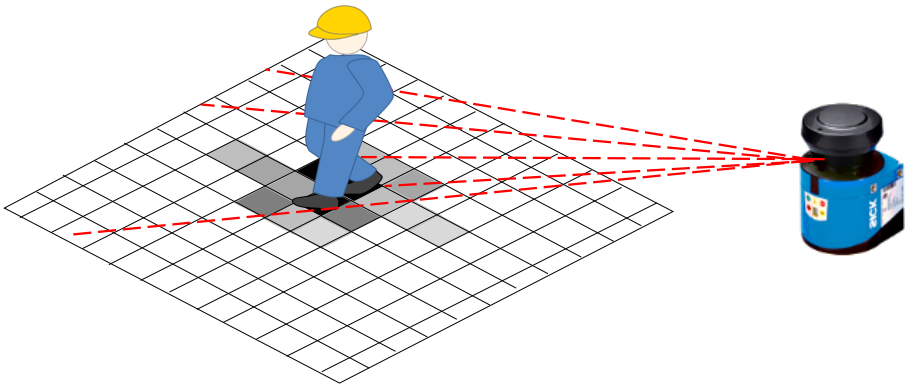


Abbildung 3.3: *Karte mit Belegungswahrscheinlichkeiten*

eigenen Objekt und einem möglichen Hindernis kontinuierlich überprüfen. Zur Transformation werden innerhalb des Datenstrommanagementsystems trigonometrische Abbildungsfunktionen verwendet. Für die Filterung der Distanz zu einem Hindernis wird eine Distanzfunktion verwendet, sowie ein Selektionsoperator welcher abhängig von einem Schwellwert Messungen entweder verwirft oder sie an die Ausgabe zur Auslösung des Nothalts weiterleitet.

Für die Merkmalebene wird eine Belegungskarte (Abb. 3.3) verwendet, auf deren Basis eine Planung der Trajektorien durchgeführt werden kann. Eine Belegungskarte ist eine diskrete Repräsentation, welche in jeder Zelle die Wahrscheinlichkeit für eine Belegung der Fläche in der wahren Welt hält. Ein Gitter erlaubt eine schnelle Verarbeitung und bietet dennoch genug Informationen über die Existenz von Hindernissen in der Umgebung des Fahrzeugs. Mit einer zuvor festgelegten Menge von Diskretisierungspunkten

Aufgabe	Signalebene	Merkmalebene	Objektebene
Nothalt	⊗		
Ausweichen statischer Hindernisse	⊗	⊗	
Ausweichen dynamischer Hindernisse	⊗	⊗	⊗

Tabelle 3.2: *Verwendung von Kontextmodellebenen in der Anwendung Fahrerlose Transportsysteme*

$X = \{x_1, \ldots, x_{N_x}\}$, $Y = \{y_1, \ldots, y_{N_y}\}$, referenzieren wir eine Zelle $\{(x, y) \in \mathbb{R}^2 | x_i \leq x < x_{i+1}, y_j \leq y < y_{j+1}\}$ mit c_{x_i,y_j}. Analog ist $P(c_{x,y})$ die Wahrscheinlichkeit mit der eine Zelle $c_{x,y}$ belegt ist.

Zur Kombination der Sensorinformationen werden die folgenden 4 Schritte durchgeführt:

1. Umwandlung des globalen kartesischen Gitters in ein, um den Distanzsensor zentriertes, Polargitter.
2. Berechnung der Abstandsverteilung induziert durch die globale Belegungskarte.
3. Integrieren der Abstandsinformationen von einem Sensor mit Hilfe des Satzes von Bayes ähnlich wie der, von [CPL+06, FSL07] verwendete.
4. Umwandlung des aktualisierten Polargitters zurück in das globale kartesische Gitter, welches die Sensorinformationen beinhaltet.

Wenn wir mit einer initialen Belegungskarte starten, können wir die Messwerte eines Distanzsensors, oder sogar eines beliebigen Sensors, in eine existierende Belegungskarte einarbeiten, indem wir diese Belegungskarte zunächst in ein Polargitter $P(c_{r,\phi})$ indiziert bei den Abständen r und den Winkeln ϕ, analog zu dem kartesischen Gitter, transformieren. Hierbei zentrieren wir das Polargitter um die globale Position des jeweiligen Sensors, um im Folgenden das Messrauschen der Distanzsensoren zu berücksichtigen. Da sich die Zellen eines Polargitters nur teilweise mit den Zellen des globalen kartesischen Gitters überlappen, weisen wir einen Wahrscheinlichkeitswert einer Zelle zu, indem pessimistisch die maximale Wahrscheinlichkeitsmasse aller überlappenden Zellen bestimmt wird.

$$P(c_{r,\phi}) = \max_{c_{x,y}: c_{x,y} \cap c_{r,\phi} \neq \emptyset} P(c_{x,y})$$

Deshalb wird einer Zelle $c_{r,\phi}$ im Polargitter die maximale Wahrscheinlichkeit aus allen Zellen im kartesischen Gitter zugewiesen die diese Zelle überlappen.

Im zweiten Schritt wird die Distanzverteilung entlang jedes Winkels ψ induziert durch das Polargitter berechnet. Dies kann mit Hilfe folgender Formel geschehen:

$$P(s|\psi) = P(c_{s,\psi}) \prod_{r' < s} \left(1 - P(c_{r',\psi})\right),$$

wobei s der Abstand vom Ursprung (in diesem Fall die Position des Sensors) bis zum nächsten Objekt entlang des Strahls mit Winkel ψ ist. Um einen Abstand s zu erzeugen

muss jede Zelle entlang des Strahls mit Winkel ψ frei sein (Produktterm) und die Zelle mit Abstand S muss belegt sein (Erster Term).

Um die Informationen der Distanzsensoren einzuarbeiten wird mit Hilfe des Satzes von Bayes die A-posteriori-Verteilung der Abstandsverteilung konditioniert auf die Sensormesswerte berechnet.

$$p(s|d(\psi)) = \frac{\rho(d(\psi)|s,\psi)P(s|\psi)}{\sum_{r'} \rho(d(\psi)|r',\psi)P(r'|\psi)} \tag{3.1}$$

Hierbei ist $d(\psi)$ der Abstand, wie er von einem Sensor in Richtung ψ gemessen wurde. $\rho(d(\psi)|s,\psi)$ ist die Wahrscheinlichkeit, dass der Sensor den Abstand $d(\psi)$ misst obwohl der wahre zugrunde liegende Abstand s ist. Die Wahrscheinlichkeit gibt daher die Zuverlässigkeit der Sensormessungen an und muss nach den Angaben in der Sensorspezifikation festgelegt werden.

Die Aktualisierung in Gleichung (3.1) muss für jeden Winkel $\psi_1, \ldots, \psi_n$, für welche Distanzmessung verfügbar sind, ausgeführt werden. Ist die A-posteriori-Verteilung für die Distanzen berechnet, kann das entsprechende Polargitter $P(c_{r,\phi}|d(\psi_1), \ldots, d(\psi_n))$ bestimmt werden.

$$\begin{aligned} P(c_{r,\psi}|d(\psi)) &= \underbrace{P(c_{r,\psi}|s=r,d(\psi))}_{=1} P(s=r|d(\psi)) \\ &+ \underbrace{P(c_{r,\psi}|s<r,d(\psi))}_{P(c_{r\psi})} P(s<r|d(\psi)) \\ &+ \underbrace{P(c_{r,\psi}|s>r,d(\psi))}_{=0} P(s>r|d(\psi)) \\ &= P(r|d(\psi)) + P(c_{r,\psi}) \cdot \left(\sum_{s<r} p(s|\psi,d) \right) \end{aligned} \tag{3.2}$$

Diese Schritte werden für jede Distanzmessung für die Winkel $\psi_1, \ldots, \psi_n$ ausgeführt um ein aktualisiertes Polargitter $P(c_{r,\phi}|d(\psi_1), \ldots, d(\psi_n))$ zu erhalten. Dies ist insofern gerechtfertigt, als dass die Winkel ψ_i eines jeden Laserstrahls sich nicht überlappen und somit die Messungen als unabhängig betrachtet werden können. Um das globale kartesische Gitter zu aktualisieren und somit das dynamische Kontextmodell, müssen die Polargitter zurück in das kartesische Gitter transformiert werden. Hierzu wird ein pessimistischer Ansatz verwendet, bei dem für den ersten Schritt die Maximalwahrscheinlichkeit aller überlappenden Zellen gewählt wird.

$$P(c_{x,y}) = \max_{c_{r,\phi}:c_{x,y}\cap c_{r,\phi}\neq\emptyset} P(c_{r,\phi}|d(\psi_1),\ldots,d(\psi_n)) \tag{3.3}$$

3.4.2 Logische Integration

Im Folgenden wird die Integration der beschriebenen Erstellung einer Belegungskarte als Kontextinformation in der Merkmalebene eines Kontextmodells in ein Datenstrommanagementsystem aufgezeigt. Zu diesem Zweck wird die Erstellung und Aktualisierung der Belegungskarte in zwei Schritten vorgenommen. Schritt 1 besteht aus der Integration von Distanzmessungen eines Sensors in eine bestehende Belegungskarte. Schritt 2 besteht aus der Vorhersage des Zustandes einer existierenden Belegungskarte auf den aktuellen Zeitpunkt, um so neue Sensorwahrnehmungen in eine Belegungskarte integrieren zu können. Die Beschreibung der Semantik der beiden Schritte geschieht zunächst durch die Definition von zwei voneinander unabhängigen logischen Operatoren, dem Integrationsoperator und dem Ausbreitungsoperator. Dieses Vorgehen hat nach [Gee13] den Vorteil, dass eine einheitliche Schnittstellen zwischen Operatoren geschaffen wird. Dies erhöht wiederum die Flexibilität und erlaubt umfassende Kombinationsmöglichkeiten der Operatoren mit anderen Operatoren des Systems.

Definition 9 (Integrationsalgorithmus). Die Integrationsfunktion $f_{merge} : \{\Omega_{\mathcal{T}} \times \mathbb{N}\} \times \mathbb{C} \rightarrow \mathbb{C}$ ist eine Abbildung, die die Nutzdaten mit Schema $\mathcal{T}$, die zum Zeitpunkt t gültig sind, in ein bestehendes Kontextmodell zum Zeitpunkt t integriert und so ein neues Kontextmodell schafft.

$$\begin{aligned} f_{merge} :&\{(e,t,n)|e \in \Omega_{\mathcal{T}} \wedge n \in \mathbb{N}_{>0} \wedge t \in T\} \times \{(c,t)|c \in \mathbb{C} \wedge t \in T\} \\ &\rightarrow \{(\tilde{c},t)|\tilde{c} \in \mathbb{C} \wedge t \in T\} \end{aligned} \tag{3.4}$$

Zur Ausführung der Integrationsfunktion wird ein logischer Operator benötigt, der diese Funktion auf den logischen Datenstrom anwendet.

Definition 10 (Logischer Integrationsoperator). Der Integrationsoperator $\mu_{f_{merge}} : S^l_{\mathcal{T}} \times \mathbb{F}_{merge} \rightarrow S^l_{\tilde{\mathcal{T}}}$ ist eine Abbildung, die einen logischen Datenstrom mit Schema $\mathcal{T}$ in das bestehende Kontextmodell mit Hilfe eines Integrationsalgorithmus integriert und das Ergebnis wieder in einen logischen Datenstrom mit Schema $\tilde{\mathcal{T}}$ überführt. Sei $S \in \mathbb{S}^l_{\mathcal{T}}$ dann gilt:

$$\begin{aligned} \mu_{f_{merge}}(S) := \{(\hat{e},t,1)|\hat{e} := f_{merge}(X) \\ \wedge X := \{(e,t,n)|(e,t,n) \in S\}\} \end{aligned} \tag{3.5}$$

Durch den definierten Operator lassen sich nun Sensormessungen in das Kontextmodell integrieren. Da allerdings Sensoren nur ein diskretes Bild über ihre Umgebung liefern ist

es notwendig, zwischen zwei Messungen das Kontextmodell auf den aktuellen Zeitpunkt voraus zu berechnen. Dies geschieht mit Hilfe einer Ausbreitungsfunktion.

Definition 11 (Ausbreitungsalgorithmus). Die Ausbreitungsfunktion $f_{spread} : \mathbb{C} \times T \rightarrow \mathbb{C}$ ist eine Abbildung, die ein bestehendes Kontextmodell zum Zeitpunkt t in ein Kontextmodell zum Zeitpunkt $\tilde{t}$ überführt. Der Parameter v gibt zusätzlich die maximale Geschwindigkeit vor, mit der sich Objekte in der jeweiligen Anwendung maximal bewegen können.

$$\begin{aligned} f_{spread} :&\{(c,t,v)|c \in C \wedge t \in T \wedge v \in \mathbb{R}_{\geq 0}\} \times \tilde{t} \in T \\ &\rightarrow \{(\tilde{c},\tilde{t})|\tilde{c} \in \mathbb{C} \wedge \tilde{t} \in T\} \end{aligned} \tag{3.6}$$

Zur Anwendung der Ausbreitungsfunktion wird wieder ein logischer Operator verwendet, der diese Berechnung des Kontextmodells auf Basis eines logischen Datenstroms durchführt.

Definition 12 (Logischer Ausbreitungsoperator). Der Ausbreitungsoperator $\mu_{f_{spread}} : S^l_{\mathcal{T}} \times \mathbb{F}_{spread} \rightarrow S^l_{\tilde{\mathcal{T}}}$ ist eine Abbildung, die einen logischen Datenstrom mit Schema $\mathcal{T}$ aus dem Kontextspeicher mit Hilfe der Ausbreitungsfunktion auf den aktuellen Zeitpunkt vorausberechnet und das Ergebnis wieder in einen logischen Datenstrom mit Schema $\tilde{\mathcal{T}}$ überführt. Sei $S \in \mathbb{S}^l_{\mathcal{T}}$ gilt:

$$\begin{aligned} \mu_{f_{spread}}(S) := \{(e,\hat{t},1)|e :=& f_{spread}(X,\hat{t},v) \\ &\wedge X := \{(c,t)|c \in \mathbb{C} \wedge t \in T\}\} \end{aligned} \tag{3.7}$$

3.4.3 Physische Integration

Zur Anwendung der beschriebenen logischen Verarbeitung in einer konkreten Implementierung wird im Folgenden die physische Verarbeitung durch die beiden Abbildungsfunktionen erläutert.

3.4.3.1 Physischer Ausbreitungsoperator

Hierzu wird zunächst der physische Ausbreitungsoperator in Listing 3.1 beschrieben. Als Parameter erhält dieser Operator zunächst das aktuelle Kontextmodell, sowie den letzten Zeitpunkt zu dem dieses Kontextmodell aktualisiert wurde. Zusätzliche wird über den dritten Parameter dem Operator mitgeteilt, wie schnell sich Objekte in der betrachteten Umwelt maximal bewegen können. Im Falle von fahrerlosen Transportsystemen ist dies etwa die maximale Geschwindigkeit der Fahrzeuge oder der Personen, die sich in dem betrachten Gebiet aufhalten.

Anschließend wird über jedes Element im Eingangsstrom S_{in} die zeitliche Differenz zwischen der Messung und dem letzten Aktualisierungszeitpunkt der Belegungswahrscheinlichkeiten des Kontextmodells bestimmt umso festzulegen, wie viele Zellen maximal von

Algorithmus 3.1 : Integration des Ausbreitungsoperators

Input : Physischer Eingangsdatenstrom S_{in}
Input : Kontextmodell mit den Belegungswahrscheinlichkeiten P
Input : Zeitstempel des Kontextmodells t_P
Input : Geschwindigkeit v
Output : Physischer Ausgangsdatenstrom S_{out}

$S_{out} \leftarrow \emptyset$
for $s := (e, [t_S, t_E)) \leftarrow S_{in}$ **do**
  // Maximale Anzahl von Zellen die ein Objekt in dieser Zeit erreichen kann
  $t_\Delta \leftarrow t_S - t_P$
  $|c| \leftarrow 2t_\Delta v + 1$
  // Erstelle einen Kernel der Größe $|c| \times |c|$ und setze jede Zelle auf 1
  $K[c, c] \leftarrow 1$
  // Addiere die umliegenden Auftrittswahrscheinlichkeiten zu jeder Zelle im Kontextmodell
  $P(c_{x,y}) \leftarrow \exp^{1-\sum_i^{|c|}\sum_j^{|c|}\log(1-P(c_{x+i-a_i,y+j-a_j}))K[i,j])}$
  $(P, [t_S, t_E)) \rightarrow S_{out}$
end

Objekt erreicht werden können. Anschließend wird mit der Anzahl der Zellen eine Kernel-Operation ausgeführt um die Summe jeder Zelle auf Basis der umliegenden Zellen, welche durch den Kernel ausgewählt werden, zu berechnen. Die Summierung ist dabei die Multiplikation der umliegenden Wahrscheinlichkeiten auf den logarithmischen Werten der Wahrscheinlichkeit. In dem hier beschriebenen Ausbreitungsoperator wird dabei davon ausgegangenen, dass eine Ausbreitung in jede Richtung stattfinden kann. Ist allerdings die Objektklasse zuvor bekannt oder findet die Ausbreitung in der Objektebene statt kann auf ein vorhandenes Dynamikmodell zurückgegriffen werden. Dieses wird allerdings in dieser Arbeit nicht weiter behandelt.

3.4.3.2 Physischer Integrationsoperator

Der physische Integrationsoperator in Listing 3.2 nimmt ein Nutzdatentupel mit Sensormessungen entgegen und integriert sie in ein bestehendes Kontextmodell. Zunächst wird hierzu eine Kopie der bisherigen Belegungskarte erstellt, diese dient dazu, die neuen Messwerte einzuarbeiten ohne die Werte der bisherigen Belegungskarte zu verlieren. Anschließend wird in dieser neuen Belegungskarte der Bereich, der von dem Sensor erfasst wurde, als *Frei* deklariert. Im Anschluss wird für jeden einzelnen Messwert jeweils die Wahrscheinlichkeiten pro Zelle in einem Polargitter bestimmt.

Hierzu wird bei jedem Strahl und jedem Abschnitt in dem Polargitter die maximale Belegungswahrscheinlichkeit in der überlappenden Fläche in dem Belegungsgitter bestimmt.

Algorithmus 3.2 : Integration des Integrationsoperators

Input : Physischer Eingangsdatenstrom S_{in}
Input : Kontextmodell mit den Belegungswahrscheinlichkeiten P
Input : Sensorwahrnehmung D
Input : Position des Sensors $p_{x,y}$
Input : Standardabweichung des Sensors σ
Output : Physischer Ausgangsdatenstrom S_{out}

$S_{out} \leftarrow \emptyset$
for $s := (e, [t_S, t_E)) \leftarrow S_{in}$ **do**
  $P' \leftarrow P$
  $P'(D) \leftarrow FREE$
  for $d(\phi) \in D$ **do**
    $p \leftarrow 1$
    for $r \in R$ **do**
      $b_r \leftarrow \max\{P(c_{x,y}) | (x,y) \in area(p_{x,y}, r, \phi)\}$
      $P(c_{r'}, \phi) \leftarrow p b_r \int_r \mathcal{N}(d(\phi), \sigma) dx$
      $p \leftarrow p(1 - P(c_{r',\phi}))$
    end
    $\hat{p} \leftarrow 0$
    for $r \in R$ **do**
      $\overline{p} = \frac{P(c_{r'},\phi)}{\sum_i P(c_i,\phi)}$
      $v \leftarrow \overline{p} + (b_r \hat{p})$
      $P'(c_{x,y}) \leftarrow \max v, P'(c_{x,y})$
      $\hat{p} = \hat{p} + \overline{p}$
    end
  end
  $(P', [t_S, t_E)) \rightarrow S_{out}$
end

Die Fläche wird dabei durch die Funktion *area* berechnet, welche als Parameter die Position des Sensors, den aktuellen Polargitterabschnitt und den Winkel ϕ benötigt. Auf Basis dieser maximalen Belegungswahrscheinlichkeit, der vorhandenen Wahrscheinlichkeitsmasse p, sowie der Wahrscheinlichkeitsdichte der Normalverteilung mit dem Messwert als Erwartungswert, wird nun die neue Belegungswahrscheinlichkeit in dem Polargitter bestimmt.

Diese Berechnung der Wahrscheinlichkeiten auf jedem Strahl kann dabei parallel ausgeführt werden, da sich die Messwerte im Polargitter nicht überlappen. Allerdings muss dafür bei der Rücktransformation in das globale Belegungsgitter eine Synchronisation stattfinden, da sich diese Zellen wiederum überlappen können.

3.4.3.3 Zusammenspiel der Verarbeitungsschritte eines Kontextmodells in einem Datenstrommanagementsystem

Die beschriebenen Ebenen können nun zusammen in einem Datenstrommanagementsystem verwendet werden. Hierbei fokussieren wir uns auf die Verarbeitung der Sensordaten auf dem fahrerlosen Transportsystem, da hierfür sowohl die Informationen auf der Signalebene, wie auch die Informationen auf der Merkmalebene relevant sind. Die Sensorik in der Umgebung stellt zwar auch die Informationen der Merkmalebene bereit, jedoch sind hier die Informationen der Signalebene nicht relevant, da die Entscheidung über einen Nothalt in diesem Szenario nur lokal, auf dem Fahrzeug, getroffen wird.

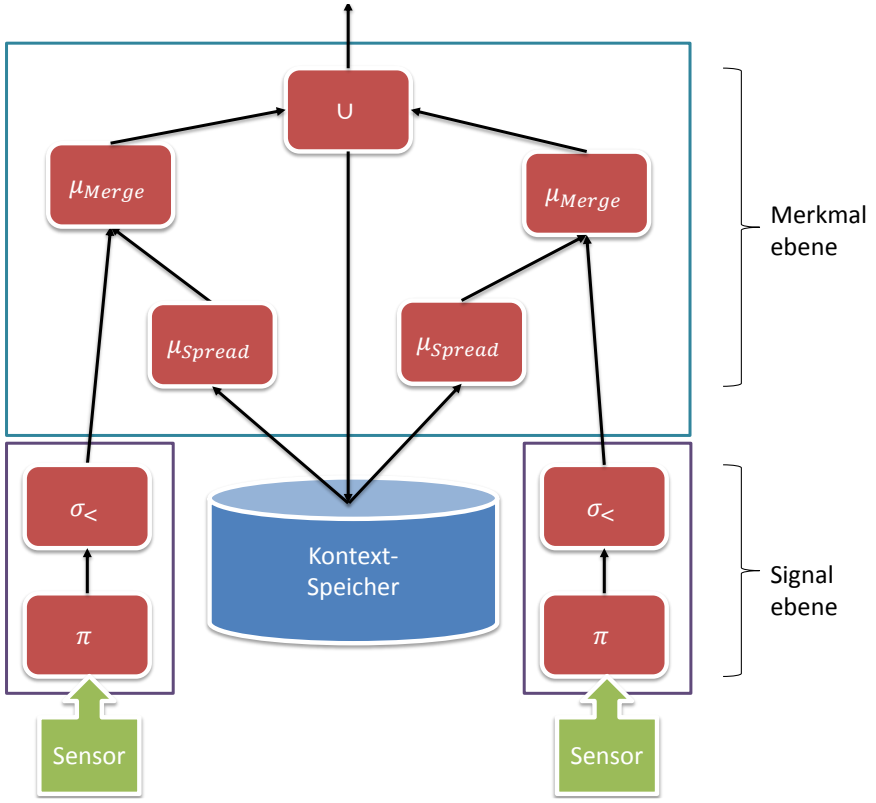


Abbildung 3.4: *Anfrageplan für die kontinuierliche Bereitstellung von Kontextinformationen für das Anwendungsszenario Fahrerlose Transportsysteme*

In Abbildung 3.4 ist der komplette Verarbeitungsgraph als logischer Anfragegraph für die Erstellung des dynamischen Kontextmodells für das Anwendungsszenario abgebildet. Im unteren Bereich des Graphen sind dabei die Transformations- und Filterfunktionen angebracht, die direkt eine Ausgabe erzeugen, wenn ein Hindernis zu nah an dem Objekt ist. Diese Transformations- und Filterfunktionen repräsentieren die Signalebene. Ihre Aufga-

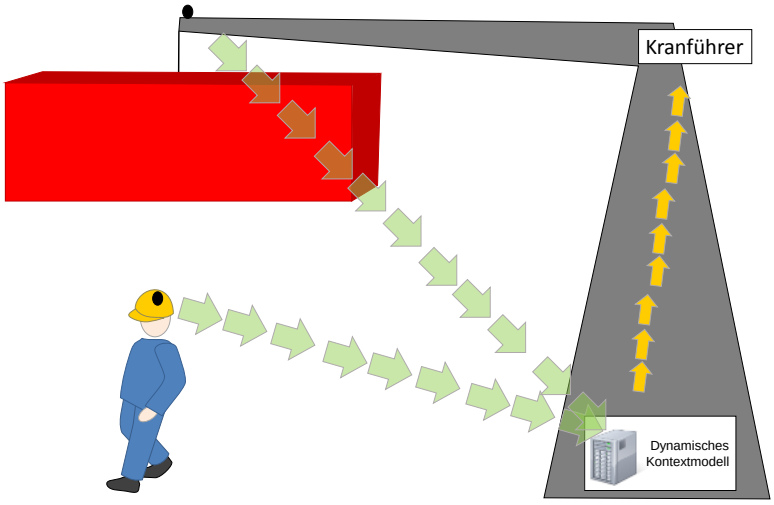


Abbildung 3.5: *Überwachung von Offshore-Operationen durch Sensoren*

be ist es die gemessenen Entfernungsdaten der Sensoren in das lokale Koordinatensystem des Transportsystems zu transformieren und auf Basis der Entfernung zu filtern.

Die Daten der Sensoren laufen nach dem Transformationsoperator noch in eine weitere Abbildungsfunktion, welche die Integration der Daten in die bisherige globale Belegungskarte durch Zuhilfenahme eines temporären Kontextspeichers vornimmt. Der temporäre Kontextspeicher dient hierbei dazu, Zwischenergebnisse der Verarbeitung temporär zu speichern um sie so in anderen Verarbeitungszyklen zu nutzen. Die Grundlagen zur Verwendung eines Kontextspeichern finden sich in [Bol11] und werden hier nicht weiter vertieft. Der Operator nach dem Kontextspeicher nimmt dabei direkt die Funktion des Ausbreitungsoperators vor, wodurch die Integration der Daten in eine bereits vorverarbeitete Belegungskarte stattfinden kann. Das Ergebnis dieser Verarbeitung wird direkt wieder in den Kontextspeicher und in die Ausgabe an die Anwendung übertragen. Der zusätzliche Vereinigungsoperator im oberen Teil des Graphen dient dazu, dass die Veränderungen an der globalen Kontextkarte zeitlich sortiert in den Kontextspeicher und an die Anwendung geschrieben werden, um so die zeitliche Ordnung innerhalb des physikalischen Stroms sicherzustellen. Auf diese Weise ist garantiert, dass neue Kontextinformationen nicht durch alte Informationen fälschlicherweise überschrieben werden. In Folge der Integration der Distanzmessungen in die globale Belegungskarte findet zudem der Übergang in die Merkmalebene statt. Die Ergebnisse der Verarbeitung können nun also dazu verwendet werden, auf Basis der Belegungskarte statischen Objekten auszuweichen.

3.4.4 Sichere Offshore-Operation

Als zweite Anwendung eines dynamischen Kontextmodells betrachten wir im Weiteren die Anwendung Sichere Offshore-Operationen. Unter einer Offshore-Operation versteht man Arbeiten auf hoher See oder in Küstengegenden, wie etwa die Wartung von Unterwasserpipelines, die Konstruktion von Windkraftwerken, dem Cargo-Betrieb (Frachtbeladung und -löschung, Öffnen / Schließen von Ventilen) und Tank-Operationen (Tankreinigung, Umgang mit Schmutzwasser).

Nach dem HSE Offshore Safety Statistics Bulletin 2011/12 [HSE12] kam es, basierend auf vorläufigen Zahlen für 2011/12, hierbei zu 36 schweren Verletzung. Die Gesamtzahl der schweren und mittelschweren Verletzungsrate beläuft sich dabei auf 130,77 pro 100.000 Arbeiter in 2011/2012. 86% aller Verletzungen lassen sich auf das Ausrutschen/Stürzen, Verletzungen durch bewegliche Objekte und Verletzungen in Verbindung mit der Handhabung von Lasten zurückführen. Um die Sicherheit von Arbeitern bei Offshore-Operationen zu überwachen und vor möglichen Gefahren zu warnen benötigt eine Anwendung, wie etwa ein Assistenzsystem, unterschiedliche relevante Informationen. Zu den relevanten Informationen zählen die Positionen der Arbeiter, die Positionsdaten von beweglichen Frachten, sowie Daten über den Grad der Verschmutzung von Laufwegen durch Flüssigkeiten. Zudem sind auch die Sensoren und ihre Wahrnehmungen nicht immer akkurat, da sie zum einen durch die dort herrschenden Wetterbedingungen starken Störungen ausgesetzt sind und durch die in diesem Bereich verwendeten Baumaterialien beeinträchtigt werden. Besonders Radiosignale zur Kommunikation und Ortung von Personen sind durch den Einsatz von Stahl im Schiffbau und auf Plattformen in ihrer Funktionsweise und Qualität beeinträchtigt. Daher werden auch Daten über aktuelle Wetterbedingung benötigt.

Bei dem hier betrachteten Anwendungsszenario, welches in Abbildung 3.5 dargestellt ist, werden die Positionen der Mitarbeiter durch Funksensoren erfasst. Bei der Verarbeitung dieser Daten zu relevanten Kontextinformationen können die Positionen der Mitarbeiter und Frachten direkt auf der Objektebene behandelt werden, da ein implizites Wissen über den jeweiligen Sensor existiert. Dagegen sind der Grad der Verschmutzung von Laufwegen, Wetterdaten und der aktuelle Seegang auf der Signalebene anzusiedeln. Auf Basis dieser Informationen können die Objektdaten weiter angereichert werden, etwa um die zu überwachenden Sicherheitsbereiche zu vergrößern.

In Tabelle 3.3 sind die beschriebenen Aufgaben in diesem Anwendungsszenario mit den für diese Aufgaben notwendigen Ebenen des Kontextmodells aufgezeigt.

Für die Erstellung des Kontextmodells müssen in diesem Fall zum einen auf Basis von Sensoren die Verschmutzung der Wege überprüft werden und die Distanzen zwischen Mitarbeitern und beweglichen Lasten bestimmt werden. Die Verschmutzung der Wege ist dabei direkt über einen Sensor möglich und bedarf keiner komplexen Verarbeitung. Die Verarbeitung findet also direkt auf der Signalebene statt und kann ähnlich zu der Verarbeitung im vorherigen Szenario über einen Selektionsoperator ermöglicht werden. Aus

Aufgabe	Signalebene	Merkmalebene	Objektebene
Verschmutzter Fußboden	⊗		
Verlassen von sicheren Pfaden	⊗	⊗	
Personen unter Lasten	⊗	⊗	
Vorwarnen vor Out-of-sight Situationen bei Verladeoperationen	⊗	⊗	⊗

Tabelle 3.3: *Verwendung von Kontextmodellebenen in der Anwendung Sichere Offshore-Operation*

diesem Grund wird diese Verarbeitung hier nicht weiter betrachtet. Für die Distanzen zwischen Mitarbeitern und beweglichen Lasten müssen deren Positionen verarbeitet werden. Da sowohl die Personen, wie auch die Lasten jeweils mit Positionssensoren ausgestattet sind, können deren Messwerte direkt als Informationen innerhalb der Objektebene behandelt werden.

Zur Bestimmung der Distanz kann hierzu eine Abstandfunktion im zweidimensionalen Raum verwendet werden. Um zu bestimmen, ob sich zwei Objekte, wie etwa der Mitarbeiter und eine schwebende Last, in einer kritischen Entfernung zueinander befinden existieren unterschiedliche Distanzmetriken. Je nach Qualität der verwendeten Sensoren kann etwa die Euklidische-Abstandfunktion verwendet werden um die Distanz zwischen den Objekten zu bestimmen. Dies ist der Fall, wenn die Sensoren nahezu exakte Positionsmessungen erheben können oder das Sensorrauschen zu vernachlässigen ist. Die Distanz $D(p, q)$ zweier Objekte p und q lässt sich dann wie folgt bestimmen:

$$D(p, q) = \sqrt{(q_1 - p_1)^2 + (q_2 - p_2)^2}$$

Da allerdings aufgrund der äußeren Einflüsse im Offshore-Bereich exakte Messungen von Positionssensoren häufig nicht möglich sind, bedarf es einer Distanzfunktion, welche die Unsicherheiten der verwendeten Sensoren mit in die Distanzbestimmung einfließen lassen kann. Hierbei kann unterschieden werden, ob nur ein Sensor starken äußeren Einflüssen unterliegt, oder ob beide Sensoren von äußeren Einflüssen betroffen sind. Der erste Fall trifft ein, wenn die Position von Personen über Funksensoren bestimmt wird, die Position der Last aber über Sensoren innerhalb der Lastkräne bestimmt werden kann und daher wesentlich genauer sein kann. In einem solchen Fall bietet sich die Mahalanobis-Distanz an, die die Nähe eines Punktes zu einer Verteilung bestimmt.

$$D(p, q) = \sqrt{(q - \mu_p)^T \Sigma_p^{-1} (q - \mu_p)}$$

Im zweiten Fall, wenn beide Sensoren starken äußeren Einflüssen unterliegen bedarf es einer Distanzfunktion, die beide Unsicherheiten betrachtet. Ein Beispiel für ein solches Szenario wäre die Distanz zwischen zwei Personen oder die Distanz zwischen einem Mitarbeiter und der zu transportierenden Last unter starkem Seegang, bei dem die gewöhnliche Positionsbestimmung über interne Sensoren nicht mehr möglich ist. Hierzu bietet sich eine statistische Distanzfunktion für Verteilungen an, die Bhattacharyya-Distanzfunktion, welche die Ähnlichkeit zweier Verteilungen ermittelt.

$$D(p,q) = \frac{1}{8}(\mu_1 - \mu_2)^T \Sigma^{-1}(\mu_1 - \mu_2) + \frac{1}{2}\ln(\frac{\det \Sigma}{\sqrt{\det \Sigma_1 \det \Sigma_2}})$$

mit

$$\Sigma = \frac{\Sigma_1 + \Sigma_2}{2}$$

Auf Basis dieser drei unterschiedlichen Distanzfunktionen kann nun, je nach gegebenen Umwelteinflüssen, eine kritische Situation, wie etwa die Nähe zu einer schwebenden Last, durch Anwendung eines Schwellwerts bestimmt werden. Dieser Schwellwert muss allerdings abhängig von der Distanzfunktion unterschiedlich bewerten werden, da die Euklidische-Distanz eine Strecke als Distanz ermittelt, die Bhattacharyya-Distanzfunktion dagegen ein Maß der Ähnlichkeit zweier Verteilungen.

3.4.5 Logische Integration

Zur Verwendung der Distanzfunktionen, um die Distanz zwischen den beteiligten Entitäten in dem Anwendungsszenario zu bestimmen, müssen diese zunächst wieder in ein Datenstrommanagementsystem integriert werden. Dies geschieht, wie bereits bei dem vorherigen Anwendungsszenario über einen logischen Operator. Der Distanzalgorithmus, welcher innerhalb des Operators verwendet wird, ist dabei ein generischer Algorithmus, der je nach verwendeten Parametern einen der drei Distanzfunktionen auswählt um die Distanz zwischen zwei Objekten zu bestimmen.

Definition 13 (Distanzalgorithmus). Die Distanzfunktion $f_{distance} : \Omega_{\mathcal{T}} \rightarrow \mathbb{Q}_{\geq 0}$ ist eine Abbildung, die die Nutzdaten mit Schema $\mathcal{T}$ durch Anwendung einer der beschriebenen Distanzfunktionen auf ein Ähnlichkeitsmaß der beiden in den Nutzdaten dargestellten Entitäten bestimmt.

$$f_{distanz} : \{(e,n) | e \in \Omega_{\mathcal{T}} \wedge n \in N_{>0}\} \rightarrow d \in \mathbb{Q}_{\geq 0} \tag{3.8}$$

Zur Anwendung der Distanzfunktion wird wieder zunächst ein logischer Operator verwendet, der diese Distanzfunktion auf die Elemente in dem logischen Datenstrom anwendet.

Definition 14 (Logischer Distanzoperator). Der Distanzoperator $\mu : S^l_{\mathcal{T}} \times \mathbb{F}_{distance} \rightarrow S^l_{\tilde{\mathcal{T}}}$ ist eine Abbildung, die einen logischen Datenstrom mit Schema $\mathcal{T}$ durch Anwendung einer Distanzfunktion auf einen neuen logischen Datenstrom mit dem Abstand zwischen den Entitäten und dem Schema $\tilde{\mathcal{T}}$ überführt. Sei $S \in \mathbb{S}^l_{\mathcal{T}}$ und $f_{distance} \in \mathbb{F}_{distance}$ dann gilt:

$$\begin{aligned}\mu_{f_{distance}}(S) :=& \{(\hat{e}, t, \hat{n}) | \exists X \subseteq S : X \neq \emptyset \wedge X = \{(e, t, n) \in S | \\ & f_{distance}(e) = \hat{e}\} \wedge \hat{n} = \sum_{(e,t,n) \in X} n\}\end{aligned} \tag{3.9}$$

3.4.6 Physische Integration

Die Integration der physischen Implementierung des Distanzoperators verläuft wie in Listing 3.3 dargestellt.

Algorithmus 3.3 : Integration des Distanzoperators

Input : Physischer Eingangsdatenstrom S_{in}
Input : Kontextmodell mit den Belegungswahrscheinlichkeiten P
Output : Physischer Ausgangsdatenstrom S_{out}

```
1 S_out ← ∅
2 for s := (e1 ∘ e2, [t_S, t_E)) ← S_in do
3 |   d ← D(e1, e2)
4 |   (d, [t_S, t_E)) → S_out
5 end
```

Für jedes Element aus dem Datenstrom wird jeweils die Distanzfunktion aufgerufen und das Resultat als zusätzliches Element in den Ausgabedatenstrom geschrieben. Die Funktion D ist dabei die oben beschriebene Distanzfunktion.

3.4.6.1 Zusammenspiel der Verarbeitungsschritte eines Kontextmodells in einem Datenstrommanagementsystem

Die beschriebenen Ebenen können nun zusammen in einem Datenstrommanagementsystem verwendet werden. Die Sensorik der beteiligten Akteure bieten durch die indirekte Verknüpfung zwischen Sensormesswert und Träger des Sensors die Daten bereits als Objektinformationen an und können somit direkt auf der Objektebene des Kontextmodells bereitgestellt werden.

In Abbildung 3.6 ist der komplette Verarbeitungsgraph als logischer Anfragegraph für die Erstellung des dynamischen Kontextmodells für das Anwendungsszenario Sichere Offshore-Operationen abgebildet. Im unteren Bereich des Graphen sind dabei wieder die Trans-

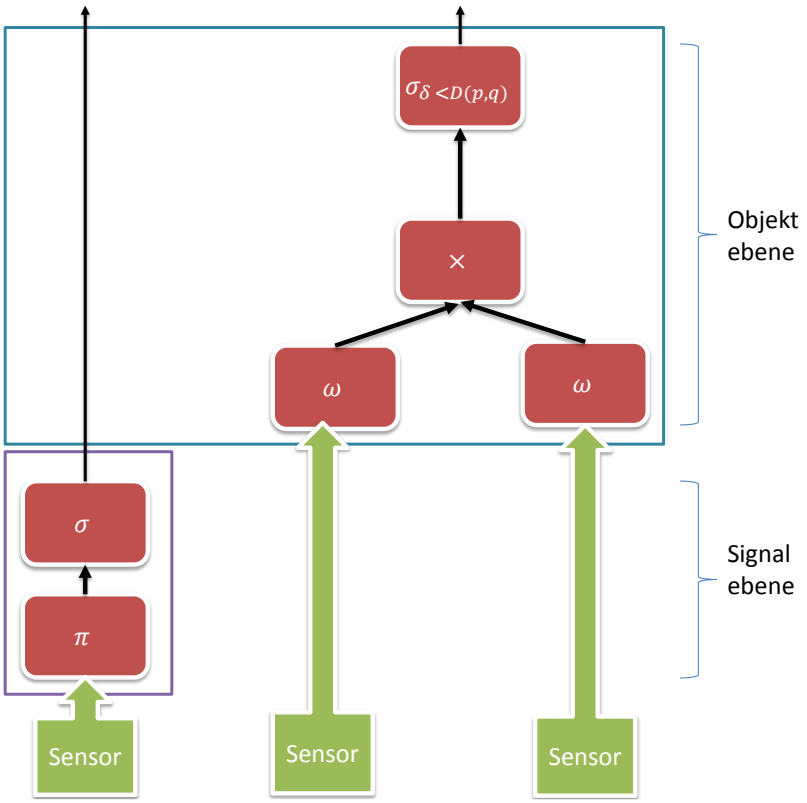


Abbildung 3.6: *Anfrageplan für die kontinuierliche Bereitstellung von Kontextinformationen für das Anwendungsszenario Sichere Offshore-Operationen*

formations- und Filterfunktionen angebracht. Diese Transformations- und Filterfunktionen repräsentieren wieder die Signalebene und sind für die Benachrichtigung über einen verschmutzen Boden zuständig.

Zusätzlich werden die Positionsdaten der Mitarbeiter und der zu bewegenden Lasten verarbeitet. Hierzu werden zunächst die aktuellen Positionsdaten konkatiniert. Dies geschieht durch die Verwendung eines Zeitfensters, welches die Gültigkeit der Stromelemente festlegt und auf diese Weise den maximalen Versatz von Sensorwahrnehmungen vorgibt. Anschließend wird über einen Selektionsoperator der resultierende Datenstrom gefiltert. Als Selektionskriterium dient die Distanzfunktion. In einer konkreten Implementierung werden dabei das Kreuzprodukt und die Filterung zusammen in einem Verbundoperator realisiert, wodurch weniger Elemente miteinander konkatiniert werden müssen.

3.5 Zusammenfassung

In diesem Kapitel wurde die grundlegende Idee eines dynamischen Kontextmodells erörtert. Zu diesem Zweck wurde zunächst der Begriff des Kontextes analysiert und existierende Ansätze, Architekturen und Systeme aufgezeigt.

Anschließend wurde das Kontextmodell in die drei Kontextebenen, Signalebene, Merkmalebene und Objektebene unterteilt und erläutert, welche Funktion die einzelnen Ebenen erfüllen und wir die darin enthaltenen Kontextinformationen charakterisiert werden können. Die Signalebene hält sehr flüchtige semantikarme Daten bereit. Die Merkmalebene beinhaltet dagegen bereits fusionierte Daten, die einen breiten Blickwinkel auf den aktuellen Kontext zulassen. Die Objektebene verfügt über semantisch höherwertige Informationen zu konkreten Objekten in der Arbeitsumgebung, für die sie allerdings historische Daten und Daten aus zeitlich versetzten Quellen fusionieren muss.

In Folge dessen wurden die beiden Anwendungsszenarien Fahrerlose Transportsysteme und Sichere Offshore-Operationen für dynamische Kontextmodelle näher beleuchtet und die, in diesen Anwendungen notwendigen, Kontextinformationen aufgelistet. Des Weiteren wurden einige Beispieloperationen aus den Anwendungsszenarien herausgegriffen und mit den hierfür notwendigen Kontextebenen verknüpft.

Aufbauend darauf, wurden exemplarisch am Beispiel einer probabilistischen Belegungskarte im Kontext von fahrerlosen Transportsystemen und am Beispiel der Distanzbestimmung zwischen Arbeitern und schwebenden Lasten gezeigt, wie ein solches dynamisches Kontextmodell innerhalb eines Datenstrommanagementsystems mit Hilfe der dortigen temporal relationalen Operatoren und speziellen Abbildungsfunktionen als Anfrage hinterlegt und berechnet werden kann. Zu diesem Zweck wurde zunächst die logische Integration aufgezeigt, umso eine höchst mögliche Flexibilität und Kombinierbarkeit der Operatoren zu erreichen. Des Weiteren wurden mögliche physische Implementierung der logischen Operatoren vorgestellt. Durch diese Flexibilität lassen sich nun auch beide Verarbeitungen kombinieren.

Am Beispiel der Belegungskarte und der Distanzbestimmung zeigte sich bereits, dass die Sensorwahrnehmungen als nicht exakte Werte betrachtet werden dürfen. Daher wurde bei der Anfragebeschreibung des Ausbreitungsoperators etwa verlangt, dass das Messrauschen von einem Anwender zuvor in dem Ausbreitungsoperator hinterlegt wird. Hier wurde zudem davon ausgegangen, dass die Varianz der Distanzsensoren zuvor bekannt ist und sich zur Laufzeit auch konstant verhält. Dies ist bei realen Anwendungen allerdings nicht immer der Fall. Auch am Beispiel der Distanzbestimmung wurden Ansätze vorgestellt, wie mit ungenauen Daten verfahren werden kann. Allerdings wurden auch bei dem hier vorgestellten Ansatz keine anderen Umwelteinflüsse oder sonstige, die Messwerte negativ beeinflussenden, Größen betrachtet.

4 Qualitätsbestimmung in Datenströmen

Im vorangegangenen Kapitel wurde aufgezeigt, wie mit Hilfe eines Datenstrommanagementsystems ein dynamisches Kontextmodell erstellt werden kann. Hierbei zeigte sich auch schon, dass die von Sensoren wahrgenommenen Eigenschaften durch das verwendete Messverfahren und die zum Zeitpunkt der Messung herrschenden Umweltbedingungen in ihrer Wahrnehmung gestört werden können. Hierdurch kann die Qualität des erstellten Kontextmodells und die darin enthaltenen Informationen beeinträchtigt werden.

4.1 Einführung

Bei der bisherigen Erstellung und Verwaltung von Kontextmodellen wurde die Qualität nur durch die direkte Angabe einer Qualitätsdimension, der Genauigkeit, bei der Berechnung der Kontextinformationen des dynamischen Kontextmodells berücksichtigt, welche dann im Laufe der Verarbeitung als konstant angesehen wurde. Allgemein lässt sich aber sagen, dass die Qualität von Kontextmodellen und die hierbei verwendeten Sensoren durch eine Vielzahl von Einflüssen beeinträchtigt sein können und dass die Qualität der Wahrnehmung von Sensoren und der daraus gewonnenen Kontextinformationen aus einer Vielzahl von Gründen benötigt wird. Zu den wichtigsten Gründen für die Nutzung von Qualitäten zählen

- die Auswahl von geeigneten Sensoren,
- die Unterstützung von Qualitätsschwellwerten für kontextsensitive Anwendungen,
- die Verbesserung der Argumentationen und Entscheidungen und
- die Verringerung von inkorrekten Adaptionsprozessen.

Aber was bedeutet Qualität und was ist die Qualität eines Kontextes? Eine häufig zitierte Definition der Qualität eines Kontextes (QoC) stammt von Buchholz et al. [BKS03], welche die Qualität eines Kontextes definieren als:

> *„Any information that describes the quality of information that is used as context information. Thus, QoC refers to information and not to the process nor the hardware component that possibly provide the information.“*

Eine Kontextqualität ist also jedwede Information, die die Qualität einer Information beschreibt, die als Kontextinformation verwendet wird. Somit bezieht sich die Kontextqualität auf Informationen und nicht auf den Prozess oder die Hardwarekomponente, die diese Information möglicherweise bereitstellt. In Abbildung 4.1 sehen wir noch einmal den Prozess zur Erstellung der verschiedenen Ebenen eines Kontextmodells. Ziel dieses Kapitels

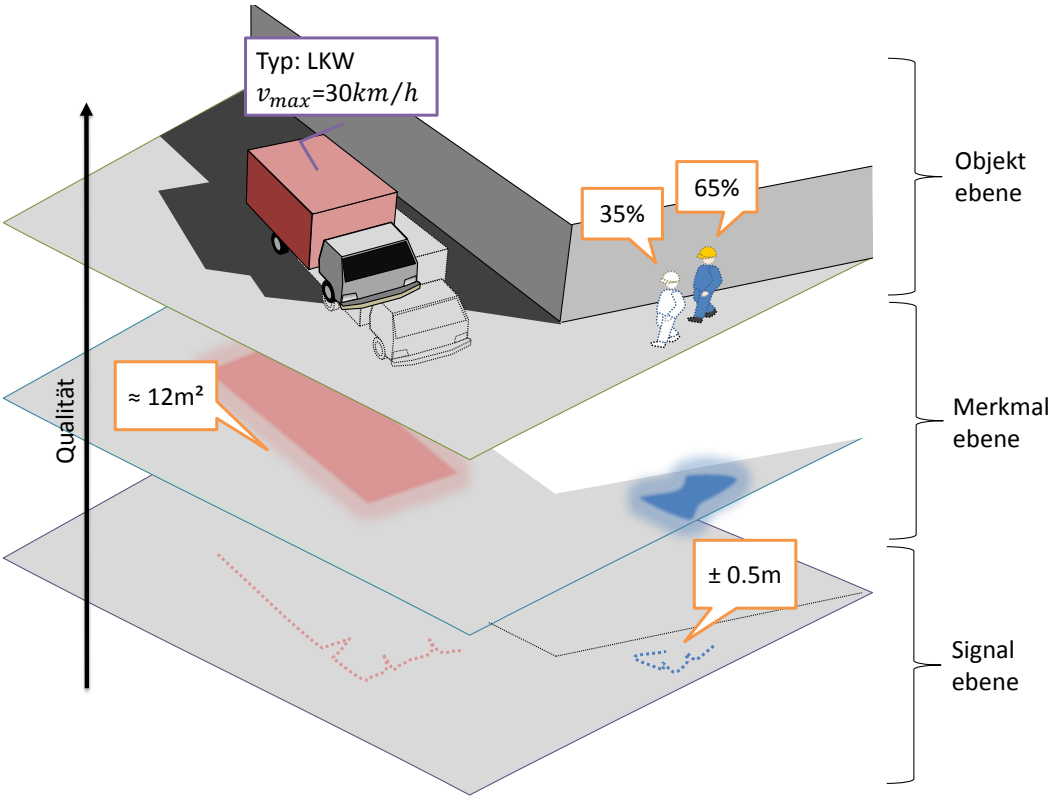


Abbildung 4.1: *Qualitätsbetrachtung in allen Ebenen der Kontextmodellerstellung*

wird es nun sein, einen Ansatz zu entwickelt, wie Qualitäten innerhalb der kompletten Verarbeitung und Erstellung dieser Ebenen bestimmt und bei der Verarbeitung durch alle Ebenen hindurch berücksichtigt werden können. Dazu wird zunächst in Abschnitt 4.2 genauer der Frage nachgegangen, welche verschiedenen Dimensionen von Qualitäten existieren und erörtert, welche dieser Dimension für die Erstellung und Verwaltung eines dynamischen Kontextmodells relevant sind. Um die Qualität von Elementen aus unterschiedlichen Quellen vergleichbar zu machen, werden in Abschnitt 4.3 zusätzliche Qualitätsindikatoren eingeführt, welche während der Anfrage berücksichtigt werden. In Abschnitt 4.4 und Abschnitt 4.5 wird dann aufgezeigt, wie diese Qualitätsdimensionen durch die Modellierung von Beziehungen zwischen Sensoren und die darauf aufbauende Verarbeitung ihrer Messungen bestimmt werden können. Anschließend wird verdeutlicht, wie dieser Ansatz in ein Datenstrommanagementsystem (DSMS) auf Basis des dort verwendeten Modells integriert werden kann. Hierzu gilt es zunächst diese Qualitätsinformationen zu bestimmen, mit den Daten zu verknüpfen und während der Verarbeitung zu beachten, um so eine qualitätssensitive Verarbeitung von Sensordatenströmen für dynamische Kontextmodelle zu ermöglichen. Abschließend findet in Abschnitt 4.7 eine Diskussion über die hier verwendeten Techniken statt.

4.2 Qualitätsdimensionen

In der Literatur finden sich verschiedene Qualitätsdimensionen für die Kontextqualität und die Datenqualität, welche im Allgemeinen verwendet werden um das Verhalten von Anwendungen an die aktuell herrschenden Umweltbedingungen anzupassen. Im Folgenden liegt daher der Fokus darauf, die am weitesten verbreiteten Dimensionen zur Beschreibung der Qualität von Daten und Kontextmodellen und ihrer jeweiligen Definitionen von verschiedenen Autoren in der Literatur aufzuzeigen.

In einer Übersicht über kontextsensitive Systeme [BDR07] nennen die Autoren neben der Kategorie der Kontextinformation und dem eigentlich Messwert eines Sensors noch den Zeitstempel, die Quelle und das Vertrauen in den Wert, als Maß für die Unsicherheit, als zusätzliches nützliches Attribut für kontextsensitive Systeme.

Für die Adaption von kontextqualitätsbasierenden Anwendungen verwenden Sheikh et al. [SWS07] die Qualitätsdimensionen Präzision, Frische, zeitliche und räumliche Auflösung und die Korrektheitswahrscheinlichkeit. Hierbei konzentrieren sich die Autoren auf menschliche Benutzer im Gegensatz zu Objekten im Allgemeinen. Dazu stützen die Autoren ihre Arbeit auf die Tele-Überwachung von Epilepsiepatienten und nutzen Kontextinformation zur Benachrichtigung von medizinischem Personal je nach Kontext des Patienten. Die Präzision beschreibt hierbei die Granularität mit welcher Kontextinformation eine Situation in der wahren Welt beschreiben. Unter der Frische verstehen die Autoren die Zeit, die verstreicht zwischen der Wahrnehmung einer Kontextinformation und ihrer Auslieferung an einen Anfragenden. Die zeitliche Auflösung ist der Zeitraum in dem eine einzelne Kontextinformation nutzbar ist und die Korrektheitswahrscheinlichkeit ist die Wahrscheinlichkeit, dass eine Instanz eines Kontextes genau die entsprechende Realweltsituation wiedergibt zu dem Zeitpunkt zu der diese erkannt wurde.

Zur Beschreibung von Kontextqualitäten in kontextsensitiven Anwendungen fokussieren sich Buchholz et al. [BKS03], am Beispiel eines „Restaurantfinders“ und eines „Dating-Portals“, auf die Verwendung der Qualitätsdimensionen Präzision, Korrektheitswahrscheinlichkeit, Vertrauenswürdigkeit, Auflösung und Aktualität. Unter der Präzision verstehen die Autoren wie exakt die Kontextinformationen die Wirklichkeit wiedergeben. Dabei wird die Präzision in Form von Grenzen angegeben. Die Dimension Korrektheitswahrscheinlichkeit gibt die Wahrscheinlichkeit an, mit der eine Kontextinformation korrekt ist. Die Vertrauenswürdigkeit gibt ebenfalls an, wie wahrscheinlich es ist, dass eine Kontextinformation korrekt ist. Hierbei unterscheiden die Autoren allerdings von der Korrektheitswahrscheinlichkeit dahingehend, dass die Vertrauenswürdigkeit sich auf die Quelle der Kontextinformation bezieht. Die Dimension der Auflösung bezieht sich auf die Granularität der Information und die Aktualität beschreibt das Alter einer Kontextinformation. Hierbei ist zu beachten, dass die verwendeten Kontextinformationen aus unterschiedlichen Quellen stammen, wie etwa einer Datenbank, einem Kartenanbieter oder einem GPS-fähigen Mobiltelefone. Auf Grund der betrachteten Anwendung sind die verwendeten Kontextinformationen weder zeitkritisch noch sicherheitskritisch und die Datenraten

der betrachteten Kontextinformationen verglichen mit den Datenraten der verwendeten Sensoren in den betrachteten Anwendungsfällen in dieser Arbeit als gering anzusehen.

Mit einem Fokus auf die Qualität von Sensordaten bei der Datenstromverarbeitung verweisen Klein et al. [KL09a] auf die Qualitätsdimensionen Genauigkeit, Sicherheitswahrscheinlichkeit, Vollständigkeit, Datenmenge und Aktualität der Daten. Die Genauigkeit beschreibt dabei den systematischen Messfehler aufgrund von statischen Fehlern im Messprozess. Die Sicherheitswahrscheinlichkeit stellt dagegen den statistischen Messfehler aufgrund von zufälligen Umwelteinflüssen dar. Unter der Vollständigkeit verstehen die Autoren das Fehlen von Werten aufgrund von Sensorausfällen und Fehlfunktionen. Die Datenmenge stellt die Menge an Rohdaten dar, die für die Ermittlung eines Resultats verwendet wurden. Für die Aktualität geben die Autoren zwei Möglichkeiten der Interpretation an: Zum einen das Alter eines Datums als Differenz zwischen Messzeitpunkt und Systemzeit und zum anderen die Pünktlichkeit eines Datums in Bezug auf den Anwendungskontext. In [KL09b] klassifizieren die Autoren zudem die Datenqualitäten in immanente Datenqualitäten, welche ein einzelnes Datum charakterisieren, wie etwa die Genauigkeit oder die Glaubwürdigkeit, sowie kontextuelle Datenqualitäten, wie etwa die Vollständigkeit, die fehlende Werte in einer Datenmenge charakterisiert.

Filho et al. [FMS+10] unterscheiden für die Auswertung von Qualitäten von hergeleiteten und gefolgerten Kontextinformationen in einem Kontextverwaltungssystem zwischen Kontextqualitätsindikatoren und Kontextqualitätsparametern. Kontextqualitätsindikatoren sind jedwede wohldefinierte Qualitätsaspekte, die ausgewertet und genutzt werden können um die Qualität einer Kontextinformation zu beschreiben. Kontextparameter dagegen sind jedwede wahrgenommenen Informationen über die Umgebung, die genutzt werden können um Kontextqualitätsindikatoren zu messen. Ein Kontextqualitätsindikator wird dabei als ein Wert in dem Bereich $[0, 1]$ dargestellt. In ihrer Arbeit verweisen die Autoren auf die Kontextqualitätsindikatoren Sensitivität, Zugriffssicherheit, Vollständigkeit, Präzision und Auflösung und definieren diese Indikatoren wie folgt: Sensitivität ist die Offenlegung von Kontextinformationen zu einem bestimmten Zeitpunkt, Zugriffssicherheit ist die Wahrscheinlichkeit, mit der die Kontextinformationen vertraulich zu einem Nutzer übermittelt werden, Vollständigkeit ist der Grad der Anwendbarkeit, mit dem die Kontextinformationen einem Nutzer zur Verfügung gestellt wird, Präzision ist der Detailgrad, mit dem die Kontextinformationen die wahre Welt charakterisieren und Auflösung ist die räumliche Granularität, mit der die Kontextinformationen von der Umwelt gemessen wurden.

In Batini et al. [BS06] geben die Autoren eine breitere Übersicht über die verschiedenen Qualitätsdimensionen und ihre Definitionen. Hierbei werden nicht nur die Datenqualität berücksichtigt, sondern auch Qualitätsdimensionen untersucht, die das Verwalten der Daten beschreiben. Die Liste der Qualitätsdimensionen beinhaltet dabei unter anderem die Qualitätsdimension Genauigkeit, Vollständigkeit, zeitliche Dimensionen und Konsistenz. Die Genauigkeit ist dabei definiert als die Nähe zwischen zwei Werten unter der Annahme, dass einer die korrekte Repräsentation eines Phänomens in der wahren Welt ist, welches

der andere Wert vorgibt zu sein. Die Autoren identifizieren zwei Arten von Genauigkeit, nämlich die syntaktische Genauigkeit und die semantische Genauigkeit:

- Syntaktische Genauigkeit ist die Nähe eines Wertes zu den Elementen in der dazugehörigen Definitionsdomäne.
- Semantische Genauigkeit ist die Nähe eines Wertes zu dem wahren Wert.

Mit dem Fokus auf Datenbanken unterscheiden die Autoren zwischen vier Typen von Vollständigkeit, nämlich die Wertvollständigkeit, die Tupelvollständigkeit, die Attributvollständigkeit und die Relationsvollständigkeit als:

- Wertvollständigkeit bildet die Präsenz von `NULL`-Werten für ein Feld in einem Tupel ab.
- Tupelvollständigkeit charakterisiert die Vollständigkeit eines Tupels mit Bezug auf die Werte in allen Feldern.
- Attributvollständigkeit misst die Anzahl von `NULL`-Werten für ein bestimmtes Attribut in einer Relation.
- Relationsvollständigkeit nimmt die Präsenz von `NULL`-Werten in einer ganzen Relation auf.

Weiterhin wird bei der zeitlichen Dimension unterschieden zwischen Zeitnähe, Flüchtigkeit und Aktualität und wie folgt definiert:

- Zeitnähe gibt an, wie häufig Daten aktualisiert werden.
- Flüchtigkeit ist eine Metrik, die angibt, wie lange ein bestimmtes Datum noch als gültig betrachtet werden kann.
- Aktualität drückt aus, wie aktuell die Daten für eine bestimmte Aufgabe sind.

Die letzte Qualitätsdimension, die Konsistenz, bildet die Verletzung von semantischen Regeln über definierte Datenelemente ab. Zudem klassifizieren die Autoren die Qualitätsdimension in die Qualitätsdimensionsklassen Immanente Qualitätsdimensionen, kontextuelle Qualitätsdimensionen, repräsentationelle Qualitätsdimensionen und Qualitätsdimensionen der Zugänglichkeit. Immanente Datenqualitäten erfassen die selbstbezogenen Qualitäten, wie etwa die Genauigkeit, die untrennbar mit den Daten verknüpft sind. Kontextuelle Datenqualität betrachtet den Kontext, in dem Daten verwendet werden, zum Beispiel ist die Vollständigkeit strikt mit der Aufgabe verknüpft. Repräsentationelle Datenqualität erfasst Qualitätsaspekte der Darstellung der Daten, z. B. die Interpretierbarkeit. Die Datenqualitätsklasse der Zugänglichkeit betrachtet den Zugriff auf die Daten und den Grad der Sicherheit.

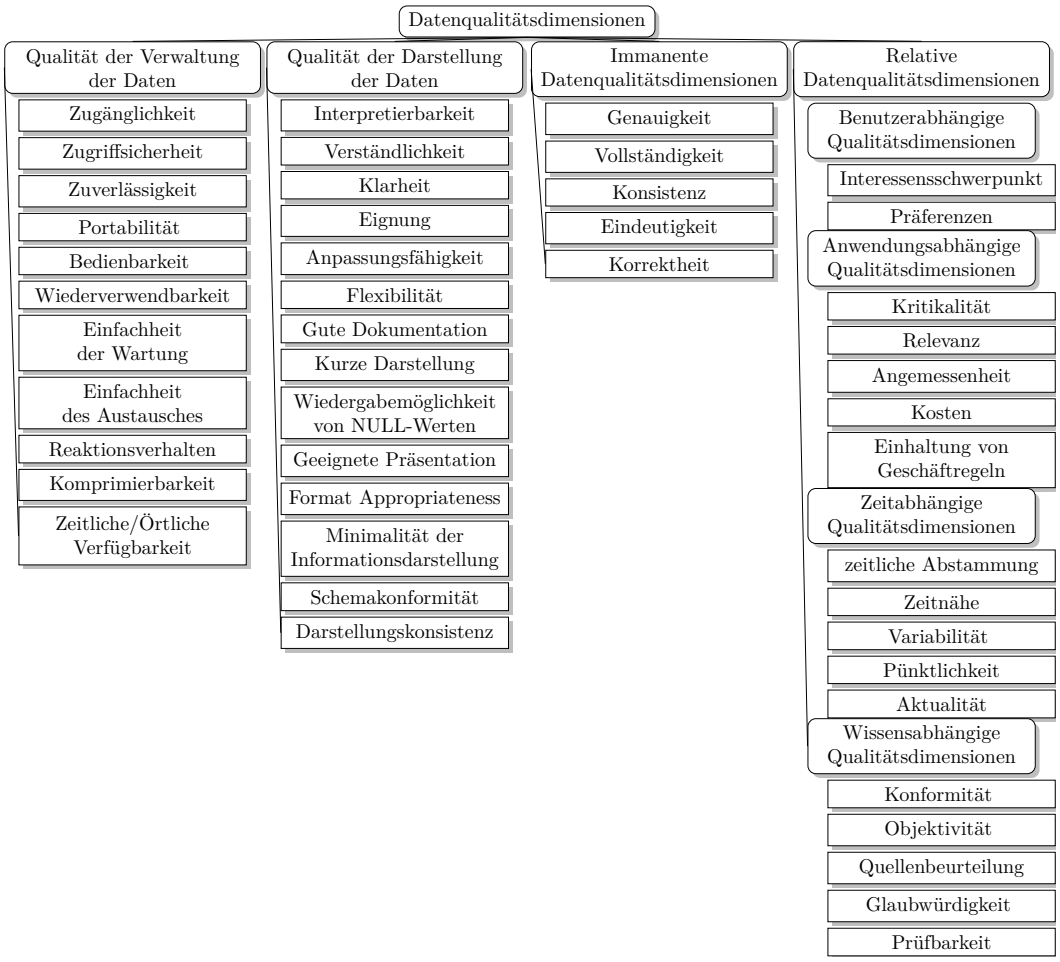


Abbildung 4.2: *Klassifikation der Qualitätsdimensionen in Qualität der Verwaltung, Darstellungsqualität, Eigenqualität und relative Datenqualität nach [BE06]*

Mit einem Fokus auf die Verbesserung des Qualitätsbewusstseins von Data-Mining-Prozessen zeigt Berti-Equille in [BE06] weitere gängige Definitionen der Qualitätsdimensionen Genauigkeit, Vollständigkeit, Zeitnähe und Konsistenz in der Literatur auf. Hierbei wird die Genauigkeit als das Ausmaß, in welchem gesammelte Daten frei von Fehlern sind, angesehen oder aber als der Quotient der korrekten Daten in einer Quelle im Verhältnis zur Anzahl aller Daten in einer Quelle gemessen. Die Autorin zeigt dabei auch auf, dass die Vollständigkeit in der Literatur als der prozentuale Anteil von Information der wahren Welt in einer Quelle angesehen wird oder auch als der Quotient der Anzahl von `NULL`-Werten in einer Quelle und der Größe der Relation gemessen werden kann. Die letzte Definition gleicht sich hierbei mit der Definition von [BS06] über die Relationsvollständigkeit. Die Zeitnähe wiederum gibt an, in wie weit Daten aktuell genug sind für eine bestimmte Aufgabe. Die Qualitätsdimension Konsistenz wird definiert als die Kohärenz von gleichen

Daten in mehreren Kopien oder andere Daten mit Bezug auf vordefinierte Integritätsbedingungen und Regeln. Des Weiteren klassifiziert die Autorin die Datenqualitätsdimensionen in vier Klassen: Qualität der Verwaltung von Daten, Qualität der Darstellung von Daten, Eigenqualitäten und relative Datenqualitäten (vgl. Abb. 4.2). Die Klasse der Immanenten Datenqualitäten und der Darstellung der Daten entsprechen dabei direkt den von [BS06] genannten Klassen. Die Klasse der Verwaltung der Daten entspricht der Qualitätsklasse der Zugänglichkeit und die Relativen Datenqualitäten den von den Autoren genannte Kontextuelle Datenqualitätsklasse.

In Geisler et al. [GQWJ11] untersuchen die Autoren mit einem Fokus auf die Datenstromverarbeitung die Qualitätsdimensionen Vollständigkeit, Datenvolumen, Zeitnähe, Konsistenz und Vertrauen. Die Vollständigkeit ist dabei das Verhältnis zwischen der Anzahl von fehlenden Werten oder Tupel im Verhältnis zu der Anzahl von empfangenen Werten oder Tupeln. Das Datenvolumen ist die Anzahl der Tupel oder Werte auf denen ein Ergebnis basiert, z. B. die Anzahl von Tupeln, die verwendet werden um eine Aggregation wie etwa einen Median zu berechnen. Somit kann das Datenvolumen als ein Teil der Vollständigkeitsdimension interpretiert werden. Die Zeitnähe wird von den Autoren als das Alter eines Wertes oder Tupels angesehen. Für die Genauigkeit geben die Autoren das Beispiel eines konstanten Messfehlers oder das Ergebnis eines Data-Mining-Algorithmus an. Die Konsistenz wiederum gibt den Grad an, zu dem ein Wert eines Attributs bestimmte Beschränkungen befolgt, etwa in einem bestimmten Bereich liegt. Unter der statistischen Sicherheit verstehen die Autoren die Glaubwürdigkeit eines Wertes oder Tupels.

4.2.1 Diskussion

In Tabelle 4.1 finden sich alle betrachteten Qualitätsdimensionen mit den jeweiligen Ansätzen und ihrem Fokus wieder. Bei den zeitabhängigen Qualitätsdimensionen wird dabei unterschieden, ob die vorgestellten Ansätze unabhängig von einer konkreten Anwendung sind oder immer im Bezug zu einer konkreten Anwendung stehen und so immer auch ein Wissen über die Anwendung benötigen. Bei den immanenten Datenqualitätsdimensionen wird ebenfalls unterschieden, ob die Dimensionen nur durch ein a-priori Wissen über die Anwendung gemessen werden können oder rein auf den Daten ohne ein konkretes Wissen über die Anwendung ermittelt werden können. Diese Unterscheidung ist deswegen relevant, weil eines der Ziele der hier entwickelten Ansätze eine weitestgehend automatisierte Weise der Qualitätsannotation von Daten ist, welche daher über kein konkretes Anwendungswissen verfügt.

In der Tabelle 4.1 unterscheiden sich die Definition der Zeitnähe von Berti-Equille [BE06] von den anderen Ansätzen dahingehend, dass sie die Zeitnähe als die Aktualität von Daten für eine bestimmte Aufgabe definiert, während bei Klein et al. [KL09a] die Zeitnähe unterteilt wird in Alter und Pünktlichkeit. Bei Batini et al. [BS06] stellt die Zeitnähe, die Aktualität und die Flüchtigkeit drei getrennte Dimensionen dar. Die Konsistenz ist dage-

gen in allen Ansätzen als eine klar anwendungsabhängige Qualitätsdimension genannt worden, bei der die jeweiligen Daten mit zuvor anwendungsspezifischen Regeln verglichen werden.

Ansatz	Fokus	Zeitabhängige Qualitätsdimensionen		Immanente Qualitätsdimensionen	
		Abhängig	Unabhängig	Abhängig	Unabhängig
Sheikh et al. [SWS07]	Menschliche Benutzer	Auflösung	Frische	Präzision; Auflösung; Korrektheit	
Buchholz et al. [BKS03]	Datenbank		Aktualität	Präzision; Korrektheit; Vertrauen; Auflösung	
Klein et al. [KL09a]	Datenstrom	Pünktlichkeit	Alter		Genauigkeit; Sicherheit; Vollständigkeit; Datenmenge
Filho et al. [FMS+10]	Datenstrom			Sensitivität; Sicherheit; Vollständigkeit; Präzision; Auflösung	
Batini et al. [BS06]	Datenbank	Flüchtigkeit; Aktualität	Zeitnähe	Konsistenz	Genauigkeit; Vollständigkeit
Berti-Equille [BE06]	Data-Mining	Zeitnähe		Konsistenz	Genauigkeit; Vollständigkeit
Geisler et al. [GQWJ11]	Datenstrom		Zeitnähe	Konsistenz	Vollständigkeit; Datenmenge; Vertrauen

Tabelle 4.1: *Gegenüberstellung der Qualitätsdimensionen*

4.3 Qualitätsindikatoren

In den bisher betrachteten Arbeiten waren Zeitnähe, Vollständigkeit, Konsistenz und Genauigkeit die vier häufigsten Qualitätsdimensionen zur Beschreibung von Kontextqualitäten. Im Folgenden werden daher diese Qualitätsdimensionen bei der Erstellung eines dynamischen Kontextmodells fokussiert. Zunächst aber gilt es, aus den bisherigen Definition der vier Qualitätsdimensionen die jeweils passende Definition für die Verarbeitung in einem DSMS anzupassen.

4.3.1 Zeitindikatoren

Für die Definition der zeitlichen Qualitäten wird auf die Definition von [BS06] zurückgegriffen. Hierbei wird die Aktualität eines Datums durch das Alter des Datums und die Latenz der Verarbeitung des Datums definiert. Sei hierzu $t_S, t_O, t_A \in T$ einzelne Zeitstempel aus der Menge aller Zeitstempel T mit t_S dem Zeitpunkt der Messung, t_A der Systemzeit beim Eintreffen des Datums und t_O der Systemzeit zum Zeitpunkt der Ausgabe, dann gilt:

$$t_{Currency} = (t_A - t_S) + (t_O - t_A) \tag{4.1}$$

Der erste Term bestimmt das Alter der Messung beim Eintreffen in das System. Der zweite Term stellt die Latenz der Verarbeitung dar.

Die Flüchtigkeit ist nach [BS06] definiert als die Zeitspanne, die ein Datum gültig ist. Diese Definition ist identisch mit der Definition der Länge eines Zeitintervalls $|[t_S, t_E)|$ mit Startzeitstempel t_S und Endzeitstempel t_E für ein physisches Stromelement in dem hier betrachteten DSMS. Daher werden für diesen Qualitätsindikator keine weiteren Informationen benötigt, die nicht schon in einem Stromelement vorhanden sind. Sei $t_S, t_E \in T$ dann gilt:

$$t_{Volatility} = |[t_S, t_E)| \tag{4.2}$$

Die Zeitnähe wiederum ist von den Autoren definiert als das Verhältnis zwischen der Aktualität eines Datums und dessen Flüchtigkeit. Sei $t_S, t_E, t_O, t_A \in T$ dann gilt:

$$q_{Timeliness} = \max 0, 1 - \frac{(t_A - t_S) + (t_O - t_A)}{|[t_S, t_E)|} \tag{4.3}$$

Auf diese Weise wird der Wert der Zeitnähe auf das Intervall $[0, 1]$ normiert. Offen ist allerdings noch die Frage, woher der Endzeitpunkt für ein Datum bzw. eine Sensormessung stammt. Zum einen könnte dieser explizit im Rahmen der Kontexterstellung gesetzt werden, zum anderen kann allerdings auch die Frequenz des Sensors verwendet werden um den Endzeitpunkt zu setzen. Die Verwendung der Frequenz berücksichtigt zudem die Tatsache, dass Messwerte von hochfrequenten Sensoren generell schneller ihre Gültigkeit und

somit ihren Nutzen für eine korrekte Berechnung verlieren als Sensoren mit einer niedrigeren Frequenz, da sie zumeist in Szenarien eingesetzt werden, in denen ihre Messungen auch zu einem späteren Zeitpunkt von Nutzen sind. Somit erhalten wir unter Verwendung der Frequenz f folgende Definition für die Zeitnähe eines Datums. Sei $t_S, t_E, t_O, t_A \in T$ und $f \in \mathbb{R}_{>0}$ dann gilt:

$$q_{Timeliness} = \max 0, 1 - \frac{(t_A - t_S) + (t_O - t_A)}{|[t_S, t_S + \frac{1}{f})|} \tag{4.4}$$

Bei der Bestimmung der Zeitnähe muss allerdings darauf geachtet werden, dass sowohl der interne Zeitgeber des Sensors, wie auch die Zeit des Systems synchron laufen, da hier die Zeiten von zwei getrennten Systemen verwendet werden, während bei der bisherigen Betrachtung der Berechnungen in den temporal relationalen Operatoren in einem DSMS nur die Zeitstempel der Stromelemente, aber nie die Systemzeit eine Rolle für die deterministische Verarbeitung spielte.

4.3.2 Vollständigkeit

Für die Definition der Vollständigkeit und ihrer Unterteilung in die vier Klassen Wertvollständigkeit, Tupelvollständigkeit, Attributvollständigkeit und Relationsvollständigkeit wird ebenfalls auf [BS06] verwiesen. Im Kontext der Datenstromverarbeitung sind die Definitionen der Attributvollständigkeit und der Relationsvollständigkeit allerdings nicht direkt anwendbar. Grund hierfür ist die Tatsache, dass sich diese auf den kompletten Datenstrom beziehen, welcher unter Umständen eine unendliche Sequenz von Messwerten darstellt und somit die Anzahl von `NULL`-Werten für ein Attribut oder für den kompletten Strom nicht messbar ist. Hingegen lassen sich die Definitionen der Wertvollständigkeit und der Tupelvollständigkeit direkt auf die Datenstromverarbeitung übertragen. Durch die Anwendung von Fensteroperatoren lassen sich allerdings auch die Attributvollständigkeit und die Relationsvollständigkeit auf den Inhalt eines Fensters in einem logischen Datenstrom anwenden. Sei $S \in \mathbb{S}^l_{\mathcal{T}}$ hierzu ein logischer Datenstrom, dann gilt mit Bezug auf die Definition 1 eines logischen Datenstroms aus Abschnitt 2.3 für die Relationsvollständigkeit

$$q_{R-Completeness} = \frac{|\{(e, n, t) \in S | e \neq NULL\}|}{|S|} \tag{4.5}$$

bzw. für die Attributvollständigkeit

$$q_{A-Completeness} = \frac{|\{a \in e | a \neq NULL \wedge (e, n, t) \in S|\}|}{|S|} \tag{4.6}$$

Hierbei ist a eine Attributausprägung aus dem Nutzdatentupel $e \in \Omega_{\mathcal{T}}$ mit Schema $\mathcal{T}$. Die Größe eines solchen Fensters, welches die Größe des Betrachtungsrahmen $|S|$ auf

den Datenstrom S definiert, kann, wie schon bei der Ermittlung der Zeitnähe, entweder explizit im Rahmen der Kontexterstellung gesetzt werden, oder auch durch die Frequenz des Sensors ermittelt werden.

4.3.3 Konsistenz

Die Konsistenz wird sowohl von [GQWJ11], wie auch von [BS06] und [BE06] als die Verletzung von semantischen Regeln bzw. Bedingungen angesehen. Eine semantische Regel $p_{Consistency} \in \mathbb{P}_{\mathcal{T}}$ kann im Kontext der Datenstromverarbeitung als eine Funktion $p_{Consistency} : \Omega_{\mathcal{T}} \rightarrow \{true, false\}$ aufgefasst werden, die die semantische Regel auf die Elemente eines Datenstroms mit Schema $\mathcal{T}$ anwendet. Diese Definition entspricht einer Selektion auf dem Datenstrom anhand des Prädikats $p_{Consistency}$. Somit stellt der Konsistenzindikator das Verhältnis zwischen der Anzahl der Elemente, die der semantischen Regel genügen, und der Anzahl aller Elemente eines Stroms dar. Sei hierzu $p_{Consistency}$ eine semantische Regel und $S \in \mathbb{S}^{l}_{\mathcal{T}}$ ein logischer Datenstrom, dann gilt für die Konsistenz auf einem logischen Datenstrom:

$$q_{Consistency} = \frac{|\{(e, n, t) \in S | p_{Consistency}(e)\}|}{|S|} \tag{4.7}$$

Ähnlich der Vollständigkeitsdimension ist hier die Größe des betrachteten Bereichs $|S|$ des Stroms durch einen Fensteroperator zu setzen.

4.3.4 Genauigkeit

Für die Definition der Genauigkeit fassen wir die von [KL09a] genannte Genauigkeit und Sicherheitswahrscheinlichkeit zusammen, da diese ebenfalls der von [BS06] genannten semantischen Genauigkeit und dem allgemeinen systematischen und statistischen Fehler von Sensordaten entspricht. Da allerdings die Genauigkeit direkt eine Eigenschaft des Messwertes darstellt, ist hier keine Normierung, wie bei den anderen Qualitäten, möglich. Es stellt sich vielmehr die Frage, wie dieser Wert ermittelt und bei der Erstellung und Verarbeitung eines dynamischen Kontextmodells repräsentiert und ausgewertet werden kann. Eine Möglichkeit ist die Repräsentation von Ungenauigkeiten in Form einer Mischverteilung wie sie in [TPD+12] angewendet wird.

4.4 Indirekte Bestimmung von Qualitätsinformationen

Die Qualitäten von Sensorwahrnehmungen hängen von vielen verschiedenen Eigenschaften in der Arbeitsumgebung eines Sensors ab, können aber meist vom eigentlichen Sensor nicht direkt wahrgenommen werden. So hat beispielsweise die Temperatur im Arbeitsbereich eines Sensors immer einen direkten Einfluss auf die Funktionsfähigkeit des

Sensors, wie dies etwa in [BRB+14] am Beispiel von Funknetzwerken demonstriert wurde. Allerdings verfügt nicht jeder Sensor über die Möglichkeit die aktuelle Temperatur zu erfassen. Zur Bestimmung der vorherrschenden Qualitätsinformationen von Sensorwahrnehmungen müssen daher zusätzliche Sensoren eingesetzt werden, die die aktuellen Umweltbedingungen, unter denen andere Sensor arbeiten, überwachen. Auf diese Weise können indirekt, durch die zusätzlichen Sensoren, die Qualitätsinformationen von Sensorwahrnehmungen kontinuierlich durch deren Messungen bestimmt werden. Hierzu ist es allerdings notwendig, die Beziehungen zwischen Sensoren auszudrücken und den Einfluss einer Messgröße auf eine Sensorwahrnehmung bereit zu stellen. Zu diesem Zweck gibt es prinzipiell zwei Möglichkeiten um die Beziehung anzugeben. Zum einen kann die Beziehung explizit durch den Anwender innerhalb der Verarbeitung der Sensorwahrnehmungen formuliert werden, also eine anwendergestützte Bestimmung der Qualitätsinformationen, bei der die Werte der Qualitätsdimensionen allein vom Wissen des Anwenders über die verwendete Sensorik, die Anwendung und den Verarbeitungsprozess abhängen. Zum anderen kann die Beziehung a-priori in einem System in Form einer Wissensbasis modelliert werden, also eine systemgestützte Bestimmung von Qualitätsinformationen, bei der die Werte der Qualitätsdimension auf Basis der zuvor in einer Wissensbasis allgemein modellierten Beziehungen zwischen Sensoren und Messgrößen bestimmt werden.

4.4.1 Anwendergestützte Bestimmung von Qualitätsinformationen

Bei der anwendergestützten Bestimmung von Qualitätsinformationen werden die Beziehung explizit innerhalb einer Verarbeitungsanweisung formuliert. Ein solches Vorgehen ist typisch bei Anwendungen, die für eine festgelegte Menge expliziter Sensoren entwickelt werden. Wobei die Daten für die Qualität, wie etwa die Streuung, und die Abhängigkeiten zu bestimmten Umwelteinflüssen, wie Feuchtigkeit und Temperatur, aus dem Datenblatt des jeweiligen Sensorherstellers entnommen werden.

Auch in [KN12] wurde die Bestimmung der Unsicherheit von Ergebnissen einer Sensorfusion manuell vorgenommen, indem explizit die Verbindung zwischen primären Sensoren und sekundären Sensor formuliert wurde um Aussagen über die Funktionsfähigkeit bzw. die Nicht-Funktionsfähigkeit des primären Sensors zu tätigen. Ein sekundärer Sensor ist dabei ein Sensor, der die Umwelteinflüsse auf einen anderen Sensor, den primären Sensor, misst und somit eine Aussage über den Arbeitsbereich in der wahrgenommen Dimension des Sensors tätigen kann. Die Aussage über den Arbeitsbereich wurde in der genannten Arbeit nach dem Dempster-Shafer Theorem [Sha76] ausgewertet. In dieser Arbeit wurde zusätzlich zur Qualität des primären Sensors auch die Varianz des sekundären Sensors in die Auswertung mit einbezogen. Die Varianz des sekundären Sensors war hierbei allerdings explizit durch den Anwender vorgegeben. Durch die Modellierung der Varianz entstand so ein Wahrscheinlichkeitsintervall über die Aussage der Funktionsfähigkeit des primären Sensors. Weiterhin wurde auch beschrieben, wie die Aussagen mehrerer sekundärer Sensoren durch die Anwendung der Kombinationsregel von L. Zhang [Zha94] zu

einer Aussage über die Funktionsfähigkeit des primären Sensors kombiniert werden können. Hierbei zeigte sich allerdings auch, dass die Interpretation der Kombination aus Wahrscheinlichkeitsintervallen, wie sie bei diesem Ansatz entstanden sind, nicht eindeutig ist.

Bei der anwendergestützten Bestimmung von Qualitätsinformationen besteht der Vorteil explizites Domänenwissen über die Sensoren und die Anwendung schnell in der Verarbeitung zu formulieren und entsprechend leicht zu ändern. Dies trifft allerdings nur für Anwendungen zu, bei denen vor allem gleiche Sensoren oder eine geringe Anzahl unterschiedlicher Sensoren im Einsatz sind. Bei größeren Anwendungen mit verschiedenen Sensoren ist dagegen die explizite Modellierung der Beziehung zwischen Sensoren sehr zeitaufwändig und fehleranfällig. Auch lässt sich in diesem Fall eine spätere Erweiterung der verwendeten Sensoren nur mit viel Mehraufwand realisieren.

4.4.2 Systemgestützte Bestimmung von Qualitätsinformationen

Bei der systemgestützten Modellierung von Beziehung ist das Ziel, eine Beziehung einmalig zu modellieren und in unterschiedlichen Anwendungen nutzen zu können. Als Werkzeug für die Modellierung bietet sich hier eine Ontologie an, wie sie bereits auch in anderen Arbeiten [GQWJ11, FMS+10] verwendet wurde. Wie bereits von den Autoren in [GQWJ11] angemerkt, bringt die Modellierung der Qualität von Sensorbeobachtungen in einer Ontologie den Vorteil, dass sowohl Domänenwissen, Konzepte und ihre Beziehungen modelliert werden können. Zudem wird so eine Modularisierung ermöglicht, wodurch auch die Wiederverwendbarkeit der modellierten Informationen erhöht wird. Außerdem können Anwender durch die Vielzahl an vorhandenen Modellierungswerkzeugen das Wissen, welches in einer Ontologie gespeichert ist, jederzeit erweitern und verändern.

In [FMS+10] erläutern die Autoren ihr Kontextqualitätsmodell und mehrere Messmethoden um die Kontextqualität zu bestimmen. Hierfür modellieren die Autoren zwei Ontologien für den Benutzerkontext und für die Kontextqualität. In der Benutzerkontextontologie klassifizieren die Autoren Kontextinformation nach räumlich, zeitlich, räumlichzeitlich, Sozialen- und Rechendimensionen. In der Kontextqualitätsontologie klassifizieren die Autoren mehrere Kontextqualitäten. Des Weiteren messen die Autoren die Kontextqualität von abgeleiteten und hergeleiteten Kontextinformationen durch die Bildung des Minimums, Maximums und Durchschnitts von mehreren Kontextqualitätsindikatoren. Für die Berechnung der Kontextqualitätsindikatoren nutzen die Autoren dabei relative Messungen. So wird etwa die Präzision auf Basis der Anzahl von Präzisionsstufen berechnet. Diese Form der relativen Bestimmung von Qualitätsinformation macht den Ansatz allerdings nicht nutzbar für Anwendungen, in denen neue zuvor unbekannte Quellen hinzugefügt oder entfernt werden sollen. Außerdem modelliert der Ansatz nicht die Beziehungen zwischen Sensoren und den Qualitäten von Kontextinformation auf Basis von äußeren Einflüssen, was allerdings benötigt wird um den Einfluss von unterschiedlichen Messgrößen auf einen Sensor zu bestimmen.

Die Autoren von [GQWJ11] beschreiben in ihrer Arbeit ein ontologiebasiertes Datenqualitätssystem für das Global Sensor Network[1] Datenstrommanagementsystem. In der vorgestellten Arbeit unterscheiden die Autoren zwischen drei Qualitätsmetriken für ihr System: Anfragebasierte Metriken, inhaltsbasierte Metriken und anwendungsbasierte Metriken. Anfragebasierte Metriken sind semantische Regeln und Funktionen für die Messung von Datenqualitäten. Inhaltsbasierte Metriken sind Methoden, die die Datenqualität für spezifische Operatoren in dem DSMS messen, beispielsweise wird die Granularität eines Aggregationsoperators durch die durchschnittliche Granularität der Eingangstupel ermittelt. Die anwendungsspezifischen Metriken sind benutzerdefinierte Funktionen für eine spezifische Anwendung. Um dies zu erreichen nutzen die Autoren die Ontologie um anwendungsspezifische und systemspezifische Datenqualitätsinformationen zu hinterlegen. In der Ontologie selbst definieren die Autoren semantische Regeln in Form von mathematischen Ausdrücken. Diese semantischen Regeln werden anschließend verwendet um Aktionen auszulösen, wie etwa das Hinzufügen und das Entfernen von zusätzlichen Quellen, wenn die Qualität unter einen bestimmten Schwellwert sinkt. Hierfür referenzieren die mathematischen Ausdrücke in der Ontologie auf Attribute innerhalb des Schemas des Datenstroms. Der implementierte Dienst für die Datenqualität ist dabei in zwei Gruppen aufgeteilt. Die erste Gruppe ist für die Offline-Verarbeitung während des Starts einer Anfrage zuständig. Hierbei wird die Anfrage, welche in SQL vorliegt, durch zusätzliche Attribute, welche die Qualitätsinformationen tragen, erweitert. Die zweite Gruppe ist während der Laufzeit der Anfrage dafür zuständig, die zuvor genannten semantischen Regeln auszuführen. Hierzu wird die Ontologie für jedes eintreffende Nutzdatentupel angefragt und die Regeln ausgewertet. Die Autoren haben somit eine Beziehung zwischen Datenqualitätsdimensionen und SQL hergestellt. Jedoch ist zu beachten, dass der vorgestellte Ansatz keine Beziehung zwischen Sensoren und wahrgenommenen Eigenschaften in der Umwelt herstellt. Somit sind die Qualitätsdimensionen auf einen einzigen Datenstrom und die Elemente in diesem Strom limitiert. Eine Beziehung zwischen mehreren Sensoren ist in der vorgestellten Ontologie nicht möglich. Außerdem sind die Bezeichner der Attribute in einer semantischen Regel fest und somit an das Schema eines Datenstroms gebunden. Eine Wiederverwendung von semantischen Regeln für einen anderen Datenstrom ist nicht möglich ohne hierzu explizit die semantischen Regeln anzupassen.

In den beiden vorgestellten Arbeiten definierten die Autoren jeweils ihre eigene Ontologie um Qualitätsinformationen über Sensorbeobachtungen zu speichern und aufzulösen. Dieses birgt prinzipiell das Problem einer Insellösung. Hier wäre eine Lösung wünschenswert die auf einer bereits existierenden Ontologie aufbaut, welche auch eine entsprechende Verbreitung aufweist. Eine solche Ontologie ist beispielsweise die Semantic-Sensor-Network Ontologie welche im Folgenden näher betrachtet werden soll.

[1] `http://gsn.sourceforge.net/`

4.4.3 Semantic-Sensor-Network-Ontologie (SSN)

Die Semantic-Sensor-Network-Ontologie wurde erstmals im Jahr 2010 von der W3C Semantic Sensor Network Incubator Group veröffentlicht. Ziel der Semantic-Sensor-Network-Ontologie (SSN) ist es, neben der syntaktischen Interoperabilität von Sensoren, wie sie von der Sensor Model Language (SensorML) und der Observations and Measurements (O&M) der OGC Sensor Web Enablement bereitgestellt wird, auch eine zusätzliche Schicht zu bieten um die semantische Kompatibilität zu gewährleisten. Die SSN-Ontologie wurde unter anderem in dem Projekt SPITFIRE[2] eingesetzt um Sensornetzwerkgeräte und ihre Fähigkeiten zu beschreiben. In dem Projekt SemSorGrid4Env[3] dient sie zur Repräsentation eines gemeinsamen Datenmodells und in dem Projekt IoT.est[4] wird die SSN-Ontologie verwendet um Sensorressourcen und das System welches sie formen, sowie ihre Wahrnehmungen und Messinformation, zu beschreiben.

In [CBB+12] beschreiben die Autoren den Beginn und die Architektur der SSN-Ontologie. Die Gruppe fokussiert dabei vier Kategorien von Anwendungsszenarien, nämlich die Datenermittlung und Verknüpfung, die Geräteerkennung und -auswahl, die Diagnose und Fehlerherkunft und den Gerätebetrieb und die Geräteprogrammierung. Die SSN-Ontologie ist in die 10 konzeptionellen Modelle Bereitstellung, System, Betriebsbeschränkung, Plattform, Gerät, Verfahren, Daten, Skelett, Messfähigkeit und Einschränkungsblock unterteilt (vgl. Abb. 4.3) und besteht aus 41 Konzepten und 39 Objekteigenschaften, die von 11 Konzepten und 14 Objekteigenschaften der Descriptive Ontology for Linguistic and Cognitive Engineering Ontologie (DOLCE) abgeleitet sind. Die DOLCE-Ontologie ist selbst wieder eine Ontologie zur Abbildung der ontologischen Kategorien der zugrunde liegenden natürlichen Sprache.

Die SSN-Ontologie kann nach Auffassung der Autoren von 4 Perspektiven gesehen werden: Die Sensorperspektive mit dem Fokus auf was beobachtet wird, wie es beobachtet wird und was beobachtet; Die Daten- und Beobachtungsperspektive, die Beobachtungsdaten und zugehörige Metadaten beschreibt; Die Systemperspektive um Systeme von Sensoren zu repräsentieren; Die Merkmals- und Eigenschaftenperspektive mit Wissen über Merkmale, Eigenschaften von Merkmalen und was diese Eigenschaften beobachten kann.

Die SSN-Ontologie ist auf Basis der Web Ontology Language (OWL)[5] realisiert worden. Die OWL ist eine semantische Erweiterung der Resource Description Framework Schema (RDFS) hervorgebracht durch das W3C um Wissen über Klassen von Individuen und Beziehung zwischen ihnen zu repräsentieren. Eine OWL-Wissensbasis kann in verschiedene Formate serialisiert werden, z.B. in das Turtle oder das XML Format. Auf diese Weise kann die Wissensbasis verteilt und in unterschiedlichen Anwendungen wiederverwendet werden. Im Folgenden fokussieren wir uns auf die Sensorperspektive der SSN-Ontologie

[2] http://spitfire-project.eu
[3] http://semsorgrid4env.eu
[4] http://ict-iotest.eu
[5] http://www.w3.org/TR/owl-ref

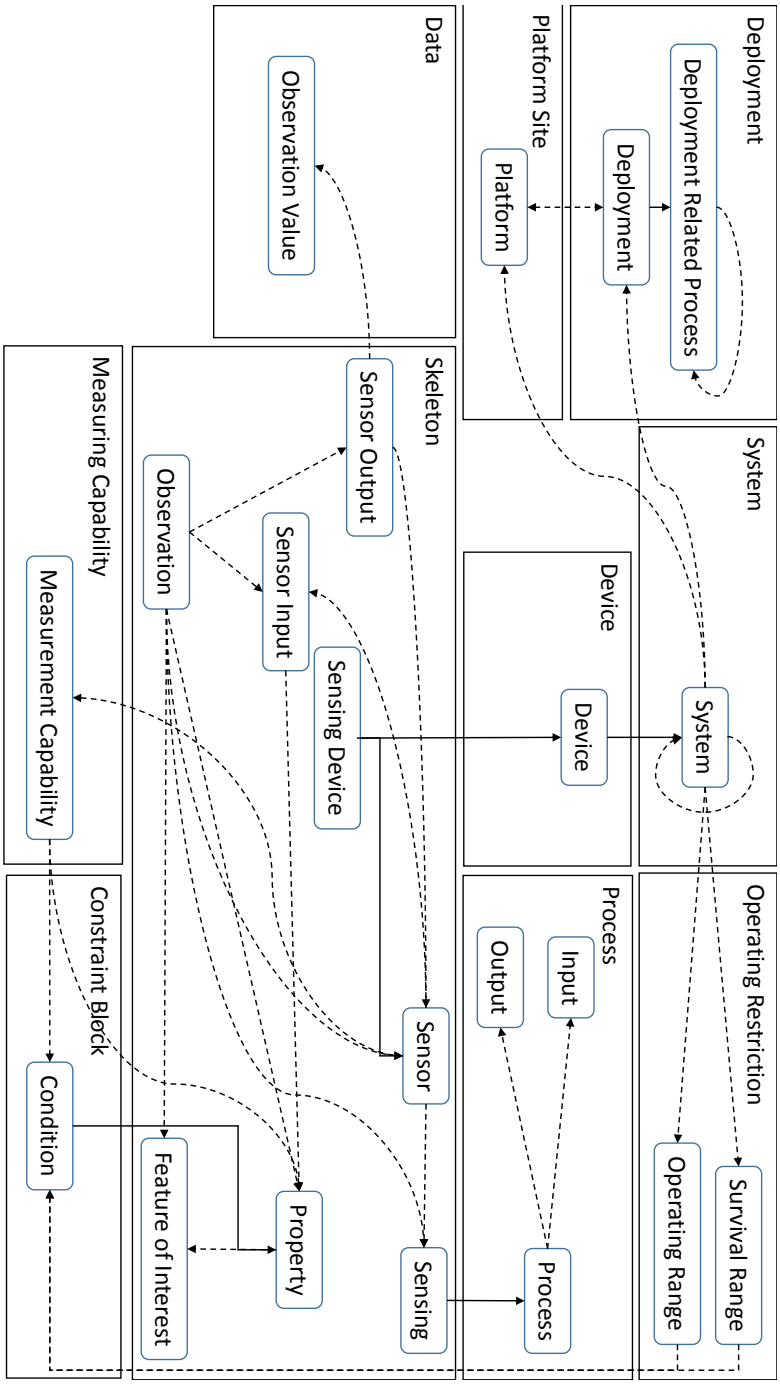


Abbildung 4.3: *Konzeptionelle Modelle der SSN-Ontologie*

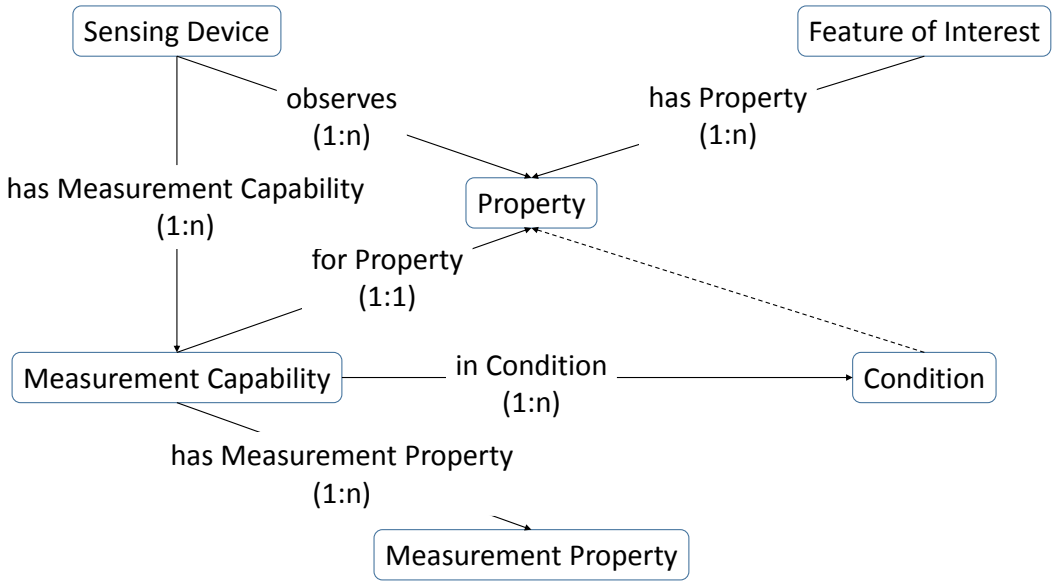


Abbildung 4.4: *Sensorperspektive der SSN-Ontologie*

(Abb. 4.4) um die Beziehungen zwischen Sensoren und ihrer Wahrnehmungen in unserem System zu beschreiben. Ein Sensorgerät (*Sensing Device*) kann dabei mehrere Eigenschaften (*Property*) in einem Einsatzgebiet (*Feature of Interest*) durch seine Messmöglichkeiten (*Measurement Capability*) wahrnehmen. Jede dieser Messmöglichkeiten ist dabei unter bestimmten Bedingungen (*Condition*) definiert und weist innerhalb dieser Bedingungen bestimmte Messeigenschaften (*Measurement Property*) auf. Die Messeigenschaften stellen dabei Qualitätsdimensionen der Sensorwahrnehmung dar. Die SSN-Ontologie verfügt bereits über die zuvor erläuterten Qualitätsdimensionen, sowie einige weitere Qualitätsdimensionen, die in den bisher betrachteten Arbeiten zur Beschreibung der Qualität von Kontextinformationen allerdings nicht verwendet wurden.

Eine Qualitätsdimension für Sensorbeobachtungen ist in der Ontologie (vgl. Abb. 4.5) als eine Erweiterung der Klasse *Measurement Property* dargestellt, welche von der Klasse *Property* ableitet. Eine *Measurement Property* kann dadurch mit jedem Sensorgerät verknüpft werden und wird dadurch selbst zu einer wahrnehmbaren Eigenschaft in der modellierten Welt.

Im Folgenden werden die Klassennamen der Qualitätsdimension und ihre Definition, wie sie in der SSN-Ontologie definiert sind, erläutert:

Genauigkeit (Accuracy)
Der Grad der Übereinstimmung zwischen dem Wert einer Beobachtung und dem wahren Wert einer beobachteten Qualität.

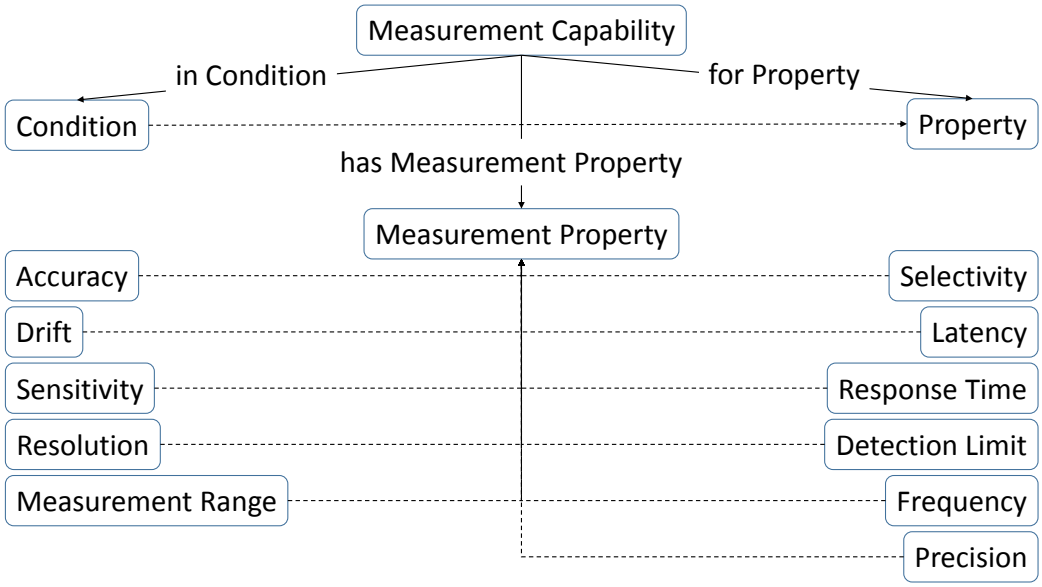


Abbildung 4.5: *Qualitätsdimensionen in der SSN-Ontologie*

Nachweisgrenze (Detection limit)
Ein beobachteter Wert, für den die Wahrscheinlichkeit, fälschlicherweise die Abwesenheit einer Komponente in einem Material zu behaupten β ist, unter einer Wahrscheinlichkeit von α für die fälschlicherweise Behauptung der Anwesenheit.

Drift (Drift)
Eine kontinuierliche, inkrementelle Änderung der gemeldeten Werte von Beobachtungen im Laufe der Zeit für eine gleichbleibende Qualität.

Frequenz (Frequency)
Die kleinstmögliche Zeit zwischen einer Beobachtung und der nächsten Beobachtung.

Latenz (Latency)
Die Zeit zwischen einer Anforderung für eine Beobachtung und der Antwort des Sensors mit einem Ergebnis.

Messbereich (Measurement range)
Die Menge von Werten, die ein Sensor als Ergebnis einer Beobachtung unter den definierten Bedingungen mit den definierten Messeigenschaften zurück liefern kann. Wenn keine Bedingungen angegeben werden oder die Bedingungen nicht einen Bereich für die beobachteten Eigenschaften angeben, ist der Messbereich als die Bedingung für die beobachteten Eigenschaften anzunehmen.

Präzision (Precision)
Der Grad der Übereinstimmung zwischen sich wiederholenden Beobachtungen auf un-

veränderte oder ähnliche Qualitäten, d.h., ein Maß für die Fähigkeit eines Sensors, eine Beobachtung konsequent zu reproduzieren.

Ansprechzeit (Response time)
Die Zeit zwischen einer Veränderung des Wertes einer beobachteten Qualität und dem (möglicherweise mit bestimmten Fehlern) setzen des Sensors auf einen beobachteten Wert.

Auflösung (Resolution)
Der kleinste Unterschied in dem Wert einer Qualität welche beobachtet wird, die zu einem deutlich unterschiedlichen Wert für ein Beobachtungsergebnissen führt.

Empfindlichkeit (Sensitivity)
Der Quotient aus der Änderung in Folge der Sensorbeobachtung und der entsprechenden Änderung in einem Wert einer Qualität.

Selektivität (Selectivity)
Selektivität ist eine Eigenschaft des Sensors, wodurch es ermöglicht wird Beobachtungswerte für eine oder mehrere Eigenschaften bereit zu stellen, so dass die Werte jeder Qualität unabhängig von anderen Eigenschaften in der Erscheinung, dem Körper oder der Substanz untersucht werden können.

Neben den, auf Sensorbeobachtung bezogenen, Qualitätsdimensionen werden die Qualitätsdimensionen für den Überlebensbereich und den Betriebsbereich des gesamten Sensorgeräts von der SSN-Ontologie wie folgt definiert:

Überlebensbereich (Survival range)
Die Bedingungen unter denen ein Sensor ohne Beschädigung ausgesetzt werden kann, d.h., der Sensor arbeitet wie definiert unter Verwendung seiner Messmöglichkeiten. Wenn jedoch der Überlebensbereich überschritten wird, ist der Sensor beschädigt und die definierten Messmöglichkeiten können nicht länger bestehen.

Betriebsbereich (Operating range)
Die Umweltbedingungen und Eigenschaften einer normalen Betriebssystemumgebung für ein System oder einen Sensor. Dies kann verwendet werden, um beispielsweise die Standardumgebungsbedingungen, in denen der Sensoreinsatz vorgesehen ist (eine Bedingung ohne Betriebseigenschaften), oder wie die Umwelt- und sonstige betrieblichen Eigenschaften in Beziehung stehen.

Auf diese Weise enthält die SSN-Ontologie Konzepte zur Wahrnehmung von Qualitäten von Sensorbeobachtungen durch andere Sensoren und gibt uns auf diese Weise die Möglichkeit die Beziehungen zwischen Sensoren, ihren beobachteten Eigenschaften und den dabei geltenden Qualitäten zu modellieren.

4.4.3.1 Beispiel

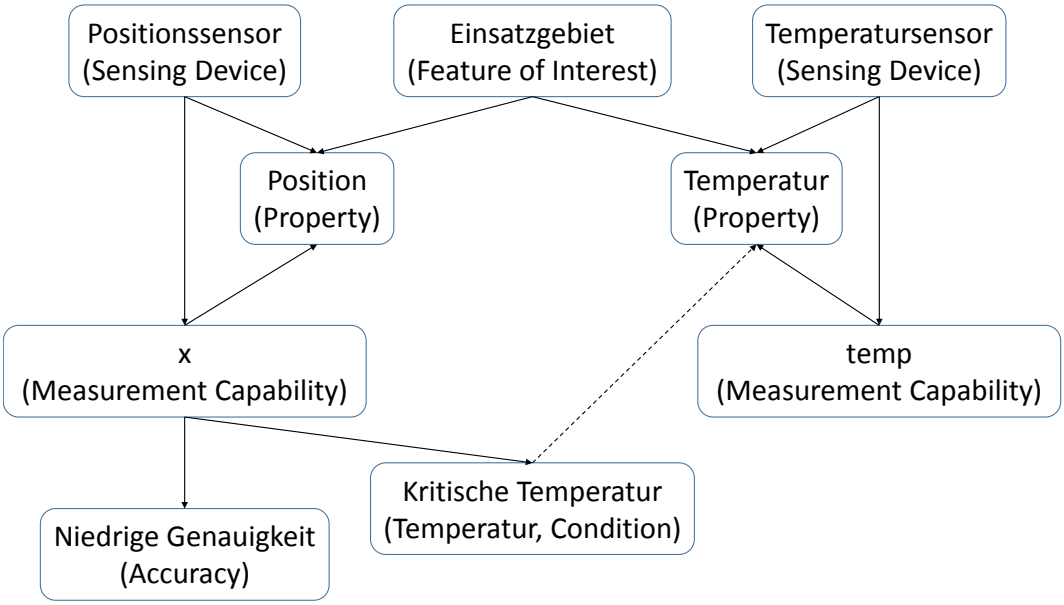


Abbildung 4.6: *Sensormodellierung am Beispiel eines Positionssensors*

Zum besseren Verständnis der Modellierung von Beziehungen zwischen Sensoren in der SSN-Ontologie betrachten wir das Beispiel in Abb. 4.6. In diesem Beispiel existieren zwei Instanzen von Sensoren, die unterschiedliche Eigenschaften ihrer Umgebung überwachen. In der Abbildung ist die jeweils abgeleitete Klasse in Klammern dargestellt. Der Temperatursensor überwacht dabei die Eigenschaft *Temperatur* und der Positionssensor die Eigenschaft *Position*. Hierfür hat der Temperatursensor die Messmöglichkeit *temp* welche mit der Eigenschaft *Temperatur* verknüpft ist. Der Positionssensor, welcher die Eigenschaft *Position* überwacht, hat entsprechend die Messmöglichkeit *x*, welche zusätzlich noch mit der Eigenschaft *Niedrige Genauigkeit* verknüpft ist. Die Eigenschaft *Niedrige Genauigkeit* gibt die Genauigkeit einer Messung wieder. Des Weiteren ist die Messmöglichkeit des Positionssensors mit der Bedingung *kritische Temperatur* verknüpft, daher gilt die Angabe der Genauigkeit nur wenn diese Bedingung erfüllt ist. In diesem Beispiel ist die Bedingung *kritische Temperatur* direkt von der Eigenschaft *Temperatur* abgeleitet. Allerdings verfügt der Positionssensor über keine Messmöglichkeit um diese Eigenschaft zu überwachen. Jedoch kann nun auf Basis des Wissens in der Ontologie ein Sensor gefunden werden, der über eine Messmöglichkeit verfügt um diese Eigenschaft zu erfassen. Ist diesem einfachen Beispiel ist dies der Temperatursensor, der diese Eigenschaft durch eine Messmöglichkeit innerhalb des betrachteten Einsatzgebietes überwachen kann. Hierdurch kann wiederum die Bedingung für die Messmöglichkeit der Positionssensoren auf Basis der Messungen des Temperatursensors ausgewertet werden und somit die Genauigkeit des Positionssensors ermittelt werden.

4.4.4 Diskussion

Bei der indirekten Bestimmung von Qualitätsdimensionen wurde gezeigt, wie sowohl über den Anwender, wie auch über eine, in einer Ontologie hinterlegten, Wissensbasis die Beziehungen zwischen Sensoren und ihren Wahrnehmungen formuliert werden können. Dabei eignet sich die nutzergestützte Bestimmung von Qualitätsinformationen zunächst nur für kleinere Anwendungen mit gleichen Sensoren. Dagegen bietet die systemgestützte Bestimmung eine allgemeinere Modellierung der Beziehung und der damit verknüpften Qualitätsinformationen. Bei der verwendeten Ontologie sind bereits eine Vielzahl von Qualitätsdimensionen enthalten, so dass die zuvor besprochenen Qualitätsdimensionen direkt abgebildet werden können bzw. auf deren Basis die Qualitätsindikatoren Zeitnähe, Vollständigkeit, Konsistenz und Genauigkeit auf dem Sensordatenstrom bestimmt werden können. Durch die Verwendung der SSN-Ontologie ist zudem eine entsprechende Verbreitung der Technologie sichergestellt, welche in bisherigen Arbeiten, die auf einer Ontologie aufbauten, nicht gegeben ist.

4.5 Direkte Bestimmung von Qualitätsinformationen

Durch die Verwendung der SSN-Ontologie haben wir bereits die Möglichkeit geschaffen, in Abhängigkeit der Wahrnehmung anderer Sensoren und dem hinterlegten Wissen über die Beziehungen der Sensoren und ihrer Wahrnehmungen in der Ontologie, Qualitätsinformationen abhängig von vordefinierten Bedingungen abzufragen. Die Informationen in einer Ontologie bieten zwar die Möglichkeit an, Wissen, was in der Ontologie initial hinterlegt wurde, in unterschiedlichen Szenarien und Anwendungen immer wieder zu verwenden. Allerdings sind gerade Qualitätsdimensionen wie die Genauigkeit stark sensor- und applikationsabhängig, so dass eine allgemeine Modellierung etwa auf Basis des Datenblatts des Sensors nicht den exakten Wert der Qualitätsinformation wiedergeben kann bzw. die Messwerte ungenauer darstellt werden als sie es zum Zeitpunkt der Messung sind. Zwar könnte für jede Eigenschaft und jede Kombination von Umweltbedingungen eine Prüfbedingung in der Ontologie hinterlegt werden, um so etwa die aktuell herrschende Genauigkeit eines Sensors zu modellieren. Allerdings wäre die Modellierung der Abhängigkeiten in der Ontologie sehr aufwendig, wenn nicht sogar unmöglich, da nicht jede Eigenschaft, die eine Auswirkung auf die Genauigkeit eines Sensors ausübt, schon allein aus Kostengründen überwacht werden kann. Im Folgenden soll daher nun das Ziel verfolgt werden, Qualitätsinformationen der Sensorwahrnehmungen allein auf den Messwerten des Sensors zu ermitteln.

Die grundlegende Idee dabei ist, das hinter den Messwerten liegende stochastische Modell der Verteilung von Messwerten auf Basis der Sensorbeobachtungen zu erlernen. Zur Bestimmung des stochastischen Modells existieren verschiedene Ansätze. Im einfachsten Fall lässt sich unter Annahme einer Normalverteilung der Messwerte die Varianz und der Erwartungswert anhand mehrerer Stichprobe, etwa durch ein Zeitfenster, bestimmen. Ist

die Art der Verteilung der Sensoren in dem betrachteten Anwendungsszenarium nicht im vornherein bekannt oder ändert es sich über die Zeit, bieten sich Verfahren an, die das stochastische Modell annähern. Zu diesen Verfahren zählen das Erwartungswertmaximierungsverfahren [DLR77] und die Kerndichteschätzung [Sil86]. Ein Vergleich verschiedener Verfahren findet sich in [DW04].

4.5.1 Bestimmung von Erwartungswert und Varianz

Der Erwartungswert eines normalverteilten Wertes lässt sich inkrementell mit jeder Sensorwahrnehmung v wie folgt bestimmen. Sei hierzu n die Anzahl der bisherigen Sensorwahrnehmungen, dann gilt für den Erwartungswert μ:

$$\mu^{(t+1)} = \mu^{(t)} + \frac{v - \mu^{(t)}}{n} \tag{4.8}$$

Aufbauend darauf lässt sich die Standardabweichung eines Wertes nach [Knu97] inkrementell durch den Erwartungswert und die Sensorwahrnehmungen bestimmen. Sei hierzu n die Anzahl der Sensorwahrnehmungen, dann gilt für die Standardabweichung σ:

$$\Delta^{(t+1)} = \Delta^{(t)} + (v - \mu^{(t)})(v - \mu^{(t+1)}) \tag{4.9}$$

$$\sigma^{(t+1)} = \frac{\Delta^{(t+1)}}{n-1} \tag{4.10}$$

Die Standardabweichung lässt sich somit zu jedem Zeitpunkt aus der Anzahl der bisher wahrgenommenen Werte n und der Differenz der quadratischen Summen Δ bestimmen. Für die Gauß-Verteilung als stochastisches Modell für die wahrgenommenen Werte ergibt sich dann $\mathcal{N}(\mu, \sigma)$ mit dem Erwartungswert μ und der Standardabweichung σ.

Auch im Mehrdimensionalen Fall lässt sich die Kovarianz zwischen Sensorwahrnehmungen inkrementell wie folgt bestimmen:

$$COV_{x,y} = \frac{\sum xy - \frac{(\sum x \sum y)}{n}}{n-1} \tag{4.11}$$

4.5.2 Erwartungswertmaximierungsverfahren

Das Erwartungswertmaximierungsverfahren [DLR77] bietet allgemein die Möglichkeit die Parameter eines stochastischen Modells an die Verteilung von Daten anzunähern. Hierzu wird versucht die Log-Likelihood L zwischen den zu bestimmenden Parametern und den zur Verfügung stehenden Daten in jeder Iteration t des Algorithmus zu maximieren. Als Parameter bieten sich hierfür die Parameter einer multivariaten Mischverteilung aus

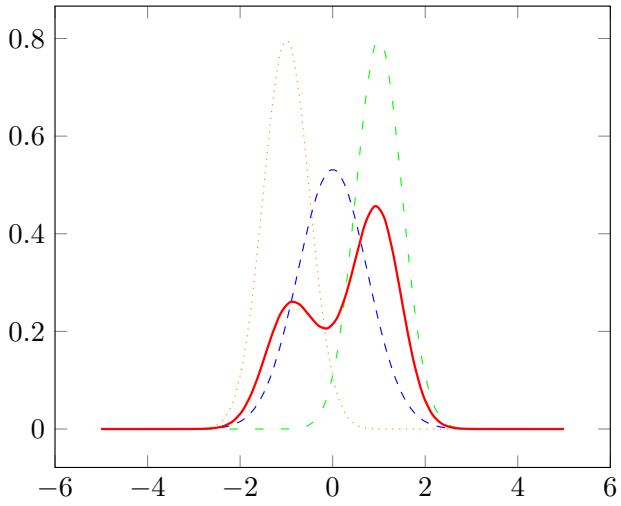


Abbildung 4.7: *Beispiel einer Gauß-Mischverteilung mit den Verteilungen* $\mathcal{N}_1(-1, 0.5)$, $\mathcal{N}_2(0, 0.75)$ *und* $\mathcal{N}_3(1, 0.5)$, *sowie der Gewichtung* $w_1 = 0.25$, $w_2 = 0.25$ *und* $w_3 = 0.5$

Gauß-Verteilungen mit Parameter $\theta = \{w_i, \mu_i, \Sigma_i\}_{i=1}^m$ an, wobei w_i die Gewichtung der i-ten Gauß-Verteilung mit Erwartungswert μ_i und Kovarianzmatrix Σ_i repräsentiert. Eine multivariate Mischverteilung aus Gauß-Verteilungen über eine kontinuierliche Zufallsvariable X ist eine Menge von m gewichteten Gauß-Verteilungen $X_1, X_2, \ldots, X_m$, wobei X die Wahrscheinlichkeitsdichtefunktion

$$f_X(x) = \sum_{i=1}^{m} w_i f_{X_i}(x) \tag{4.12}$$

mit

$$f_{X_i}(x) = \frac{1}{(2\pi)^{k/2}|\Sigma_i|^{1/2}} e^{-\frac{1}{2}(x-\mu_i)^T \Sigma_i^{-1}(x-\mu_i)} \tag{4.13}$$

ist. Dabei gilt, dass $0 \leq w_i \leq 1, \sum_{i=1}^m w_i = 1$, k die Größe des Zufallsvektors ist und jede Mischverteilungskomponente X_i eine k-variate Gauß-Verteilung (Fig.4.7) mit Erwartungswert μ_i und Kovarianzmatrix Σ_i ist. Zur Annäherung einer Gauß-Mischverteilung wird zunächst ein initiales stochastisches Modell mit m Gauß-Verteilungen bestimmt. Auf Basis dieses Modells wird anschließend die Log-Likelihood zwischen dem Modell und den zugrunde liegenden Daten bestimmt.

$$L^{(t)} = \frac{1}{n} \sum_{i=1}^{n} \log(\sum_{j=1}^{m} w_j^{(t)} F_{X_j}(x_i; \theta_j^{(t)}))$$

Auf Basis des aktuellen Modells werden nun im Erwartungswert-Schritt die Erwartungswerte bestimmt, also die Wahrscheinlichkeiten, dass die aktuellen Werte aus dem aktuellen stochastischen Modell generiert wurden.

$$\tau_{ij}^{(t)} = \frac{w_j^{(t)} F_{Xj}(x_i; \theta_j^{(t)})}{\sum_{l=1}^{m} w_l^{(t)} F_{Xj}(x_i; \theta_l^{(t)})}, i = 1, \ldots, n, j = 1, \ldots, m$$

$$\gamma_j^{(t)} = \sum_{i=1}^{n} \tau_{ij}^{(t)}, j = 1, \ldots, m$$

Während des Maximierungs-Schritts werden die neuen Parameter für θ anhand der Ergebnisse aus dem Erwartungswert-Schritt bestimmt.

$$w_j^{(t+1)} = \frac{\gamma_j^{(t)}}{n}, j = 1, \ldots, m$$

$$\mu_j^{(t+1)} = \frac{1}{\gamma_j^{(t)}} \sum_{i=1}^{n} \tau_{ij}^{(t)}, j = 1, \ldots, m$$

$$\Sigma_j^{(t+1)} = \frac{1}{\gamma_j^{(t)}} \sum_{i=1}^{n} \tau_{ij}^{(t)} (x_i - \mu_j^{(t+1)})(x_i - \mu_j^{(t+1)})^T, j = 1, \ldots, m$$

Nach jeder Iteration des Erwartungswertmaximierungsverfahrens wird die neue Log-Likelihood $L^{(t+1)}$ berechnet und mit dem gegebenen Schwellwert $|L^{(t+1)} - L^{(t)}| < \delta$ verglichen. Ist die Differenz kleiner als der gegebene Schwellwert δ oder überschreitet die Anzahl der Iterationen die maximale Anzahl, so werden die bestimmten Parameter für die Gewichte (w), den Erwartungswert (μ), sowie die Kovarianzmatrix (Σ) der Mischverteilung zurückgeliefert. Auf diese Weise können nun die Qualitätsinformationen über die Genauigkeit der Daten im Sinne des statistischen Fehlers angenähert werden. Da die Bestimmung einer möglichen Parameterkonstellation für die Mischverteilung iterativ über mehrere Sensormesswerte arbeitet, bedeutet dies in Bezug auf die Datenstromverarbeitung auch, dass die Bestimmung der Parameter eine gewisse Latenz aufweist. Um dieses Phänomen möglichst gering zu halten bietet es sich an, ein solches Verfahren nur unter bestimmten Voraussetzung auszuführen, etwa wenn die Log-Likelihood zwischen stochastischem Modell und aktuellen Messwert oberhalb eines Grenzwertes liegt, oder ein zuvor definiertes Prädikat über die aktuelle Sensormessung als gültig evaluiert wird.

4.5.3 Kerndichteschätzungsverfahren

Bei dem Kerndichteschätzungsverfahren wird aus jedem Datenpunkt eine multivariate Gauß-Verteilung gebildet, wobei der Datenpunkt selbst den Erwartungswert repräsentiert

und eine konstante Kovarianzmatrix verwendet wird. Aus der Summe aller so gebildeten Verteilungen mit jeweils gleicher Gewichtung entsteht die entsprechende Mischverteilung, welche mit einer Bandbreite geglättet wird um Datenpunkte möglichst gut wiederzugeben.

Zur Wahl der optimalen Bandbreite existieren mehrere Verfahren, wie etwa die Scott-Regel [Sco92] oder die Silverman-Regel [Sil86]. Beide Regeln werden auf Basis der Anzahl der Datenpunkte n und der Dimension d wie folgt bestimmt.

$$\begin{aligned} B_{Scott} &= n^{\frac{-1}{d+4}} \\ B_{Silverman} &= n(\frac{d+2}{4})^{\frac{-1}{d+4}} \end{aligned} \tag{4.14}$$

Ein Nachteil des Kerndichteschätzungsverfahrens besteht in der Menge der einzelnen Gauß-Verteilungen in der resultierenden Mischverteilung, was die Berechnung von Integralen bei einer weiteren Verarbeitung der Mischverteilung erheblich erschwert.

4.5.4 Bregman-Hard Clustering

Da bei dem Kerndichteschätzungsverfahren die Anzahl an Komponenten der Mischverteilung linear zu der Zahl der Messwerte steigt und somit das Ergebnis generell ungeeignet ist für eine Verarbeitung in einem Datenstrommanagementsystem, benötigt es ein Verfahren zur Reduktion der Komponenten. Hierzu bietet sich das Bregman-Hard Clustering [BMDG05] an. Bei dem Bregman-Hard Clustering Verfahren wird versucht, ähnliche Verteilungen innerhalb einer Mischverteilung durch eine einzige Verteilung zu vereinfachen. Hierbei werden zunächst Cluster gebildet, wobei jedes Cluster einen Repräsentanten hat. Anschließend wird für jedes Cluster eine Minimierung ausgeführt mit dem Ziel, den Informationsverlust zwischen dem Repräsentanten und den Komponenten zu minimieren. Das Verfahren kann als eine Generalisierung des Euklidischen k-Means-Verfahrens angesehen werden, wobei die Kullback-Leibler Divergenz als Minimierungsziel verwendet wird. Die Kullback-Leibler Divergenz zweier Normalverteilungen mit Erwartungswert μ und Varianz σ^2 gibt die relative Entropie wieder und ist definiert als:

$$KL(\mathcal{N}(x;\mu_i,\sigma_i^2)||\mathcal{N}(x;\mu_j,\sigma_j^2)) = \int_{x\in\mathbb{R}^d} \mathcal{N}(x;\mu_i,\sigma_i^2)\log\frac{\mathcal{N}(x;\mu_i,\sigma_i^2)}{\mathcal{N}(x;\mu_j,\sigma_j^2)}dx \tag{4.15}$$

Ziel des Clustering-Verfahrens ist es Cluster zu bilden, die die Kullback-Leibler Divergenz zwischen den Komponenten minimieren. Um dabei allerdings die Bestimmung des Integrals innerhalb der Kullback-Leibler Divergenz zu umgehen, wird die Kullback-Leibler Divergenz in eine Bregman Divergenz umgewandelt.

Die Bregman Divergenz ist definiert als:

$$D_F(\theta_j||\theta_i) = F(\theta_j) - F(\theta_i) - < \theta_j - \theta_i, \nabla F(\theta_i) > \tag{4.16}$$

Die Kullback-Leibler Divergenz kann nach [NN09] dabei als Mischtyp Bregman Divergenz gesehen werden. Hierbei wird die Dichtefunktion einer Normalverteilung in die kanonische Dekomposition der jeweiligen Exponentialfamilie wie folgt umgeschrieben:

$$\mathcal{N}(x;\mu,\sigma^2) = exp\{<\theta, t(x)> - F(\theta) + C(x)\} \tag{4.17}$$

Wobei $\theta = (\theta_1 = \frac{\mu}{\sigma^2}, \theta_2 = -\frac{1}{2\sigma^2})$ die natürlichen Parameter, $t(x) = (x, x^2)$ die notwendige Statistik und $F(\theta) = -\frac{\theta_1^2}{4\theta_2} + \frac{1}{2}\log\frac{-\pi}{\theta_2}$ die Log-Normalisierung für eine Normalverteilung darstellen.

Unter der Bedingung, dass beide Verteilungen von der gleichen Exponentenfamilie stammen, lässt sich die Kullback-Leibler Divergenz in die Bregman Divergenz umformen

$$KL(\mathcal{N}(x;\mu_i,\sigma_i^2)||\mathcal{N}(x;\mu_j,\sigma_j^2)) = D_F(\theta_j||\theta_i) \tag{4.18}$$

, so dass nun direkt die Bregman Divergenz als Distanz innerhalb des k-Means-Verfahrens angewendet werden kann um die Cluster zu bilden.

4.5.5 Diskussion

Bei der direkten Bestimmung von Qualitätsdimensionen wurde gezeigt, wie auf Basis der Messwerte eines Sensors das stochastische Modell der Verteilung der Werte in einem Datenstrom bestimmt werden können und so die Berechnung der Qualitätsdimension Genauigkeit im Sinne des statistischen Fehlers erlaubt. Je nach Kenntnis über die zu verarbeitenden Sensorwahrnehmungen kann entweder direkt, unter Annahme einer Normalverteilung, die Parameter auf Basis der Daten berechnet werden oder durch Nutzung eines Algorithmus, wie dem Erwartungswertmaximierungsverfahren oder dem Kerndichteschätzverfahren, die Parameter des unbekannten stochastischen Modells in Form einer Mischverteilung angenähert werden.

4.6 Qualitätsannotation in Datenstrommanagementsystemen

Durch die SSN-Ontologie haben wir bereits ein breites Spektrum von vordefinierten Qualitätsdimensionen um die Qualität von Messungen von Eigenschaften unter bestimmten Umweltbedingungen zu beschreiben, so wie etwa die Sensitivität oder die Abweichung einer Messung. Auch lässt sich die Ontologie erweitern, so dass neue Dimensionen bei Bedarf hinzugefügt werden können. Jedoch ist dies nur die Art der Modellierung. Um nun einen Nutzen aus der Ontologie zu ziehen, müssen diese Information mit den aktuellen Daten in einem Datenstrom korrekt verknüpft werden. Zu diesem Zweck ist ein zusätzlicher Operator innerhalb der logischen Operatoralgebra notwendig, der gegeben einem logischen Datenstrom $S^l \in \mathbb{S}^l$, diesen durch die aktuell herrschenden Qualitätsdimensionen d pro Messwert erweitert.

Hierzu müssen die aktuell herrschenden Umweltbedingungen bei jedem Messwert evaluiert werden. Allerdings kann es vorkommen, dass die Auswertung von bestimmten Bedingungen weitere Informationen aus zusätzlicher Quellen benötigen. Daher müssen diese zusätzlichen Quellen zunächst bestimmt und in den Datenstrom integriert werden. Anschließend müssen die Bedingungen ausgewertet und je nach herrschender Bedingung die Werte der, in der Ontologie hinterlegten, Qualitätsdimensionen auf Basis der aktuellen Messwerte in dem nun erweiterten Datenstrom errechnet werden.

Im Folgenden definieren wir hierzu zunächst den Operator zur Integration der indirekten Qualitätsbestimmung auf der logischen Ebene. Daraufhin wird gezeigt, wie die für die Berechnung der Bedingungen notwendigen Quellen bestimmt und hinzugefügt werden. Anschließend wird gezeigt, wie dieser logische Operator auf eine Kombination von existierenden Operatoren aus der temporal relationalen Operatoralgebra abgebildet werden kann, um auf diese Weise das Optimierungspotenzial der temporal relationalen Operatoralgebra zu erhalten.

4.6.1 Logische Integration der indirekten Qualitätsbestimmung

Im Folgenden werden die Information, die innerhalb der Ontologie abgebildet sind, mit den Konzepten der Datenstromverarbeitung kombiniert um damit die Qualitäten in einem logischen Datenstrom zu bestimmen. Sei hierfür $\mathbb{D}$ die Menge aller Qualitätsdimensionen, $\mathbb{S}$ die Menge aller Sensoren und $s \in \mathbb{S}$ ein Sensor. Weiterhin sei $\mathbb{P}$ die Menge aller wahrnehmbaren Eigenschaften und $p \in \mathbb{P}$ eine Eigenschaft, sowie $\mathbb{M}_p^s$ die Menge aller Messmöglichkeiten eines Sensors $s \in \mathbb{S}$ für eine Eigenschaft $p \in \mathbb{P}$ und $m \in \mathbb{M}_p^s$ eine Messmöglichkeit des Sensors s für die Eigenschaft p. Außerdem sei $name(m)$ der Name der Messmöglichkeit in der Ontologie, welcher auch im Schema $\mathcal{T}$ des logischen Datenstroms S_s^l des Sensors enthalten ist. Hierbei ist zu beachten, dass der Name der Messeigenschaft in der Ontologie nicht eindeutig über alle $m \in \mathbb{M}_p^s$ sein muss. Sei des Weiteren $\mathbb{C}^m$ die Menge aller Bedingungen, die für die Messeigenschaft m definiert sind und $c \in \mathbb{C}$ eine Bedingung. Die Bedingung ist dabei eine Funktion $c : \Omega_{\tilde{\mathcal{T}}} \rightarrow [true, false]$, welche eine Prädikatsfunktion auf einem logischen Datenstrom mit Schema $\tilde{\mathcal{T}}$ ausführt und entweder $true$ liefert wenn die Bedingung erfüllt ist oder $false$ liefert wenn die Bedingung nicht erfüllt ist. Das Schema des logischen Datenstroms des Sensors s ist dabei ein Teil des Schemas auf dem das Prädikat definiert sein kann, daher gilt $\mathcal{T} \subseteq \tilde{\mathcal{T}}$. Sei $\mathbb{D}^m$ die Menge aller Qualitätsdimensionen mit $\mathbb{D}^m \subseteq \mathbb{D}$, die für die Messmöglichkeit m definiert sind und $d \in \mathbb{D}^m$ eine Qualitätsdimension. Die Qualitätsdimension ist dabei ebenfalls eine Funktion $d : \Omega_{\tilde{\mathcal{T}}} \rightarrow \Omega_{\tilde{\mathcal{T}}_d}$ welche den logischen Datenstrom $S_{\tilde{\mathcal{T}}}^l$ auf einen logischen Datenstrom mit Schema $\tilde{\mathcal{T}}_d$ abbildet und damit den logischen Datenstrom um die Qualitätsdimensionen erweitert. Hierbei ist zunächst zu beachten, dass prinzipiell mehrere Bedingungen über eine Messmöglichkeit definiert sein können und somit auch über die damit verknüpften Qualitätsdimensionen. Dieses Verhalten wird im Folgenden als eine Kombination von Bedingungen verstanden, sodass die für die Qualitätsdimension zu prüfende Bedingung als

eine Prüfung aller einzelnen Bedingungen zu verstehen ist und die entstehende Bedingung wie folgt definiert ist:

$$\tilde{c}(\hat{e}) = \bigvee_{c \in \mathbb{C}^m} c(e) \tag{4.19}$$

Hierbei ist $\hat{e}$ das Nutzdatentupel aus einem logischen Datenstrom mit Schema $\tilde{\mathcal{T}}$. Möchte man nun die Qualitätsdimensionen für jedes Attribut $a \in e$ eines Nutzdatentupels in einem logischen Datenstrom $S^l_{\mathcal{T}}$ bestimmen, genügt es die Qualitätsdimensionen pro Messmöglichkeit zu bestimmen, die den gleichen Namen trägt wie das Attribut und bei denen die Bedingung $\tilde{c}$ zu $true$ evaluiert. Anschließend müssen die Teilergebnisse nur noch zu einem Gesamtergebnis verdichtet werden.

$$\begin{aligned} q(e) = &\{\hat{d} | \exists M \subseteq \mathbb{M}^s : M \neq \emptyset \wedge M = \{m \in \mathbb{M}^s | name(a) = name(m) \\ &\wedge \tilde{c}(a) \wedge \tilde{c} \in \mathbb{C}^m\} \wedge \hat{d} = Agg_{m \in M} d(a)\}_{a \in e} \end{aligned} \tag{4.20}$$

Unter der Annahme, dass ein hoher Wert der Qualitätsdimension sich prinzipiell negativer auf das Berechnungsergebnis auswirkt als ein niedriger, z.B. eine hohe Latenz oder ein größere Streuung, bietet sich hier als Aggregationsfunktion Agg das Maximum an, um die Werte der Qualitätsdimension bei unterschiedlichen Messmöglichkeiten mit gleichem Namen, bei denen die aktuellen Bedingungen zutreffen, zu verdichten.

Diese Qualitätsfunktion eingesetzt in den logischen Qualitätsoperator ergibt folgende logische Berechnung der Qualitäten eines logischen Datenstroms. Sei $S \in \mathbb{S}^l_{\mathcal{T}}$ dann gilt:

$$\begin{aligned} \mu_q(S) = &\{(\hat{e}, t, \hat{n}) | \exists X \subseteq S : X \neq \emptyset \wedge X = \{(e, t, n) \in S | e \circ q(e) = \hat{e}\} \\ &\wedge \hat{n} = \sum_{(e,t,n) \in X} n\} \end{aligned} \tag{4.21}$$

Zur Bestimmung dieser Qualitätsinformationen werden allerdings, je nach Verknüpfung der in der Ontologie enthaltenen Daten, zusätzliche Quellen notwendig. Die Auswahl der notwendigen Quellen kann allerdings schon auf logischer Ebene geschehen, da sowohl die Quellennamen, wie auch die notwendigen Attribute der Nutzdatentupel aus der Ontologie bekannt sind.

4.6.1.1 Quellenauswahl

Aufbauend auf den, in der Ontologie definierten, Bedingungen für eine Messeigenschaft und der damit verbundenen Qualitätsdimension können die hierfür notwendigen Quellen wie in Listing 4.1 bestimmt werden. Zunächst werden pro Attribut in dem Schema des logischen Datenstroms die, für diese Attribute definierten Bedingungen aus den Verknüpfungen in der Ontologie gesucht. Anschließend werden pro Bedingung die möglichen Sensoren ermittelt, die verwendet werden können um die Bedingung auszuwerten. Hierzu wird zunächst die Methode *parent()* auf der Bedingung ausgeführt. Die Methode

Algorithmus 4.1 : Quellenauswahl für Qualitätsbestimmung

Input : Logischer Datenstrom S_{in}
Output : Menge von logischen Datenströmen $\mathbb{S}_Q$

```
1  S_Q ← ∅;
2  for a ← Ω_T do
3      // Selektiere alle Bedingungen für passende Messmöglichkeiten des Sensors;
4      C ← {c ∈ ℂ^m | m ∈ 𝕄^{S_in} ∧ name(m) = name(a)};
5      for c ← C do
6          // Liefere die abgeleitete Eigenschaft der Bedingung;
7          p ← parent(c̃);
8          // Liefere alle Sensoren, die eine Messmöglichkeit haben um p
           wahrzunehmen;
9          S̃_Q ← {S ∈ 𝕊 | ∃m ∧ m ∈ 𝕄_p^s};
10         if S_in ∉ S̃_Q then
11             S_Q ← S_Q ∪ S̃_Q;
12         end
13     end
14 end
15 return S_Q;
```

parent() liefert dabei die wahrnehmbare Eigenschaft in der Ontologie von der die Bedingung ursprünglich ableitet. Anschließend wird die Ontologie nach der Menge der Sensoren durchsucht, die eine oder mehrere Messmöglichkeit aufweisen um die Eigenschaft wahrzunehmen. Verfügt der logische Datenstrom des Sensors schon über eine Messmöglichkeit die gewünschte Eigenschaft zu erfassen, macht es natürlich Sinn, diese zu nutzen statt eine zusätzliche Quelle anzubinden, da hierdurch zum einen die Datenlast für die zusätzliche Quelle entfällt, aber auch ein Kreuzprodukt mit dieser Quelle vermieden werden kann. Das Einsatzgebiet (*Feature of Interest*) wurde in dem Pseudocode aus Gründen der Lesbarkeit bei der Quellenauswahl vernachlässigt, da es lediglich einen zusätzlichen Filter bei der Wahl der möglichen Sensoren innerhalb der Ontologie darstellt.

4.6.2 Logische Reduktion

Da die primäre Aufgabe des Qualitätsoperators die Abbildung eines Stromelements, unter Verwendung von Bedingungen und Qualitätsinformationen aus der Ontologie und Sensorwahrnehmungen aus anderen Sensordatenströmen, auf ein mit Qualitätsmerkmalen versehenes Stromelement ist, kann dieser auch durch eine Kombination von mehreren Operatoren aus der temporal relationalen Operatoralgebra abgebildet werden. Zunächst kann die Berechnung der Bedingungen und die Ermittlung der Qualitätsdimensionen innerhalb eines Abbildungsoperators μ_q mit der Funktion f_Q abgebildet werden. Das Hinzufügen

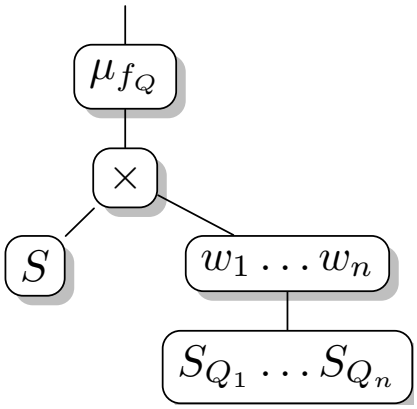


Abbildung 4.8: *Logischer Anfragegraph für die Annotation von Qualitätsinformationen an einen logischen Datenstrom*

von notwendigen Quellen zu dem logischen Datenstrom auf dem die Abbildung ausgeführt wird kann durch einen Verbund der beiden Ströme realisiert werden. Um dieses zu ermöglichen bedarf allerdings noch einer Relation auf der dieser Verbund ermittelt werden kann. Hierzu bietet sich ein gleitendes Zeitfenster an. Das gleitende Zeitfenster nutzt dabei als Größe die Frequenz des Sensors um sicherzustellen, dass immer ein möglicher Partner für das kartesische Produkt zur Verfügung steht.

Sei hierzu $\mathbb{S}^l$ die Menge aller logischen Datenströme und $\mathbb{S}^l_q$ mit $\mathbb{S}^l_q \subseteq \mathbb{S}^l$ die Menge aller logischen Datenströme, die von Quellen stammen, die in der Ontologie mit einer zu überwachenden Eigenschaft verknüpft sind, so wie sie über den Quellenselektionsalgorithmus (Lst. 4.1) bestimmt werden können. Der Qualitätsoperator kann dann wie folgt repräsentiert werden:

$$\mu_q(S) = \mu_{f_Q}(\times(\omega_w(S, \{S_Q | S_Q \in \mathbb{S}^l_q\}))) \tag{4.22}$$

Der Qualitätsoperator kann also auf eine Kombination aus den drei Operatoren der temporal relationalen Operatoralgebra Abbildung, Verbund und Zeitfenster abgebildet werden. Somit lassen sich Elemente in dem logischen Datenstrom auf Basis der vorherrschenden Bedingungen, welche mit Hilfe von Elementen aus anderen logischen Datenströmen bestimmt werden können, mit Qualitätsinformationen anreichern. Auf diese Weise können die zu Beginn definierten Qualitätsindikatoren nun bestimmt werden. Allerdings fehlt noch die Integration der direkten Qualitätsbestimmung um auf diese Weise auch die Qualitätsdimension der Genauigkeit zu ermitteln.

4.6.3 Logische Integration der direkten Qualitätsbestimmung

Um die Genauigkeit der Sensorwerte innerhalb eines Datenstrommanagementsystems zu bestimmen benötigt es einen zusätzlichen Operator, welcher auf Basis der Elemente in einem Strom das zugrunde liegende stochastische Modell ermitteln kann. Diese sollen hier zunächst auf logischer Ebene definiert werden und später auf physischer Ebene implementiert werden. Vernachlässigt man für einen Moment den konkreten Algorithmus zur Annäherung eines stochastischen Modells an eine gegebene Menge von Daten, lässt sich die Annäherung als eine Abbildungsfunktion f_{fit} wie folgt definieren.

Definition 15 (Annäherungsalgorithmus). Die Annäherungsfunktion $f_{fit} : \{\Omega_{\mathcal{T}} \times \mathbb{N}\} \rightarrow \mathbb{G}$ ist eine Abbildung, die ein unbekanntes stochastisches Modell $g \in \mathbb{G}$ auf Basis der Nutzdatentupelmenge $\{\Omega_{\mathcal{T}} \times \mathbb{N}\}$ mit Schema $\mathcal{T}$ bestimmt.

$$f_{fit} : \{(e, n) | e \in \Omega_{\mathcal{T}} \wedge n \in N_{>0}\} \rightarrow g \in \mathbb{G} \tag{4.23}$$

Zur Auswertung des Annäherungsalgorithmus muss dieser in einem logischen Annäherungsoperator genutzt werden. Der Annäherungsoperator bildet dabei die aktuelle Menge von Tupeln aus dem logischen Datenstrom mit Hilfe des Annäherungsalgorithmus auf einen neuen logischen Datenstrom ab. Der logische Annäherungsoperator arbeitet also auf einer Menge von Nutzdatentupel um das stochastische Modell der Daten zu bestimmen unter der Annahme, dass die Nutzdatentupel innerhalb der Menge den gleichen wahren Wert repräsentieren sollen.

Definition 16 (Logischer Annäherungsoperator). Der Annäherungsoperator $\alpha_{f_{fit}} : \mathbb{S}^l_{\mathcal{T}} \times \mathbb{G} \rightarrow \mathbb{S}^l_{\tilde{\mathcal{T}}}$ ist eine Aggregation, die einen logischen Datenstrom mit Schema $\mathcal{T}$ durch Nutzung der Annäherungsfunktion, welche ein stochastische Modell bestimmt, in einen logischen Datenstrom mit Schema $\tilde{\mathcal{T}}$ überführt. Sei $S \in \mathbb{S}^l_{\mathcal{T}}$ dann gilt:

$$\alpha_{f_{fit}}(S) := \{(g, t, 1) | \exists X \subseteq S : X \neq \emptyset \wedge X = \{(e, n) | (e, t, n) \in S\} \wedge g = f_{fit}(X)\} \tag{4.24}$$

Die Menge der Tupel kann dabei entweder durch einen Fensteroperator, wie etwa ein gleitendes Zeitfenster, bestimmt werden oder auf Basis eines Prädikats, welches eine Annäherung nur auf Nutzdatentupel durchführt, die dem Prädikat genügen. Diese Überlegung ist vor allem vor dem Hintergrund interessant, wenn die Nutzdatentupel nur unter bestimmten Bedingungen den gleichen wahren Wert repräsentieren. Als Beispiel sei angenommen, man möchte die Verteilung der Messungen eines Positionssensors eines Fahrzeugs annähern. Diese Annäherung macht dann insofern nur Sinn, wenn diese während Phasen des Stillstands des Fahrzeugs durchgeführt werden, da die Positionsmessungen sonst unterschiedliche wahre Werte repräsentieren würden.

4.6.4 Physische Integration

Zur Realisierung der Annäherung eines stochastischen Modells mit Hilfe des Erwartungswertmaximierungsverfahrens sind in Listing 4.2 die notwendigen Schritte wiedergegeben um auf Basis eines physischen Datenstroms das stochastischen Modell zu bestimmen. Hierbei stellt q eine Prioritätswarteschlange mit der Ordnungsrelation $_{t_S}$ dar, in der die

Algorithmus 4.2 : Integration des Erwartungswertmaximierungsverfahrens

Input : Physischer Eingangsdatenstrom S_{in}
Input : Log-Likelihood Schwellwert δ
Input : Maximale Anzahl an Iterationen I
Input : Prioritätswarteschlange q mit Ordnungsrelation $\leq_{t_S}$
Output : Physischer Ausgangsdatenstrom S_{out}

```
1  S_out ← ∅;
2  for s := (e, [t_S, t_E)) ← S_in do
3      iter ← iterator(q);
4      while iter.hasNext? do
5          i := (e, [t_S, t_E)) ← iter.next;
6          if i.t_E < s.t_S then
7              iter.remove;
8          end
9      end
10     q.insert(s);
11     // Erstelle initiales Modell;
12     (w^(t), μ^(t), Σ^(t)) ← init(q);
13     for I do
14         // Erwartungswert-Schritt;
15         L^(t+1) ← E(q);
16         // Log-Likelihood Test;
17         if |L^(t+1) − L^(t)| < δ then
18             break;
19         end
20         // Maximierung-Schritt;
21         (w^(t+1), μ^(t+1), Σ^(t+1)) ← M(w^(t), μ^(t), Σ^(t));
22     end
23     ((w^(t+1), μ^(t+1), Σ^(t+1)), [t_S, t_E)) → S_out;
24 end
```

Sensormesswerte temporär gehalten werden. Dazu werden bei jedem Stromelement zunächst alle Elemente aus der Prioritätswarteschlange entfernt, die nicht mehr, aufgrund des Startzeitstempels des neuen Elements, als gültig betrachtet werden können. Also ei-

nen Endzeitstempel besitzen, der kleiner ist als der Startzeitstempel des neuen Elements. Anschließend wird das neue Element in die Prioritätswarteschlange eingefügt und initial auf Basis der Daten in der Warteschlange ein stochastisches Modell erstellt. In Folge dessen, wird der Erwartungswert-Schritt auf den Daten in der Warteschlange ausgeführt und die Differenz zwischen der früheren Log-Likelihood und der aktuellen Log-Likelihood mit dem Schwellwert δ verglichen. Ist die Differenz noch zu hoch werden im Maximierungs-Schritt die neuen Parameter für die Gauß-Mischverteilung bestimmt. Wird eine der beiden Abbruchbedingungen, maximale Iterationen oder Log-Likelihood Schwellwert, erreicht, werden die zuletzt bestimmten Parameter als Ausgabe in den physischen Ausgangsdatenstrom S_{out} geschrieben.

Das Kerndichteschätzverfahren benötigt ebenfalls in der Implementierung eine Prioritätswarteschlange zur Verwaltung der gültigen Tupel. Diese sind notwendig um beim Eintreffen eines Nutzdatentupels alle Komponenten der Mischverteilung zu erzeugen. Des Weiteren verwaltet das Kerndichteschätzverfahren drei Parameter (Σ^A, Σ^B, $\Sigma^\times$) zur kontinuierlichen Bestimmung der Varianz der gültigen Daten. Diese Parameter werden im unteren Teil des Algorithmus verwendet um die aktuelle Kovarianzmatrix zu bestimmen und mit dem Bandbreitefaktor zu multipliziert, umso die Bandbreite für die resultierende Mischtypverteilung zu ermitteln. Als Bandbreitefaktor $f_{Bandbreite}$ kann dabei einer der genannten Regeln zur Bestimmung der Bandbreite verwendet werden.

4.6.5 Bregman-Hard Clustering

Das Bregman-Hard Clustering kann als zustandsloser Operator realisiert werden, da sowohl die Bestimmung der Bregman Divergenz, wie auch die einzelnen Schritte des k-Means Verfahrens unabhängig von vorherigen oder darauf folgenden Tupeln geschieht. Dabei geht die Verarbeitung wie folgt vor: Zunächst werden in den ersten Zeilen in Listing 4.4 initiale Cluster gebildet. Die Anzahl dieser Cluster ist abhängig von der Konfiguration des Operators in der jeweiligen Anfrage. Anschließend werden die Komponenten der, im eingehenden Tupel befindlichen, Mischverteilung den Clustern zugewiesen. Aus dieser Menge von Komponenten werden daraufhin die neuen Clusterzentren bestimmt. Diese zwei Schritte, die Zuweisung von Komponenten und die erneute Bestimmung von Clusterzentren, werden solange wiederholt bis entweder keine Neupartitionierung der Komponenten stattfindet oder die Anzahl an maximalen Iterationen erreicht ist. Anschließend werden die ermittelten Clusterzentren in den physischen Ausgangsdatenstrom S_{out} geschrieben.

Algorithmus 4.3 : Integration des Kerndichteschätzverfahrens

Input : Physischer Eingangsdatenstrom S_{in}
Input : Prioritätswarteschlange q mit Ordnungsrelation $_{t_S}$
Input : Kovarianzparameter Σ^A, Σ^B, $\Sigma^\times$
Input : Bandbreitefaktor $f_{Bandbreite}$
Output : Physischer Ausgangsdatenstrom S_{out}

$S_{out} \leftarrow \emptyset$;
for $s := (e, [t_S, t_E)) \leftarrow S_{in}$ **do**
  $iter \leftarrow iterator(q)$;
  while $iter.hasNext?$ **do**
    $u := (e, [t_S, t_E)) \leftarrow iter.next$;
    if $u.t_E < s.t_S$ **then**
      // Entfernen der ungültigen Werte aus der Statistik
      $update(\Sigma^A,\ \Sigma^B,\ \Sigma^\times,\ u)$;
      $iter.remove$;
    end
  end
  $q.insert(s)$;
  // Aktualisierung der Statistik mit dem neuen gültigen Messwert
  $update(\Sigma^A,\ \Sigma^B,\ \Sigma^\times,\ s)$;
  $f_{Scott} \leftarrow |q|^{\frac{-1}{|s.e|+4.0}}$;
  for $i = 0; i < |s.e|; i++$ **do**
    for $j = i; j < |s.e|; j++$ **do**
      $\Sigma_{ij} \leftarrow \Sigma_{ji} \leftarrow \frac{\Sigma^\times_{ij} - \frac{\Sigma^A_{ij}\Sigma^B_{ij}}{|q|}}{|q|-1} f^2_{Bandbreite}$;
    end
  end
  $iter \leftarrow iterator(q)$;
  M $\leftarrow \emptyset$;
  while $iter.hasNext?$ **do**
    $u := (e, [t_S, t_E)) \leftarrow iter.next$;
    $M.add(\omega \leftarrow \frac{1}{|q|}, \mu \leftarrow u.e, \Sigma)$;
  end
  $(M, [t_S, t_E)) \rightarrow S_{out}$;
end

Algorithmus 4.4 : Integration des Bregman-Hard Clustering

Input : Physischer Eingangsdatenstrom S_{in}
Input : Anzahl von Komponenten δ
Input : Maximale Anzahl an Iterationen I
Output : Physischer Ausgangsdatenstrom S_{out}

$S_{out} \leftarrow \emptyset$;
for $s := (M, [t_S, t_E)) \leftarrow S_{in}$ **do**
  // Umwandlung in natürliche Parameter
  $\hat{M} = toNatural(M)$;
  // Wähle initiale Clusterzentren
  $\hat{C} = initialize(\hat{M}, \delta)$;
  for I **do**
    $tmp = table$;
    // Weise die Komponenten den einzelnen Clusterzentren zu
    $table = partition(\hat{C}, \hat{M})$;
    $i = 0$;
    for $m := (\omega, \mu, \Sigma) \leftarrow \hat{M}$ **do**
      $c = -1,\ j = 0,\ min = MAX$;
      for $c := (\omega, \mu, \Sigma) \leftarrow \hat{C}$ **do**
        $distance = BregmanDivergence(m, c)$;
        if $distance < min$ **then**
          $min = distance$;
          $c = j$;
        end
        $j++$;
      end
      $i++$;
      $table[i] = c$;
    end
    $\hat{C} = cluster(\hat{C}, \hat{M}, table)$;
    if $tmp == table$ **then**
      break
    end
  end
  // Umwandlung in Quellen-Parameter
  $C = toSource(\hat{C})$;
  $(C, [t_S, t_E)) \rightarrow S_{out}$;
end

4.7 Zusammenfassung

In diesem Kapitel wurden zunächst die, in der Literatur häufig verwendeten Qualitätsdimensionen und Qualitätsklassen, sowie ihre Definition aufgezeigt und verglichen. Anschließend wurde gezeigt, wie auf Basis von Zusatzinformationen und den Informationen in einem Datenstrom diese Qualitätsinformationen ermittelt werden können. Um diese Zusatzinformationen zu verwalten und, je nach aktueller Bedingung unter denen ein Sensor arbeitet, herzuleiten wurde die anwendergestützte und systemgestützte Bestimmung von Qualitäten aufgezeigt. Während eine anwendergestützte Bestimmung bei kleinen Anwendungen ausreicht, ist bei größeren Anwendungen mit vielen und vor allem unterschiedlichen Sensoren eine systemgestützte Bestimmung notwendig.

Bei verwandten Arbeit zeigte sich, dass die Modellierung von Qualitäten durch die Nutzung einer Ontologie die notwendigen Anforderungen, wie Erweiterbarkeit und Anfragemöglichkeit, erfüllen. Jedoch waren die verwandten Arbeiten in der Modellierung und Verarbeitung von Qualitäten unter Verwendung einer Ontologie auf einen einzigen Datenstrom eines Sensors beschränkt, während in diesem Ansatz gerade die Verknüpfung von Informationen aus unterschiedlichen Quellen eine wichtige Rolle spielt. Eine Alternative zeigte sich in der SSN-Ontologie als mögliche Ontologie, da diese bereits auf Basis anderer existierender Alternativen aufgebaut wurde und durch das W3C vorangetrieben wird, folgte auch, dass die SSN-Ontologie auch in Zukunft eine hohe Verbreitung haben wird. Die SSN-Ontologie wurde aus diesem Grund genutzt um Sensoren und Umgebung zu modellieren, sowie Eigenschaften innerhalb dieser Umgebungen, die von diesen Sensoren überwacht werden, zu verknüpfen. Außerdem dient die Ontologie dazu die Messmöglichkeiten, ihre Qualitäten und die Bedingungen, unter welchen diese Qualitäten gültig sind, von Sensoren zu definieren.

Da allerdings die reine Modellierung von Beziehungen und Qualitäten in einer Ontologie nicht ausreicht um die aktuelle Genauigkeit einer Sensorwahrnehmung zu beschreiben, wurde zudem aufgezeigt, wie direkt auf Basis der Messwerte deren Qualität bestimmt werden kann. Hierzu wurde unter anderem das Erwartungswertmaximierungsverfahren und das Kerndichteschätzverfahren vorgestellt. Beide Verfahren bieten die Möglichkeit, ein stochastisches Modell in Form von Mischverteilungen an die aktuellen Daten anzunähern und so die Messwerte in Form des statistischen Fehlers zu beschreiben. Um nun auf Basis dieser Technologie einen Datenstrom mit den Informationen aus der Ontologie anzureichern wurde ein spezieller logischer Qualitätsoperator definiert, der einen logischen Datenstrom durch Qualitätsinformationen über die enthaltenen Elemente erweitert. In Folge dessen, wurde zunächst gezeigt, wie die aktuell herrschenden Bedingungen für eine Messmöglichkeit eines Sensors bestimmt werden können. Da diese Bedingungen jedoch von aktuellen Werten aus unterschiedlichen Quellen abhängen können, wurde anschließend aufgezeigt, wie diese Quellen auf Basis der Ontologie ausgewählt und integriert werden können. Anschließend wurde gezeigt, wie dieser logische Qualitätsoperator auf eine Kombination aus Basisoperatoren der temporal relationalen Operatoralgebra abgebildet werden kann, um

auf diese Weise die Vorteile der Optimierung bei der Anwendung von Optimierungsregeln der relationalen Algebra zu nutzen. Durch die Reduktion auf vorhandene Operatoren kann zudem die Ontologie als Wissensbasis in beliebigen relationalen Datenstrommanagementsystemen verwendet werden.

Zur Integration der direkten Qualitätsbestimmung wurde ein logischer Operator entwickelt, der auf Basis der Daten in einem logischen Datenstrom das stochastische Modell ermittelt. Hierzu wurde zunächst gezeigt, wie eine solche Bestimmung auf logischer Ebene in ein DSMS integriert werden kann und wie eine mögliche physische Integration durch Nutzung des Erwartungswertmaximierungsverfahrens und der Kerndichteschätzung entwickelt werden kann. Für die Kerndichteschätzung wurde zudem ein Verfahren integriert um die Komplexität von ermittelten stochastischen Modell zu verringern. Auf Basis der hier vorgestellten Integration von Zusatzinformationen in Form einer Ontologie oder durch die direkte Bestimmung von Qualitäten kann nun beispielsweise das Kalman Filter [Kal60] dazu verwendet werden, um mit den hinterlegten Genauigkeitsinformationen unter der Annahmen, dass das dahinter liegende Modell einer Normalverteilung folgt, den wahren Zustand einer überwachten Eigenschaft optimal zu schätzen.

5 Qualitätssensitive Datenstromverarbeitung

Im letzten Kapitel wurden die verschiedenen Qualitätsdimensionen eingeführt und veranschaulicht, wie diese Dimensionen mit Hilfe einer Ontologie beschrieben und mit den Sensoren und Messmöglichkeiten verknüpft werden können. Anschließend wurde erläutert, wie die Qualitätsinformationen auf Basis aktueller Messungen von Sensoren an die Elemente eines Datenstroms angeheftet werden können. Nun, da diese Informationen innerhalb der Verarbeitung eines Datenstrommanagementsystems vorhanden sind, müssen sie entsprechend auch bei der Verarbeitung durch die temporal relationalen Operatoren berücksichtigt werden.

5.1 Einführung

Damit die Qualitätsinformationen innerhalb eines Datenstrommanagementsystems bei der Verarbeitung durch die temporal relationalen Operatoren berücksichtigt werden, müssen die zugrunde liegenden Konzepte eines solchen Systems an die Bedürfnisse der qualitätssensitiven Verarbeitung angepasst werden. Hierzu muss

1. die Semantik der Qualitätsinformationen innerhalb der Verarbeitung definiert werden,
2. ihre aktuellen Werte innerhalb der Verarbeitung adäquat repräsentiert werden und
3. bei der Weiterreichung der Ergebnisse an eine Anwendung in eine lesbare Form serialisiert werden.

Dieses wird im Folgenden dadurch sichergestellt, dass zunächst ein einheitliches Modell definiert wird, welches die zwei bekanntesten Wege zur Repräsentation von Qualitäten integriert, nämlich die der „möglichen Welten“, welche mehrere Welten mit unterschiedlichen Auftrittswahrscheinlichkeiten darstellt, und die der kontinuierlichen Wahrscheinlichkeitsverteilungen. Ein weiteres Ziel ist zudem, die bisherige Verarbeitung in einem DSMS nicht zu beeinträchtigen oder zu verändern um auch weiterhin die bereits vorhandene Funktionalität und die Vorteile einer temporal relationalen Verarbeitung nutzen zu können und den Ansatz leicht in bestehende System zu integrieren.

5.2 Verwandte Arbeiten

Zunächst werden hierzu existierende Arbeit untersucht und mögliche existierende Ansätze aufgezeigt. Unsicherheiten werden zunehmend im Rahmen von Data-Mining-Anwendungen und Sensornetzwerken betrachtet. Zur Speicherung von Unsicherheiten in Daten wird etwa bei MayBMS [Koc09] eine probabilistische Datenbank als eine endliche Menge von Datenbankinstanzen mit gleichem Schema (als mögliche Welt bezeichnet) dargestellt,

wobei jede Welt gewichtet ist mit einem Gewicht zwischen 0 und 1 und die Summe aller Gewichte gleich 1 ist. Das System kann dabei sowohl Unsicherheiten auf Tupelebene, wie auch Unsicherheiten auf Attributebene darstellen. Im Kontext von Datenstrommanagementsystemen wurde diese Form der Unsicherheit unter anderem von Jayram et al. [JM07] für die Bestimmung von statistischen Aggregaten auf Datenströmen aufgegriffen. Die Autoren definieren hierfür Abschätzungen für die Aggregate SUM, AVG, MIN und MAX. In PODS [TPL+10] und Claro [TPD+12] wurde ein Mischtyp-Modell vorgestellt, welches es erlaubt Unsicherheiten über die Existenz von Tupeln und Attributen auszudrücken und in den bekannten relationalen Operatoren zu verarbeiten. Aber nicht nur die Messdaten von Sensoren können Unsicherheiten unterliegen, auch der Zeitpunkt einer Messung kann ungenau sein. Die Verarbeitung der Unsicherheit über den Zeitpunkt von Ereignissen wurde unter anderem in [ZDI12] behandelt. In einer Studie von Wang et al. [WLLW13] wird zwar eine breite Übersicht über den aktuellen Stand mit Fokus auf Unsicherheitstypen, Anfragen mit Unsicherheiten und Modelle zur Repräsentation von Unsicherheiten gegeben. Jedoch ist zu beachten, dass es zum Zeitpunkt dieser Arbeit kein vollständiges System auf Basis des relationalen Modells gibt, welches in der Lage ist Unsicherheiten in einem Sensordatenstrom zu bestimmen und zu verarbeiten, so wie es in den Anwendungsszenarien erforderlich ist.

5.3 Probabilistische Datenstromverarbeitung

Zur Repräsentation von Unsicherheiten in einem DSMS wird im Folgenden das in den Grundlagen beschriebene und bisher genutzte Datenmodell von [Krä07] um das Mischtyp-Modell von [TPD+12] erweitert. Dieses Modell hat den Vorteil, dass es sowohl die Unsicherheit über die Existenz einzelner Attribute, sowie auch die Unsicherheit über die Existenz ganzer Tupel repräsentieren kann und somit als Kandidat für die Darstellung der zuvor beschriebenen Qualitäten geeignet ist.

In dem Mischtyp-Modell werden Tupel mit kontinuierlichen unsicheren Attributen $\mathbf{A}^x$, diskreten unsicheren Attributen $\mathbf{A}^y$, sowie deterministischen Attributen $\mathbf{A}^d$ durch eine Mischtyp-Verteilung g repräsentiert. Die Verteilung g ist dabei ein Paar (p, f). Hierbei repräsentiert $p \in [0, 1]$ die Tupelexistenzwahrscheinlichkeit und f die gemeinsame Dichtefunktion für alle unsicheren Attribute und ist definiert als $f(\mathbf{x}, \mathbf{y}) = f_{\mathbf{A}^x|\mathbf{A}^y}(\mathbf{x}|\mathbf{y})\mathbb{P}[\mathbf{A}^y = \mathbf{y}]$. Die Verteilung g charakterisiert damit nach den Autoren einen Zufallsvektor $(\mathbf{X}, \mathbf{Y}, \mathbf{Z})$ über $(\mathbb{R}^m \times \mathbb{U}^n \times \mathbf{A}^d) \cup \{\perp\}$ mit m kontinuierlichen Attributen, n diskreten Attributen und d deterministischen Attributen. Weiterhin gilt:

$$\begin{aligned} \mathbb{P}[(\mathbf{X}, \mathbf{Y}, \mathbf{Z}) = \perp] &= (1 - p), \\ \mathbb{P}[\mathbf{X} \subseteq I, \mathbf{Y} = \mathbf{y}, \mathbf{Z} = \mathbf{A}^d] &= p \int_I f(\mathbf{x}, \mathbf{y})\, \mathrm{d}\mathbf{x}, \end{aligned} \tag{5.1}$$

für $I \subseteq \mathbb{R}^m, \mathbf{y} \in \mathbb{U}^n$. $\{\perp\}$ stellt den nichtexistenten Fall dar.

Der Wahrscheinlichkeitsraum einer Zufallsvariablen $\mathbf{X}$ wird von den Autoren wie folgt definiert:

Definition 17 (Wahrscheinlichkeitsraum einer Mischtyp-Verteilung). Ein Wahrscheinlichkeitsraum einer Zufallsvariablen $\mathbf{X}$ ist ein Tripel $(S_\mathbf{X}, \mathcal{F}_\mathbf{X}, P_\mathbf{X})$, wobei $S_X = \mathbb{R}^m \cup \{\perp\}$ den Stichprobenraum mit allen möglichen Werten von $\mathbf{X}$ darstellt. Hierbei sei wieder $\perp$ der nichtexistente Fall. Des Weiteren sei $\mathcal{F}_\mathbf{X}$ der Sigmakörper über $S_\mathbf{X}$ und $P_\mathbf{X}$ das Wahrscheinlichkeitsmaß, so dass für jede Menge A in dem Sigmakörper $\mathcal{F}_\mathbf{X}$ gilt, $P_\mathbf{X}(A) = (1-p)1(\perp \in A) + p \int_{A\setminus\{\perp\}} f(\mathbf{x})\, \mathrm{d}\mathbf{x}$.

In Anlehnung an einen logischen Datenstrom definieren wir nun auf Basis dieses Mischtyp-Modells einen logischen probabilistischen Datenstrom wie folgt:

Definition 18 (Logischer probabilistischer Datenstrom). Ein logischer probabilistischer Datenstrom $\tilde{S}^l$ mit Schema $\mathcal{T}$ ist eine potenziell unendliche Multimenge von Elementen (g, t, n), für die gilt

$$\begin{aligned}\tilde{S}^l := \{(g,t,n) | t \in T \wedge g = (p, f_\mathbf{X}) \wedge p \in [0,1] \\ \wedge f(\mathbf{x},\mathbf{y}) = f_{\mathbf{A}^x|\mathbf{A}^y}(\mathbf{x}|\mathbf{y})\mathbb{P}[\mathbf{A}^y = \mathbf{y}] \wedge n \in \mathbb{N}_{>0}\}\end{aligned} \tag{5.2}$$

Weiterhin gilt,

$$\forall (g,t,n), (\hat{g},\hat{t},\hat{n}) \in \tilde{S}^l : (g = \hat{g} \wedge t = \hat{t}) \implies (n = \hat{n}) \tag{5.3}$$

Ähnlich der Bedingung für einen deterministischen logischen Datenstrom in dem bisher genutzten Datenmodell verhindert diese Bedingung, dass zwei Elemente mit gleicher Mischtyp-Verteilung und gleichem Zeitstempel existieren. Sei weiterhin $\tilde{\mathbb{S}}^l$ die Menge aller logischen probabilistischen Datenströme und $\tilde{\mathbb{S}}^l_\mathcal{T} \subset \tilde{\mathbb{S}}^l$ die Menge aller logischen probabilistischen Datenströme mit Schema $\mathcal{T}$. Dann kann ein physischer probabilistischer Datenstrom wie folgt definiert werden:

Definition 19 (Physischer probabilistischer Datenstrom). Sei wieder $\mathbb{I} := \{[t_S, t_E) \in T \times T | t_S < t_E\}$ die Menge aller Zeitintervalle. Ein probabilistischer physischer Datenstrom ist ein Paar $\tilde{S}^p = (M, \leq_{t_S,t_E})$. M ist dabei eine potenziell unendliche Sequenz von Tupeln $(g, [t_S, t_E))$ mit $[t_S, t_E) \in \mathbb{I}$ und g eine Mischtyp-Verteilung (p, f) die einen Zufallsvektor $\mathbf{X}$ beschreibt mit Existenzwahrscheinlichkeit $p \in [0,1]$ und der Dichtefunktion f und dem Wahrscheinlichkeitsraum $(S_\mathbf{X}, \mathcal{F}_\mathbf{X}, P_\mathbf{X})$.

$$\tilde{S}^p := \{(g, [t_S, t_E)) | g \in \Omega_\mathcal{T} \wedge g = (p, f) \wedge [t_S, t_E) \in \mathbb{I}\} \tag{5.4}$$

Weiterhin besitzen alle Elemente aus M das gleiche Schema $\mathcal{T}$ und sind lexikalisch geordnet durch die Ordnungsrelation $\leq_{t_S,t_E}$ über M, so dass alle Tupel $(g, [t_S, t_E))$ nach ihren Zeitstempeln geordnet sind. Ein physisches probabilistisches Stromelement $(g, [t_S, t_E))$ mit Mischtyp-Verteilung g ist innerhalb des Zeitintervalls $[t_S, t_E)$ gültig und existiert mit einer Wahrscheinlichkeit p. $\tilde{\mathbb{S}}^p$ definiert des Weiteren die Menge aller physischen probabilistischen Datenströme.

5.3.1 Umwandlung zwischen physischen und logischen probabilistischen Strom

Wie bereits bei der Umwandlung eines deterministischen physischen Datenstroms in einen deterministischen logischen Datenstrom, lässt sich dieses auch auf die Umwandlung von einem physischen probabilistischen Datenstrom auf einen logischen probabilistischen Datenstrom anwenden. Sei hierzu $\tilde{\varphi}^{p \to l} := \tilde{S}^p \to \tilde{S}^l$ eine Abbildungsfunktion, die einen physischen probabilistischen Datenstrom auf einen logischen probabilistischen Datenstrom abbildet:

$$\begin{aligned}\varphi^{p \to l}(\tilde{S}^p) :=& \{(g,t,n) \in \Omega_{\mathcal{T}} \times T \times \mathbb{N}_{>0} | \\ & n = |\{(g,[t_S,t_E)) \in \tilde{S}^p | t \in [t_S,t_E)\}| \wedge n \in \mathbb{N}_{>0}\}\end{aligned} \tag{5.5}$$

Diese Umwandlung unterscheidet sich nur gering von der Umwandlung eines deterministischen physischen in einen deterministischen logischen Datenstroms, da die dargestellten Unsicherheiten sich nur auf die Nutzdaten und die Existenz des Tupels in dem Datenstrom beziehen nicht aber auf die Gültigkeit des Tupels in der Zeitdomäne.

5.4 Logische Operatoralgebra

Im Folgenden werden die logischen Operatoren für die Selektion, Abbildung, Projektion, Verbund und Aggregation auf Basis der beiden Ansätze aus [Krä07] und [TPD+12] definiert.

Definition 20 (Selektion (σ)). Der Selektionsoperator $\sigma : \tilde{\mathbb{S}}^l_{\mathcal{T}} \times \mathbb{P}_{\mathcal{T}} \to \tilde{\mathbb{S}}^l_{\mathcal{T}}$ bildet alle Elemente eines logischen probabilistischen Datenstroms auf einen neuen logischen probabilistischen Datenstrom ab, wobei ein Selektionskriterium $p \in \mathbb{P}_{\mathcal{T}}$ den Selektionsbereich I vorgibt. Diese Tatsache unterscheidet den probabilistischen Selektionsoperator von dem bisherigen deterministischen Selektionsoperator, da Tupel aus dem logischen probabilistischen Datenstrom nicht entfernt, sondern auf ein neues Stromelement mit einer veränderten Existenzwahrscheinlichkeit abgebildet werden.

$$\begin{aligned}\sigma_p(S) := \{&(g,t,n) \in S | p(g)X \neq \emptyset \\ &\wedge X = \{(g,t,n) \in S | p(g) = \hat{g}\} \\ &\wedge \hat{n} = \sum_{(g,t,n) \in X} n\}\end{aligned} \tag{5.6}$$

$$P_{\mathbf{X}}(A) = \begin{cases} 1 - p \int\limits_{\mathbb{R}^{|\mathbf{X}|} \cap I} f_{\mathbf{X}}(\mathbf{x})\, \mathrm{d}\mathbf{x} & \text{wenn } A = \{\perp\} \\ p \int\limits_{A \cap I} f_{\mathbf{X}}(\mathbf{x})\, \mathrm{d}\mathbf{x} & \text{wenn } A \subset \mathbb{R}^{|\mathbf{X}|} \end{cases} \tag{5.7}$$

Definition 21 (Abbildung (μ)). Der Abbildungsoperator $\mu_f : \tilde{\mathbb{S}}^l_{\mathcal{T}} \times \mathbb{F}_{Map} \to \tilde{\mathbb{S}}^l_{\hat{\mathcal{T}}}$ bildet den logischen probabilistischen Datenstrom $\tilde{S}^l_{\mathcal{T}}$ mit Schema $\mathcal{T}$ durch Anwendung einer

Abbildungsfunktion f auf den logischen probabilistischen Datenstrom $\tilde{S}^l_{\hat{\mathcal{T}}}$ mit Schema $\hat{\mathcal{T}}$ ab. Innerhalb der Abbildungsfunktion können unsichere Attribute aus dem Eingangsstrom verwendet werden, so dass die resultierende Mischtyp-Verteilung $\hat{g}$ des Ausgabeelements und somit die Existenzwahrscheinlichkeit des Tupels von der ursprünglichen Existenzwahrscheinlichkeit abweicht. Sei hierzu $\mathbb{F}_{map}$ die Menge aller Abbildungsfunktionen, die ein Tupel mit Schema $\mathcal{T}$ auf ein Tupel mit Schema $\hat{\mathcal{T}}$ abbildet und $f \in \mathbb{F}_{map}$.

$$\begin{aligned}\mu_f(S) := \{(\hat{g}, t, \hat{n}) | \exists X \subseteq S : X \neq \emptyset \\ \wedge X = \{(g, t, n) \in S | f(g) = \hat{g}\} \\ \wedge \hat{n} = \sum_{(g,t,n)\in X} n\}\end{aligned} \tag{5.8}$$

Definition 22 (Projektion (π)). Der Projektionsoperator $\pi_f : \tilde{\mathbb{S}}^l_{\mathcal{T}} \rightarrow \tilde{\mathbb{S}}^l_{\hat{\mathcal{T}}}$ bildet den logischen probabilistischen Datenstrom $\tilde{S}^l_{\mathcal{T}}$ mit Schema $\mathcal{T}$ auf den logischen probabilistischen Datenstrom $\tilde{S}^l_{\hat{\mathcal{T}}}$ mit Schema $\hat{\mathcal{T}}$ ab. Dies entspricht dabei der Integration der Mischtyp-Verteilung über die Domäne der herausprojezierten Attribute.

$$\begin{aligned}\pi_f(S) := \{(\hat{g}, t, \hat{n}) | \exists X \subseteq S : X \neq \emptyset \\ \wedge X = \{(g, t, n) \in S\} \\ \wedge \hat{n} = \sum_{(g,t,n)\in X} n\}\end{aligned} \tag{5.9}$$

Sei dazu $(\mathbf{X}, \mathbf{Y})$ ein Zufallsvektor aus der Mischtyp-Verteilung $(p, f_{\mathbf{X},\mathbf{Y}})$ und $\mathbb{R}^{|\mathbf{X}|}$ die Domäne von $\mathbf{X}$ und $\mathbb{R}^{|\mathbf{Y}|}$ die Domäne von $\mathbf{Y}$. Eine Projektion des Zufallsvektors $(\mathbf{X}, \mathbf{Y})$ auf den Zufallsvektor $\mathbf{Y}$ entspricht dann:

$$P_{\mathbf{Y}} = \begin{cases} 1 - p & \text{if } A = \{\perp\} \\ p \int\limits_A \int\limits_{\mathbb{R}^{|X|}} f_{\mathbf{XY}}(\mathbf{x}, \mathbf{y})\, \mathrm{d}\mathbf{x}\mathrm{d}\mathbf{y} & \text{wenn } A \subset \mathbb{R}^{|\mathbf{Y}|} \end{cases} \tag{5.10}$$

Definition 23 (Verbund ($\times$)). Der Verbundoperator $\times : \tilde{\mathbb{S}}^l_{\mathcal{T}_1} \times \tilde{\mathbb{S}}^l_{\mathcal{T}_2} \rightarrow \tilde{\mathbb{S}}^l_{\hat{\mathcal{T}}}$ von zwei Strömen kombiniert alle Elemente aus beiden Strömen mit Schema $\mathcal{T}_1$ und Schema $\mathcal{T}_2$ die zum selben Zeitpunkt gültig sind zu einem neuen logischen probabilistischen Datenstrom mit Schema $\hat{\mathcal{T}}$. Für die Kombination wird die Funktion $\circ : \Omega_{\mathcal{T}_1} \times \Omega_{\mathcal{T}_2} \rightarrow \Omega_{\hat{\mathcal{T}}}$ verwendet, welche die beiden Tupel konkartiniert.

$$\begin{aligned}\times(S_1, S_2) := \{(\circ(g_1, g_2), t, n_1 n_2) | (g_1, t, n_1) \in S_1 \\ \wedge (g_2, t, n_2) \in S_2\}\end{aligned} \tag{5.11}$$

Durch die Konkatenation ändert sich auch die Mischtyp-Verteilung des Ausgabeelements. Hierbei wird angenommen, dass die beiden Zufallsvektor voneinander unabhängig sind.

$$P_{\mathbf{XY}}(A) = \begin{cases} 1 - p_{\mathbf{X}} p_{\mathbf{Y}} & \text{if } A = \{\perp\} \\ p_{\mathbf{X}} p_{\mathbf{Y}} \iint\limits_{A} f_{\mathbf{X}}(\mathbf{x}) f_{\mathbf{Y}}(\mathbf{y})\ \mathbf{dxdy} & \text{wenn } A \subset (\mathbb{R}^{|\mathbf{X}|} \times \mathbb{R}^{|\mathbf{Y}|}) \end{cases} \tag{5.12}$$

Definition 24 (Aggregation (α)). Der Aggregationsoperator $\alpha : \tilde{\mathbb{S}}^l_{\mathcal{T}} \times \mathbb{F}_{agg} \rightarrow \tilde{\mathbb{S}}^l_{\hat{\mathcal{T}}}$ berechnet eine gegebene Aggregationsfunktion f_{agg} über die nicht temporale Multimenge von Elementen aus dem logischen probabilistischen Datenstrom $\tilde{\mathbb{S}}^l_{\mathcal{T}}$ mit Schema $\mathcal{T}$, die zu einem Zeitpunkt gültig sind. Sei hierzu $\mathbb{F}_{agg}$ die Menge aller Aggregationsfunktionen. Eine Aggregationsfunktion $f_{agg} \in \mathbb{F}_{agg}$ mit $f_{agg} : P(\Omega_{\mathcal{T}} \times \mathbb{N}_{>0}) \rightarrow \Omega_{\hat{\mathcal{T}}}$, wobei P die Potenzmenge symbolisiert, berechnet das Aggregate mit Schema $\mathcal{T}$ über eine Menge von Elementen (g, n):

$$\begin{aligned} \alpha_{f_{agg}}(S) := \{(agg, t, 1) | \exists X \subseteq S : X \neq \emptyset \\ \wedge X = \{(g, n) | (g, t, n) \in S\} \\ \wedge agg = f_{agg}(X)\} \end{aligned} \tag{5.13}$$

5.5 Physische Operatoralgebra

Durch die Erweiterung innerhalb der logischen Operatoralgebra ist es notwendig, die definierte Semantik in die physische Operatoralgebra einfließen zu lassen. Zur Repräsentation der gemeinsamen Dichtefunktion f aus der Mischtyp-Verteilung g wird eine multivariate Mischverteilung verwendet. Eine Mischverteilung für einen Wahrscheinlichkeitsvektor X ist eine Kombination aus gewichteten Verteilungen gleicher Art, wobei f_X die Dichtefunktion dargestellt und definiert ist als:

$$f_X(x) = \sum_{i=1}^{m} w_i f_{X_i}(x)$$

wobei gilt $0 \leq w_i \leq 1, \sum_{i=1}^{m} w_i = 1$ und jede Komponente der Mischverteilung X_i ist eine k-variate Verteilung. Somit wird ein Stromelement in einem physischen Datenstrom wie folgt repräsentiert:

$$(e, d, p, [t_S, t_E))$$

wobei e das relational Tupel ist und $[t_S, t_E)$ das halboffene Gültigkeitsintervall, wie davor. Der Parameter d mit:

$$d = \{f_{Xi}(x), n_i, [\underline{s}, \overline{s}]_i, a_i\}_{i=1}^{l}$$

ist die Menge der l multivariaten Mischverteilungen definiert über ein oder mehrere Attribute in dem relational Tupel e. Jede Mischverteilung in d ist dabei definiert in dem Intervall

$[\underline{s}, \overline{s}]$ und wird über den Faktor n normalisiert, so dass ihr Integral im Falle einer kontinuierlichen Mischverteilung wieder 1 ergibt. Im Falle einer diskreten Mischverteilung entspricht die Dichtefunktion der gewichteten Summe der Einzelinstanzen.

$$f_X(x) = \begin{cases} n \sum_{i=1}^{m} w_i f_{X_i}(x) & \text{wenn } x \in [\underline{s}, \overline{s}] \\ 0 & \text{sonst} \end{cases}$$

Der Parameter a stellt eine Referenzliste zu den einzelnen Attributen dar, wobei ihr Index die jeweilige Dimension innerhalb der Mischverteilung repräsentiert. Der Parameter repräsentiert somit zum einen den Wert der Unsicherheit des Attributes und gleichzeitig die Korrelation zwischen mehreren kontinuierlichen oder diskreten Unsicherheitsattributen in einem Stromelement.

Der dritte Parameter p eines Stromelements ist die Existenzwahrscheinlichkeit eines Elements. Während der Verarbeitung wird die Existenzwahrscheinlichkeit durch Filterungen und Verknüpfungen mit anderen Elementen modifiziert. Daher repräsentiert die Existenzwahrscheinlichkeit zu jedem Zeitpunkt immer die Wahrscheinlichkeit mit der das gegebene Stromelement zum aktuellen Zeitpunkt der physischen Verarbeitung existiert. Stromelemente mit einer Existenzwahrscheinlichkeit gleich 0 können aus Gründen der Performanz von jedem Operator innerhalb der Verarbeitung direkt verworfen werden, da sie nichtmehr existent sind.

Ein relationales Tupel kann aus beliebig vielen unsicheren kontinuierlichen, diskreten oder deterministischen Attributes bestehen. Wenn ein Attribut eine kontinuierliche oder diskrete Verteilung aufweist, wird es durch eine Referenz in e zu einer Verteilung in d dargestellt. Ist das Attribut dagegen ein deterministischer Wert wird es durch seinen exakten Wert innerhalb des relationalen Tupels e dargestellt. Hierbei wird zwischen zwei Fällen unterschieden: Entweder ist der Wahrscheinlichkeitsvektor ein diskreter Wahrscheinlichkeitsvektor, dann wird dieser durch eine Menge von diskreten Werten mit jeweiliger Gewichtung repräsentiert, oder der Wahrscheinlichkeitsvektor ist ein kontinuierlicher Wahrscheinlichkeitsvektor, dann wird dieser durch eine Kombination aus m Gauß-Vektoren $X_1, X_2, \ldots, X_m$ dargestellt mit

$$f_{X_i}(x) = \frac{1}{(2\pi)^{k/2}|\Sigma_i|^{1/2}} e^{-\frac{1}{2}(x-\mu_i)^T \Sigma_i^{-1}(x-\mu_i)}$$

Dabei stellt k die Größe des Zufallsvektors dar.

5.5.1 Arithmetische Operatoren

Zusätzlich zu den physischen Operatoren bedarf es zusätzlichen arithmetischen Operatoren zur Verarbeitung der verwendeten kontinuierlichen und diskreten Verteilungen. Dies wird zum einen in Abbildungen benötigt, aber auch innerhalb von Selektionskriterien für Filter- und Verbundoperatoren verwendet. Zu den wichtigsten arithmetischen Operatoren zählen dabei die Addition und Subtraktion, sowie die Multiplikation und Division.

5.5.1.1 Addition und Subtraktion

Die Addition zweier unabhängiger Zufallsvektoren $\mathbf{X}$, $\mathbf{Y}$ welche multivariaten Normalverteilungen folgen ist wieder ein Zufallsvektor der einer multivariaten Normalverteilung folgt:

$$\mathbf{Z} = \mathbf{X} + \mathbf{Y} \sim \mathcal{N}_{\mathbf{X}}(\mu_{\mathbf{X}}, \Sigma_{\mathbf{X}}) + \mathcal{N}_{\mathbf{Y}}(\mu_{\mathbf{Y}}, \Sigma_{\mathbf{Y}}) = \mathcal{N}(\mu_{\mathbf{X}} + \mu_{\mathbf{Y}}, \Sigma_{\mathbf{X}} + \Sigma_{\mathbf{Y}})$$

Das gleiche gilt für die Subtraktion zweier unabhängiger Zufallsvektoren. Im Falle von diskreten Verteilungen entspricht die Addition zweier Zufallsvektoren der paarweisen Addition aller möglichen Welten und der jeweiligen Multiplikation der Wahrscheinlichkeiten.

$$\mathbf{Z} = \mathbf{X} + \mathbf{Y} \sim f_Z(x) = \sum_{x,y;x+y=z} f_X(x) f_Y(x)$$

5.5.1.2 Multiplikation und Division

Das Produkt zweier unabhängiger Zufallsvektoren, welche multivariaten Normalverteilungen folgen, ergibt eine Produktverteilung:

$$\mathbf{Z} = \mathbf{XY} \sim \mathcal{N}_{\mathbf{X}}(\mu_{\mathbf{X}}\Sigma_{\mathbf{X}})\mathcal{N}_{\mathbf{Y}}(\mu_{\mathbf{Y}}, \Sigma_{\mathbf{Y}}) = \int_{-\infty}^{\infty} f_{\mathbf{X}}(\mathbf{x}) f_{\mathbf{Y}}(\frac{\mathbf{z}}{\mathbf{x}})\ \mathrm{d}\mathbf{x}$$

Diese kann allerdings in der Form nicht weiter in dem aktuellen Konzept verwendet werden. Um aber dennoch die Möglichkeit einer Multiplikation innerhalb der Anfrage zu ermöglichen, wird die entstehende Verteilung durch eine neue Mischverteilung angenähert. Das Ergebnis ist also nicht mehr exakt, sondern eine Annäherung an die eigentliche Verteilung. Zur Realisierung wurden bereits zwei Ansätze in Kapitel 4 vorgestellt, das Erwartungswertmaximierungsverfahren und die Kerndichteschätzung in Kombination mit dem Bregman-Hard Clustering. Im Falle der diskreten Wahrscheinlichkeitsverteilung lässt sich die Multiplikation ähnlich der Addition durchführen indem paarweise alle möglichen Welten berechnet werden und die Wahrscheinlichkeiten multipliziert werden.

5.5.2 Selektion

Da nun nicht mehr nur deterministische Werte, sondern auch probabilistische Werte innerhalb der Auswertung eines Selektionskriteriums vorkommen können, ist es notwendig die Wahrscheinlichkeitsmasse innerhalb einer Selektionsoperation, welche durch das Selektionskriterium vorgegeben wird, zu bestimmen. Hierzu wird bei der Evaluierung des Selektionskriteriums zunächst unterschieden, ob es sich bei dem Attribut auf das sich das Selektionskriterium bezieht um ein deterministisches oder ein probabilistisches Attribut handelt.

Im ersten Fall wird das Selektionsattribut wie bisher durch den vorhandenen Selektionsalgorithmus ausgewertet und die Existenzwahrscheinlichkeit je nach Ergebnis auf 0

gesetzt, wenn das Selektionskriterium nicht erfüllt ist, oder auf 1 gesetzt, wenn das Selektionskriterium erfüllt ist.

Algorithmus 5.1 : Probabilistischer Selektionsoperator

Input : Physischer Eingangsdatenstrom S_{in}
Input : Selektionskriterium ρ
Output : Physischer Ausgangsdatenstrom S_{out}

$S_{out} \leftarrow \emptyset$
for $s := (e, d, p, [t_S, t_E)) \leftarrow S_{in}$ **do**
  // Liste der in dem Selektionskriterium referenzierten Attributen
  $a \leftarrow attr(\rho)$
  // Auswahl der Mischverteilungen zu den referenzierten Attributen
  $\hat{f} \leftarrow s.d[\&a]$
  // Bestimmung des Selektionsbereichs
  $[\underline{s}, \overline{s}] = \min \hat{f}.support, \rho$
  // Unterscheidung zwischen diskreten und kontinuierlichen Verteilungen
  if $type(f) = discrete$ **then**
    $q \leftarrow \sum_{x \in [\underline{s}, \overline{s}]} \hat{f}$
  end
  else
    // Unterscheidung zwischen univariater und multivariater Mischverteilung
    if $dim(\hat{f}) = 1$ **then**
      $q \leftarrow \frac{1}{2}[1 + erf\frac{\overline{s} - \mu}{\sqrt{2\sigma^2}}] - \frac{1}{2}[1 + erf\frac{\underline{s} - \mu}{\sqrt{2\sigma^2}}]$
    end
    else
      // Abschätzung des Integrals durch Verwendung des Genz-Algorithmus
      $q \leftarrow Genz(\hat{f}, [\underline{s}, \overline{s}], samples)$
    end
  end
  // Aktualisierung des Normierungsfaktors und des Sigmaraums
  $\hat{f}.scale \leftarrow 1/q$
  $\hat{f}.support \leftarrow [\underline{s}, \overline{s}]$
  $\hat{p} \leftarrow pq$
  // Nichtexistente Tupel können aus Performanzgründen verworfen werden
  if $\hat{p} > 0$ **then**
    // Ausgabe des aktualisierten Elements an den Ausgabestrom
    $(e, \hat{d}, \hat{p}, [t_S, t_E)) \rightarrow S_{out}$
  end
end

Im anderen Fall wird nach Art der Verteilung unterschieden:

1. Handelt es sich bei der Wahrscheinlichkeitsverteilung um eine diskrete Verteilung wird die Wahrscheinlichkeitsmasse als Summe über alle Zustände gebildet, die dem Selektionskriterium genügen und innerhalb des Selektionsbereichs liegen.

2. Handelt es sich bei der Wahrscheinlichkeitsverteilung um eine univariate kontinuierliche Mischverteilung wird das Integral über die Mischverteilung innerhalb der Grenzen, welche durch das Selektionskriteriums und den Sigmakörper der Verteilung vorgegeben werden, gebildet. Zur Bildung des Integrals wird dabei zunächst über jede Komponente der Mischverteilung das Integral gebildet, mit dem jeweiligen Komponentengewicht gewichtet und anschließend addiert.

3. Handelt es sich bei der Wahrscheinlichkeitsverteilung um eine multivariate kontinuierliche Mischverteilung wird mit Hilfe des Algorithmus von Genz [Gen92] die Wahrscheinlichkeit der mehrdimensionalen Fläche, die durch das Selektionskriterium und den bisherigen Sigmakörper definiert wird, approximiert. Als Parameter erhält diese Funktion zusätzlich die Anzahl an Stichproben zur Approximation. Diese Anzahl hat dabei eine Auswirkung auf die Genauigkeit der Approximation und kann als Parameter zur Optimierung der Verarbeitung frei gewählt werden.

In allen drei Fällen ist das Resultat eine veränderte Mischtyp-Verteilung. Daher wird im Anschluss zur Normierung die bestimmte Wahrscheinlichkeitsmasse mit der bisherigen Existenzwahrscheinlichkeit p der Mischtyp-Verteilung des Stromelements multipliziert und der Skalierungsfaktor um die nun fehlende Wahrscheinlichkeitsmasse erhöht, so dass die Fläche unter der gemeinsamen Dichtefunktion wieder 1 ergibt.

5.5.2.1 Projektion

Bei der Projektion werden einzelne Attribute aus dem Ergebniselement entfernt. Da die gemeinsame Dichtefunktion durch den Skalierungsfaktor auf 1 normalisiert ist, müssen diese bei der Projektion nicht weiter betrachtet werden, so dass bei der konkreten Realisierung des physischen Projektionsoperators lediglich die Einträge in der jeweiligen Dimension entfernt werden können. Der folgende Pseudocode in Listing 5.2 verdeutlicht dieses Vorgehen: Im Falle einer diskreten Verteilung wird entsprechend der Stichprobenraum um die herausprojezierte Dimension verkleinert.

Der Vorteil liegt hier dabei, dass lediglich Zeilen und Spalten der herausprojezierten Dimension entfernt werden müssen, jedoch keine weiteren Änderungen an dem Stromelement vorgenommen werden müssen. Somit hängt die Laufzeit des Operators nur von der Anzahl der Komponenten der Mischverteilung ab. Auch die Existenzwahrscheinlichkeit eines Tupels wird durch eine Projektion nicht verändert.

Algorithmus 5.2 : Probabilistischer Projektionsoperator

Input : Physischer Eingangsdatenstrom S_{in}
Input : Projektionsvektor R
Output : Physischer Ausgangsdatenstrom S_{out}

$S_{out} \leftarrow \emptyset$
for $s := (e, d, p, [t_S, t_E)) \leftarrow S_{in}$ **do**
  // Liste der zu referenzierten Attributen
  $a \leftarrow attr(R)$
  // Auswahl der Mischverteilungen zu den referenzierten Attributen
  $\hat{f} \leftarrow s.d[\&a]$
  for $k := (w, \mu, \Sigma)) \leftarrow \hat{f}$ **do**
    $k.\mu \leftarrow R\, diag(k.\mu)\, R^T$
    $k.\Sigma \leftarrow R\, k.\Sigma\, R^T$
  end
  $s \rightarrow S_{out}$
end

5.5.2.2 Abbildung

Durch die Verwendung von probabilistischen Werten innerhalb einer Abbildungsfunktion können zusätzliche Unsicherheiten entstehen und somit die Existenzwahrscheinlichkeit eines Abbildungsergebnisses und des resultierenden Elements verändern. Um diesem Rechnung zu tragen muss innerhalb des Abbildungsoperators die Veränderung der Existenzwahrscheinlichkeit gespeichert werden. Dies geschieht hierbei über die Differenz der Wahrscheinlichkeitsmasse des eingehenden Elements und des ausgehenden Elements. Da das Ergebnis des Integrals bereits in Form der Tuperlexistenzwahrscheinlichkeit und des Skalierungsfaktors der einzelnen Mischverteilungen in der verwendeten Datenstruktur gespeichert sind, genügt es die Differenz zwischen dem eingehenden Tupel und dem ausgehenden Tupel, wie in Listing 5.3 dargestellt, zu vergleichen.

5.5.2.3 Verbund

Bei dem Verbund werden jeweils zwei Elemente, die zum gleichen Zeitpunkt gültig sind miteinander verbunden indem die Nutzdaten der Elemente konkatiniert werden. Dieses Vorgehen ist in Listing 5.4 dargestellt. Bei dieser Verarbeitung lässt sich die Existenzwahrscheinlichkeit des ausgehenden kombinierten Elements als Produkt der eingehenden Existenzwahrscheinlichkeiten der beiden Elemente bestimmen. Wichtig ist hierbei anzumerken, dass bei der Verarbeitung davon ausgegangen wird, dass die beiden Elemente unabhängig voneinander sind.

Algorithmus 5.3 : Probabilistischer Abbildungsoperator

Input : Physischer Eingangsdatenstrom S_{in}
Input : Abbildungsfunktion f_m
Output : Physischer Ausgangsdatenstrom S_{out}

```
1  S_out ← ∅
2  for s := (e, d, p, [t_S, t_E)) ← S_in do
3      // Liste der zu referenzierten Attributen
4      a ← attr(f_m)
5      // Auswahl der Mischverteilungen zu den referenzierten Attributen
6      f̂ ← s.d[&a]
7      // Bestimmung der neuen Existenzwahrscheinlichkeit aus der Differenz der
       beiden Skalierungsfaktoren
8      p̂ ← p(1/f_m(e).scale−1/f̂.scale)
9      // Nichtexistente Tupel können auch hier aus Performanzgründen verworfen
       werden
10     if p̂ > 0 then
11         // Ausgabe des aktualisierten Elements an den Ausgabestrom
12         (f_m(e), p̂, [t_S, t_E)) → S_out
13     end
14 end
```

Die Elemente werden dabei durch eine Prioritätswarteschlange verwaltet. In dieser Warteschlange verweilen die Elemente solange, bis auf Grund der zeitlichen Ordnung beider Ströme kein Verbundpartner mehr gefunden werden kann. Die Warteschlange bietet dabei die beiden Methoden *offer* und *poll*, die jeweils ein Element in die Warteschlange hinzufügen bzw. herausnehmen. Bei der Konkatenierung der Elemente müssen zusätzlich zu den Nutzlasten der beiden Elemente auch die Ebene mit den Verteilungen der beiden Elemente verkettet werden. Dadurch wird es auch notwendig, dass die bisherigen Referenzen der probabilistischen Elemente in der Nutzlast aktualisiert werden, da sich die Indizes durch den Verbund verändert haben. Dies wird über die Methode *updateRef* realisiert.

5.5.2.4 Mengenoperatoren

Zu den Mengenoperatoren zählen die Vereinigung, die Differenz, sowie die Teilmenge. Da die Existenzwahrscheinlichkeit sich nur auf das aktuelle Stromelement bezieht und keine Beziehung zu anderen Elementen im Strom existiert, müssen Mengenoperatoren in diesem Konzept bei der Verarbeitung nicht genauer betrachtet werden. Des Weiteren müsste bei der Differenz und der Teilmenge die Gleichheit zweier Stromelemente betrachtet werden. In den betrachteten Szenarien mit Sensordaten kommt es allerdings nur sehr selten vor, dass zwei Sensormessungen zu unterschiedlichen Zeitpunkten exakt gleiche Werte

aufweisen. Aus diesem Grund werden diese Operatoren in dieser Arbeit nicht weiter betrachtet.

5.6 Zusammenfassung

In diesem Kapitel wurde aufgezeigt, wie die zuvor beschriebenen Qualitäten in einem Datenstrommanagementsystem abgebildet und verarbeitet werden können. Hierzu wurde auf den Arbeiten von Tran et al. [TPD+12] und Krämer [Krä07] aufgebaut und eine Erweiterung der temporalen relationalen Algebra definiert, die es erlaubt Ungenauigkeiten der Werte in Form von Mischtyp-Verteilungen darzustellen und mittels der definierten temporal relationalen Operatoren zu verarbeiten. Hierzu wurde aufgezeigt, wie die Qualitätsinformation innerhalb eines Stromelementes dargestellt und zwischen Operatoren ausgetauscht werden können, sowie die Semantik der logischen Operatoren definiert.

In einem zweiten Schritt wurde gezeigt, wie eine mögliche physische Realisierung dieser logischen Operatoren implementiert werden könnte. Hierbei wurden je nach Art der Verteilung und der Dimension der Verteilung unterschiedliche Alternativen aufgezeigt. So wurde unter anderem gezeigt, wie ein Selektionsoperator sowohl für diskrete, wie auch für kontinuierliche Mischverteilungen realisiert werden kann. Im Fall der kontinuierlichen Mischverteilung wurde weiterhin gezeigt, wie sowohl univariate, wie auch multivariate Mischverteilung innerhalb eines Selektionsoperators verarbeitet werden können. Des Weiteren wurde auch auf die Verarbeitung von Mischverteilungen innerhalb eines Abbildungsoperators eingegangen und die hierfür häufig verwendeten arithmetischen Operationen definiert. Im nächsten Kapitel soll nun gezeigt werden, wie dieses Konzept konkret in ein Datenstrommanagementsystem integriert wurde und in den beschriebenen Anwendungsszenarien genutzt wurde.

Algorithmus 5.4 : Probabilistischer Verbundoperator

Input : Physischer Eingangsdatenstrom S_1 , S_2
Input : Prioritätswarteschlangen q_1, q_2 und Q mit Ordnungsrelation t_s
Output : Physischer Ausgangsdatenstrom S_{out}

```
1  S_out ← ∅
2  j, k ∈ [1, 2]
3  // Einfügen neuer Elemente in die Prioritätswarteschlange und entfernen
   abgelaufener Elemente
4  for s_j := (e, d, p, [t_S, t_E)) ← S_j do
5      k ← j mod 2 + 1
6      q_k.removeAll(≤ s.t_S)
7      q_j.insert(s)
8      iter ← iterator(q_k)
9      while iter.hasNext do
10         i := (e, d, p, [t_S, t_E)) ← iter.next
11         // Konkatenation der Nutzdaten und Multiplikation der
           Existenzwahrscheinlichkeiten
12         if j = 1 then
13             Q.offer((updateRef(s.e ∘ i.e), s.d ∘ i.d, s.p i.p, s.[t_S, t_E) ∩ i.[t_S, t_E)))
14         end
15         else
16             Q.offer((updateRef(i.e ∘ s.e), i.d ∘ s.d, i.p s.p, i.[t_S, t_E) ∩ s.[t_S, t_E)))
17         end
18     end
19     t_{S_j} ← s_j.t_S
20     min ← min t_{S_1} t_{S_2}
21     if min ≠ ⊥ then
22         while !Q.isEmpty do
23             i := (e, d, p, [t_S, t_E)) ← Q.peek
24             if i.t_S ≤ min then
25                 Q.poll → S_out
26             end
27             else
28                 break
29             end
30         end
31     end
32 end
33 while !Q.isEmpty do
34     Q.poll → S_out
35 end
```

6 Implementierung

Als Beweis für die Machbarkeit des beschriebenen Ansatzes zur qualitätssensitiven Verarbeitung und Erstellung von dynamischen Kontextmodellen wird in diesem Kapitel die Integration des Verfahrens in ein bestehendes Datenstrommanagementsystem vorgestellt. Hierzu wird zunächst in Abschnitt 6.1 das hierfür verwendete Datenstrommanagementsystem Odysseus [AGG+12] erläutert. Aufbauend darauf wird in Abschnitt 6.2 die Integration des Ansatzes in das vorhandene System beschrieben. In Abschnitt 6.3 werden abschließend die in dieser Arbeit betrachteten Anwendungsszenarien erläutert und aufgezeigt, wie diese mit der beschriebenen Implementierung realisiert werden konnten.

6.1 Das Datenstrommanagementsystem Odysseus

Das Datenstrommanagementsystem Odysseus [AGG+12] wurde an der Carl von Ossietzky Universität Oldenburg in der Abteilung Informationssysteme entwickelt und bereits erfolgreich in einer Vielzahl von Anwendungsszenarien eingesetzt. In [GBG+12] wurde das System verwendet um die Überwachung pflegebedürftiger Personen in einem Smart Home zu ermöglichen. Des Weiteren wurde Odysseus in [Bol11] zur Objektverfolgung und Prädiktion von Verkehrsteilnehmern im Straßenverkehr verwendet. In [JBG+10] wurde das System genutzt um Information bei der dezentralen Energiegewinnung kontinuierliche zu verarbeiten und dabei kritische und sicherheitsrelevante Informationen bevorzug zu behandeln. Eine weitere Arbeit beschäftigt sich mit der Verwendung zur Leistungskennlinienberechnung von Windenergieanlagen [GTN11].

6.1.1 Architektur

Die Architektur des Datenstrommanagementsystems Odysseus ist in drei Schichten unterteilt. Die Anwendungsschicht im oberen Teil, die Datenverarbeitungsschicht in der Mitte und die Sensorschicht im unteren Teil. In der Anwendungsschicht können Anwendungen Verarbeitungsanfragen an das System richten und kontinuierliche Verarbeitungsergebnisse als Datenstrom empfangen. Hierzu bietet das System sogenannte Transport- und Protokollhandler an, welche ein Ergebnistupel in ein anwendungsspezifisches Format überführen und an die Anwendung übermitteln. In der Sensorschicht können Sensoren im Gegenzug kontinuierlich ihre Messungen in Form eines Datenstroms an das System übermitteln. Auch hierzu bietet das System Transport- und Protokollhandler an, die ein anwendungsspezifisches Datenformat in die interne Repräsentation umwandeln und auch die Möglichkeit bieten Sensoren zu konfigurieren. In der Datenverarbeitungsschicht werden die einzelnen, für die Verarbeitung notwendigen, Verarbeitungsoperatoren bereitgestellt und können über eine Anfrage in einer vom System unterstützten Anfragesprache durch das *Query Interface* in der Anwendungsschicht genutzt werden. Durch diese Schnittstelle kön-

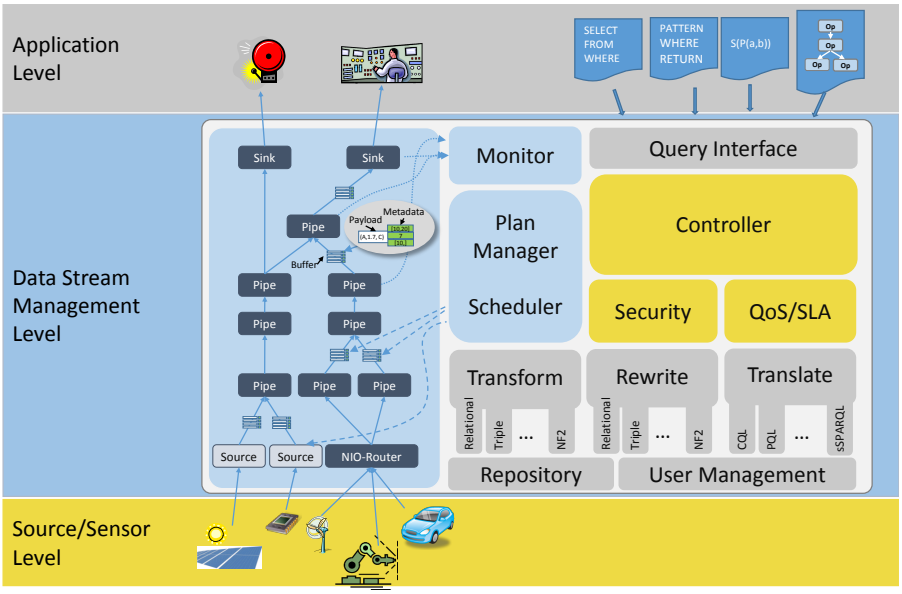


Abbildung 6.1: *Architektur des Datenstrommanagementsystems Odysseus nach [JG08]*

nen Anfragen jederzeit gestartet, pausiert und gestoppt werden oder durch neue Anfragen ersetzt werden. Die komplette Architektur des Odysseus-Systems in ist Abbildung 6.1 dargestellt.

Odysseus ist komponentenbasiert und baut auf dem Open Services Gateway Specification (OSGi)-Framwork Equinox auf. Hierbei wird stark auf die Verwendung von deklarativen Diensten innerhalb des OSGi-Frameworks gesetzt, um eine lose Kopplung einzelner Komponenten (Bundles) zu erreichen. Dies schafft die Möglichkeit einzelne Komponenten zu erweitern oder auszutauschen ohne dabei die aktuelle Verarbeitung stoppen zu müssen. Auch können so neue Datentypen, Operatoren, Transformationsregeln und sogar GUI-Elemente ohne Änderung der Kernfunktionalitäten integriert und genutzt werden. Hierdurch ist es unter anderem auch möglich, verschiedene Ansätze zur Verarbeitung von Datenströmen zu integrieren ohne dabei Änderungen an bestehenden Komponenten durchführen zu müssen. Dies gilt zum einen für Komponenten, die für die direkte Verarbeitung der Daten zuständig sind, aber auch für GUI-Elemente, die zur Visualisierung von Verarbeitungsergebnissen dienen und zur Formulierung von Anfragen bereit stehen. Des Weiteren wird innerhalb der Architektur zwischen einem Server- und einem Clientteil unterschieden, so dass ein Teil der Komponenten, die strikt für die Verarbeitung zuständig sind, getrennt von einer für den Anwender konzipierten graphische Oberfläche betrieben werden können.

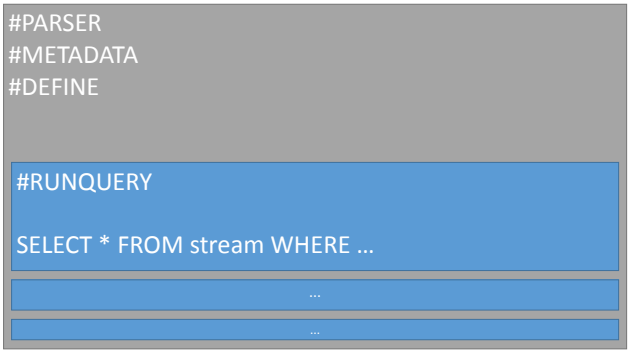


Abbildung 6.2: *Aufbau einer Anfrage in Odysseus Script*

Die offene und erweiterbare Architektur von Odysseus erlaubt es so, auf einfache Weise das System durch die Implementierung von zusätzlichen OSGi-Komponenten um neue Verarbeitungsoperatoren und neue Datenmodelle zu erweitern und so den hier beschriebenen Ansatz zur qualitätssensitiven Verarbeitung zu realisieren.

6.1.2 Verarbeitungsanfragen

Zur Installation einer Verarbeitungsanfrage für Datenströme in Odysseus existiert Odysseus Script, eine zusätzliche Sprache zur Steuerung von Odysseus. In ihr besteht die Möglichkeit nicht nur Verarbeitungsanfragen zu definieren, sondern auch das System zu konfigurieren. Hierzu zählen zum einen die gewünschte Art von Metadaten, welche innerhalb eines Stromelements verwendet werden sollen, aber auch die Festlegung von Ausführungsstrategien für Operatoren, sowie unterschiedliche Systemvariablen. Eine typische Anfrage in Odysseus weist dabei den in Abbildung 6.2 dargestellten Aufbau auf.

Zunächst wird der Parser festgelegt, welcher für die eigentliche Verarbeitungsanfrage verwendet werden soll. Zum Zeitpunkt dieser Arbeit existieren Parser für die Anfragesprachen StreamSQL[1], eine an die Structured Query Language (SQL) angelehnte deklarative Anfragesprache, PQL [AGG+12], eine prozedurale Sprache, und das Stream-based And Shared Event processing (SASE) [GADI08], eine Sprache für die Detektion und Verarbeitung von Ereignissen in Datenströmen. In der darauf folgenden Zeile werden die zu verwendeten Metadaten festgelegt. Das Odysseus-System selbst verwendet bereits für die Repräsentation der Gültigkeit von Elementen den Intervallansatz mit jeweils einem Start und einem Endzeitstempel in den Metadaten eines Stromelementes. Die Verwendung von Zeitintervallen ist daher standardmäßig aktiviert. Zusätzlich können hier Systemvariablen gesetzt werden. Durch die Instruktion *RUNQUERY* in Odysseus Script wird die nachfol-

[1] http://www.streambase.com

gende Anfrage installiert. Die Anfrage selbst kann dabei auf zuvor definierte Konstanten und bereits installierte Quellen und Views zugreifen.

Nach der Installation einer Anfrage durch eine Anwendung wird diese zunächst über den eingestellten Parser für die Anfragesprache innerhalb der Übersetzungskomponente in einen logischen Operatorgraphen überführt. Anschließend wird in einer Restrukturierungsphase durch die Anwendung von Restrukturierungsregeln der logische Operatorgraph optimiert. Zu diesen Regeln gehören unter anderem die Optimierungsregeln der relationalen Algebra. Ziel dieser Restrukturierungsregeln ist es etwa die Datenlast bei der Verarbeitung frühzeitig zu minimieren, in dem beispielsweise Selektionsoperatoren nahe zu den Quellen verschoben werden ohne dabei die Semantik der Anfrage zu verändern.

In einem weiteren Schritt wird dieser logische optimierte Operatorgraph an die Transformationskomponente übergeben, die für jeden darin enthaltenen logischen Operator einen physischen Operator auswählt und dabei den logischen Operator durch den physischen Operator im Operatorgraphen ersetzt. Nach der erfolgreichen Umwandlung aller logischen Operatoren wird der physische Operatorgraph an den Plan-Manager zur Ausführung übergeben. Wie bereits in dem Architekturbild angedeutet bietet das System die Möglichkeit, jede dieser Komponenten zu erweitern, wie etwa durch weitere Anfragesprachen, weitere Restrukturierungsregeln oder neue Transformationsregeln, sowie neue logische und physische Operatoren.

6.2 Darstellung und Verarbeitung von Qualitäten

Für die Darstellung der Qualitätsinformationen wurden zusätzliche Komponenten implementiert, die die bisherige Datenstruktur in Odysseus erweitern. Die Erweiterungen lassen sich dabei in eine Erweiterung der Nutzdaten und einer Erweiterung der Metadaten differenzieren.

6.2.1 Nutzdaten

Zur Darstellung von stochastischen Modellen innerhalb eines Tupels wurde das bisherige Tupel wie in Abbildung 6.3 erweitert. Ein probabilistisches Tupel leitet dabei die Eigenschaften eines Tupels ab und ergänzt diese durch eine zusätzliche Ebene für die Mischverteilung. Eine Mischverteilung selbst hat dabei die Attribute *scale* und *support*, die die Mischverteilung entsprechend skaliert und den Definitionsbereich in Form von numerischen Intervallen vorgibt. Des Weiteren verfügt eine Instanz der Mischverteilung über eine Liste von Attributpositionen, umso von einer Mischverteilung wieder auf die Attribute zu schließen, die durch die Mischverteilung beschrieben werden. Der Index des Attributes innerhalb der Liste gibt dabei die Dimension innerhalb der Verteilung wieder. Eine Instanz der Mischverteilung besteht zudem noch aus einer Menge von multivariaten Normalver-

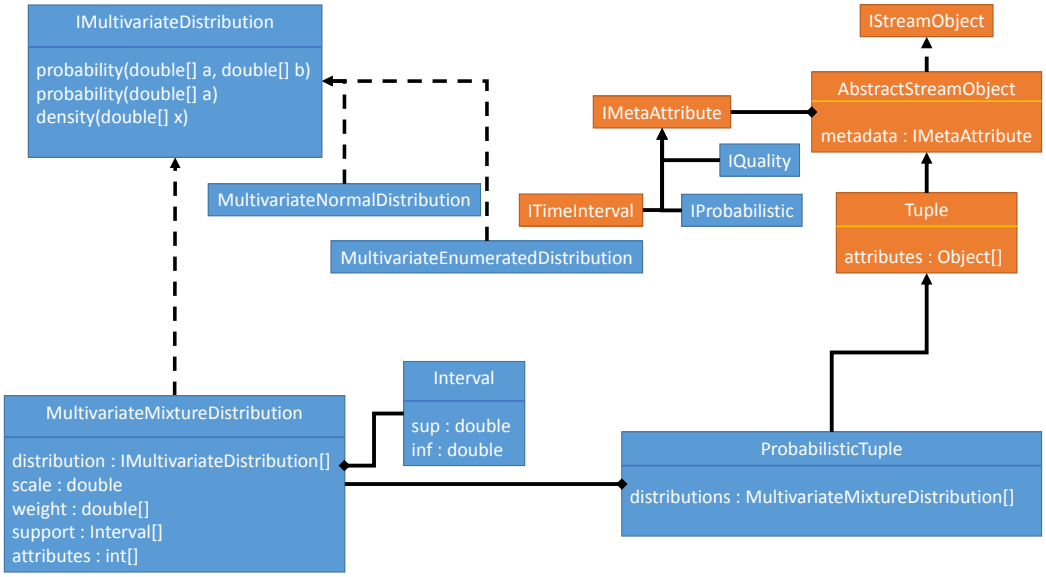


Abbildung 6.3: *Aufbau eines probabilistischen Datenstromelements*

teilungen oder multivariaten diskreten Verteilungen, sowie einem Gewichtsvektor, der die jeweilige Gewichtung der Verteilung innerhalb der Mischverteilung wiedergibt.

Die Trennung der ursprünglichen Nutzdaten von den Mischverteilungen erlaubt es die bisherige Anfragemöglichkeit und die Operatoren des Systems weiterhin uneingeschränkt zu nutzen. Allerdings muss hierzu bei der Verarbeitung in den neuen probabilistischen Operatoren zuvor eine Dereferenzierung von den Nutzdaten auf die Mischverteilungen stattfinden bevor die eigentliche Verarbeitung innerhalb der Operatoren durchgeführt werden kann. Gleichzeitig wird auf diese Weise sichergestellt, dass auch Korrelationen zwischen Attributen, also mehrdimensionale stochastische Modelle, dargestellt und verarbeitet werden können.

6.2.2 Metadatenebene

Im Folgenden fokussieren wir uns auf die Metadaten (Abb. 6.4), welche mit einem probabilistischen Nutzdatentupel verknüpft werden können. Die Metadaten in Odysseus implementieren die Schnittstelle *IMetadata*. In der bisherigen Implementierung von Odysseus bestehen die Metadaten aus einem Zeitintervall in dem die Gültigkeit eines Tupel dargestellt wird. Zur zusätzlichen Repräsentation der Qualitätsinformationen wurden hier zwei zusätzliche Metadaten integriert. Zum einen wird die Existenzwahrscheinlichkeit über eine Implementierung der *IProbability*-Schnittstelle hinterlegt und zum anderen werden die Qualitätsindikatoren Vollständigkeit und Konsistenz aus Kapitel 4.3 über eine Implemen-

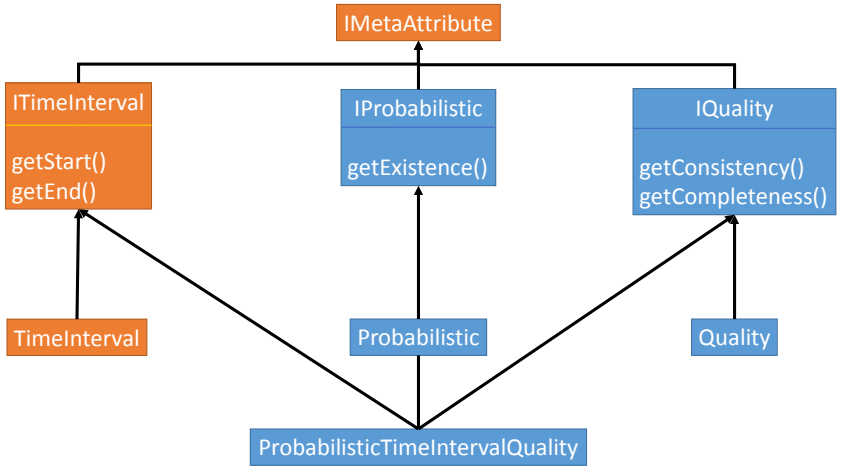


Abbildung 6.4: *Aufteilung der Metadaten eines probabilistischen Datenstromelements*

tierung der *IQuality*-Schnittstelle dargestellt. Diese zusätzlichen Metadaten werden dabei, genauso wie die Zeitstempel, von den Operatoren in dem Odysseus-System von der Eingabe in die Ausgabe kopiert, soweit sie nicht von einem Operator selbst verändert werden oder durch einen Verbund von zwei Tupeln kombiniert werden.

Die Trennung der beiden Qualitätsinformationen hat den Vorteil, dass so auch eine deterministische Verarbeitung unter Berücksichtigung der Vollständigkeit und der Konsistenz geschehen kann ohne Verwendung der probabilistischen Operatoren und umgekehrt, eine rein probabilistische Verarbeitung ohne die Betrachtung weiterer Qualitätsdimensionen stattfinden kann.

6.2.3 Logische Operatoren

Zusätzlich zur Erweiterung der Nutz- und Metadaten werden zusätzliche logische Operatoren benötigt, um auf die neuen Metadaten zuzugreifen und die in Kapitel 4 beschriebenen Konzepte zur indirekten und direkten Bestimmung der Qualitäten zu ermöglichen. Hierzu wurden vier zusätzliche logische Operatoren mit entsprechenden Transformationsregeln entwickelt. In Abbildung 6.5 sehen wir die einzelnen logischen Operatoren als UML-Klassendiagram. Alle logischen Operatoren leiten dabei von der abstrakten Klasse *AbstractLogicalOperator* ab, welche die Basisklasse aller logischen Operatoren in Odysseus bildet. Die hier in Orange dargestellten logischen Operatoren sind bereits fester Bestandteil des Odysseus-Systems. Da sich an der Semantik der logischen Operatoren für die Abbildung, die Projektion, die Selektion und den Verbund keine Änderungen ergeben, ist es hier nicht notwendig neue logische Operatoren zu definieren.

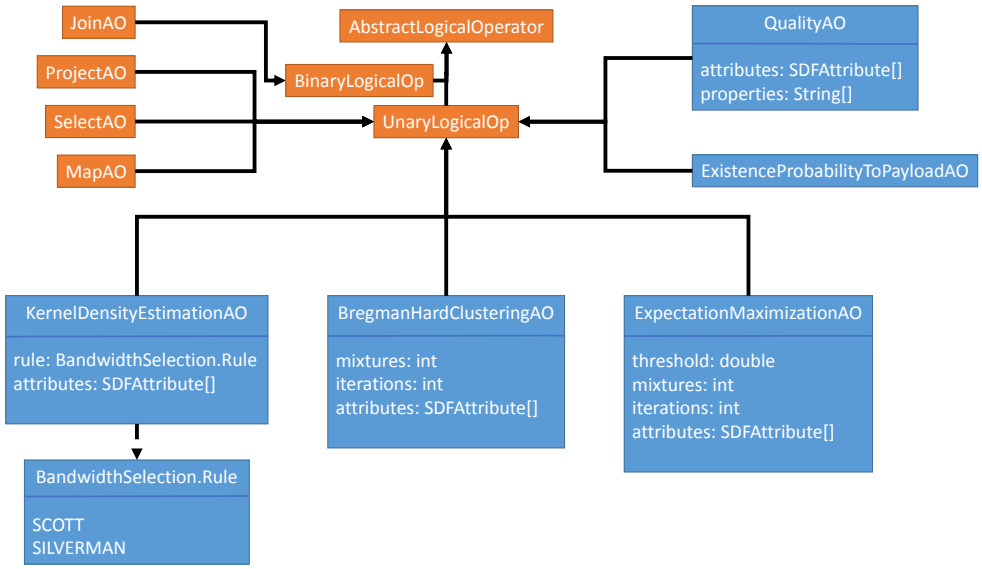


Abbildung 6.5: *Integration der logischen Operatoren in Odysseus*

Der *QualityAO*-Operator dient zur indirekten Bestimmung der Qualität durch die Nutzung der SSN-Ontologie. Der Operator benötigt zwei Parameter, die Namen der Attribute innerhalb des Eingangsschemas, sowie die Namen der Qualitätseigenschaften, die bestimmt werden sollen. Der *ExistenceProbabilityToPayloadAO*-Operator hat die Aufgabe, die Existenzwahrscheinlichkeit eines Tupels in die Nutzdaten des logischen Ausgabestroms abzubilden. Durch diese Funktionalität kann bei der Verarbeitung etwa innerhalb eines Selektionsoperators auf die Existenzwahrscheinlichkeit eines Tupels gefiltert werden. Für die Qualitätsdimensionen Vollständigkeit und Konsistenz, welche durch die Metadaten der *IQuality*-Schnittstelle bereit stehen, existierende entsprechende Gegenstücke.

Die drei logischen Operatoren zur Ermittlung des stochastischen Modells im unteren Teil benötigen jeweils die Namen der Attribute in dem Eingangsschema auf deren Werte das stochastische Modell bestimmt werden soll. Der Kerndichteschätzungsoperator bietet zudem die Möglichkeit, zwischen verschiedenen Regeln zur Bestimmung der Bandbreite zu wählen. In der vorliegenden Implementierung sind dies die Scott-Regel und die Silverman-Regel. Der Operator zur Anwendung des Bregman-Hard Clusterings und der Operator zur Erwartungswertmaximierung benötigen zusätzlich die Anzahl an maximalen Iterationen und die Anzahl an Komponenten der resultierenden Mischverteilung bzw. die Anzahl an Clustern. Während bei dem Bregman-Hard Clustering eine gleichbleibende Partitionierung der einzelnen Cluster als Abbruchbedingung verwendet wird, benötigt das Erwartungswertmaximierungsverfahren zusätzlich einen Schwellwert der Log-Likelihood als Abbruchbedingung.

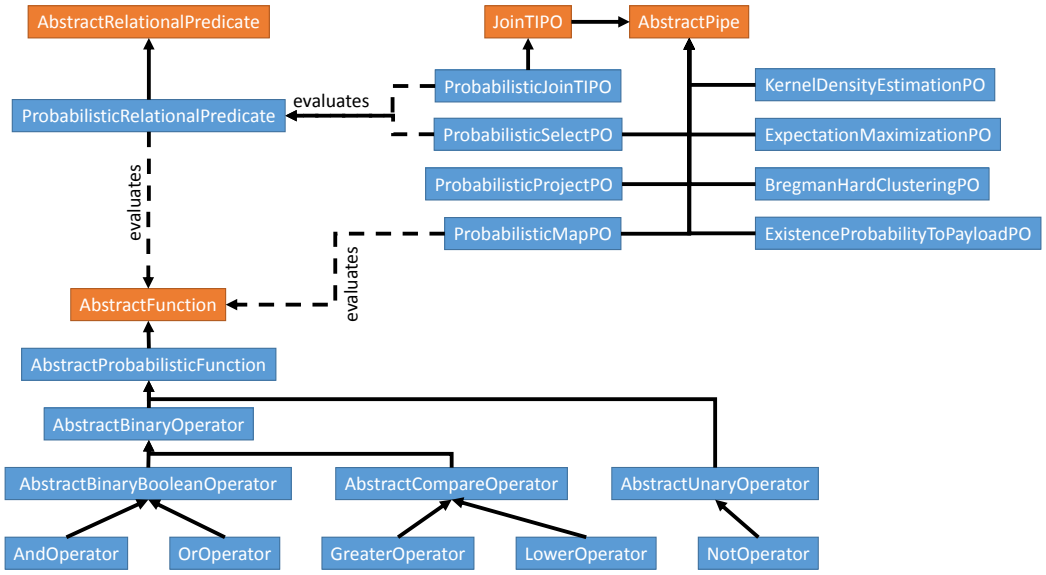


Abbildung 6.6: *Integration der physischen Operatoren in Odysseus*

6.2.4 Physische Operatoren

Als Gegenstück zu den logischen Operatoren, welche die Semantik der Verarbeitung beschreiben, wird im Folgenden auf die konkrete Implementierung der physischen Operatoren eingegangen. In Abbildung 6.6 sind alle im Rahmen dieser Arbeit entwickelten Operatoren abgebildet. Zusätzlich zu den Gegenstücken für die logischen Operatoren zur Ermittlung der stochastischen Modelle sind hier auch die physischen Gegenstücke zu den logischen Operatoren Abbildung, Projektion, Selektion und Verbund dargestellt, die während der Transformationsphase zur Verarbeitung von probabilistischen Tupel in den Verarbeitungsgraphen eingesetzt werden. Alle physischen Operatoren bauen dabei auf der bereits existierenden abstrakten Klasse *AbstractPipe* auf.

Im unteren Teil des UML-Diagramms sind zudem die zusätzlichen notwendigen mathematischen Funktionen dargestellt, welche benötigt werden um einfache Abbildungsfunktionen und Filterprädikate auf den probabilistischen Daten zu definieren. Hierzu zählen die Operatoren der booleschen Algebra und die Vergleichsoperatoren. Des Weiteren existieren als Gegenstück zu den arithmetischen Operatoren für numerische Werte zusätzliche Funktionen zur Bestimmung des resultierenden stochastischen Modells von Additionen, Subtraktionen, Multiplikation und Division zwischen zwei stochastischen Modellen und zwischen deterministischen numerischen Werten und stochastischen Modellen. Wie bereits zuvor erwähnt, ist das Resultat einer Division und einer Multiplikation eine Produktverteilung, weshalb in diesem Fall eine erneute Anpassung einer Mischverteilung aus Normalverteilung an die resultierende Produktverteilung stattfindet, umso als Resultat wieder

eine Mischverteilung aus Normalverteilungen für die weitere Verarbeitung zu erhalten. Die Wahl des geeigneten Operators wird abhängig von dem jeweiligen Datentyp des Attributes im Eingangsschema eines logischen Operators während der Transformationsphase bestimmt. Zu diesem Zweck wurde der Parser der Anfragesprache dahingehend erweitert, dass die mathematischen Operatoren durch Funktionen in anderen OSGi-Bundles zur Laufzeit überschrieben werden können und so eine datentypunabhängige Formulierung von Anfragen stattfinden kann. Konkret bedeutet dies, dass eine existierende, auf deterministischen Daten basierende, Anfrage nun ohne Umformulierung der Anfrage selbst auf probabilistischen Daten angewendet werden kann. Zu diesem Zweck verfügt jede registrierte Funktion über eine Liste von akzeptierbaren Datentypen.

Zusätzlich zu den physischen Operatoren existiert für jeden logischen Operator eine Transformationsregel, welche den logischen Operator durch sein physisches Gegenstück ersetzt. Dies gilt auch für die existierenden logischen Operatoren.

6.2.5 Komponenten

Für den flexiblen Einsatz der Implementierungen wurden die Implementierungen der einzelnen Konzepte in unterschiedliche Komponenten (OSGi-Bundles) aufgeteilt. Dabei ist die Funktionalität zur probabilistischen Verarbeitung und der direkten Qualitätsbestimmung getrennt von der indirekten Qualitätsbestimmung über eine Ontologie.

6.2.5.1 Probabilistische Verarbeitungskomponente

Die Verarbeitungskomponente für probabilistische Daten bestehen aus drei Teilkomponenten, der Server-Komponente, der Client-Komponente und der Common-Komponente die im Folgenden näherer erläutert werden.

Server

Die Server-Komponente enthält alle logischen und physischen Operatoren, sowie die für Abbildungen und Selektionen notwendigen arithmetischen und booleschen Funktionen für Mischverteilungen. Für die Überführung eines logischen in einen physischen Plan enthält die Komponente zudem die notwendigen Transformationsregeln für die enthaltenen Operatoren. Dabei wird für die Transformation eine im Bezug auf die existierenden Transformationsregeln des Systems höhere Priorität verwendet um sicherzustellen, dass die Regeln für die Basisoperatoren vor den existierenden Regeln des Systems innerhalb der Transformationsphase ausgeführt werden. Zusätzlich zu den in dieser Arbeit beschriebenen Operatoren beinhaltet die Serverkomponente noch weitere Operatoren zur Generierung von Stichproben aus Mischverteilungen und zur kontinuierlichen Verbesserung von Verteilungen durch einen Kalman Filter.

Client

Die Client-Komponente enthält Visualisierungsklassen zur Darstellung von ein- und zweidimensionalen Mischverteilungen. Durch diese zusätzlichen Darstellungsmöglichkeiten lassen sich die Ergebnisse einzelner Operatoren im physischen Operatorgraphen visuell darstellen und überprüfen.

Common

In diesem Teil befinden sich Klassen und Schnittstellen um probabilistische Stromelemente sowohl auf Server-, wie auch auf Client-Seite verwenden zu können. Hierzu zählen zum einen das probabilitische Tupel selbst, sowie Schnittstellen zu Darstellung von Mischverteilungen. Die konkrete Ausführung der Mischverteilung ist dabei nicht Teil dieser Komponente. Dies Erlaubt es nachträglich weitere Arten von Verteilungen innerhalb einer Mischverteilung zu nutzen. Eine Mischverteilung selbst muss dazu die vier Funktionen *probability*, *density* und *sample*, sowie die mathematischen Operatoren *add*, *subtract*, *multiply*, und *divide* realisieren.

6.2.5.2 Ontologieunterstützung

Die Verarbeitungskomponente zur Nutzung einer Ontologie als Wissensbasis zur Modellierung von Beziehungen zwischen Sensoren besteht aus vier Teilkomponenten, der Server-Komponente, der Client-Komponente, der Common-Komponente und der Ontology-Komponente.

Server

Die Server-Komponente besteht aus den logischen und physischen Operatoren, sowie Transformationsregeln und Restrukturierungsregeln. Die Restrukturierungsregeln werden während der Restrukturierungsphase aufgerufen und binden mit Hilfe der SSN-Ontologie die notwendigen Quellen in den logischen Graphen ein, umso die Qualität von Stromelementen zu bestimmen. Zu diesem Zweck wird der Verarbeitungsgraph um die in Kapitel 4 aufgezeigte Kombination aus Projektions-, Verbund- und Abbildungsoperator erweitert. Dies geschieht noch vor der eigentlichen Optimierung des logischen Operatorgraphen, so dass die eingefügten Operatoren bei der Optimierung durch die Anwendung der Optimierungsregeln auf Basis der relationalen Algebra mit berücksichtigt werden können.

Ontology

In dieser Komponente befindet sich die konkrete Ontologie, sowie OSGi-Service Komponenten, welche die Ontologie als Dienst innerhalb der OSGi-Umgebung anbieten. Auf diese Weise ist die Ontology-Komponente von der Server-Komponente getrennt, so dass diese

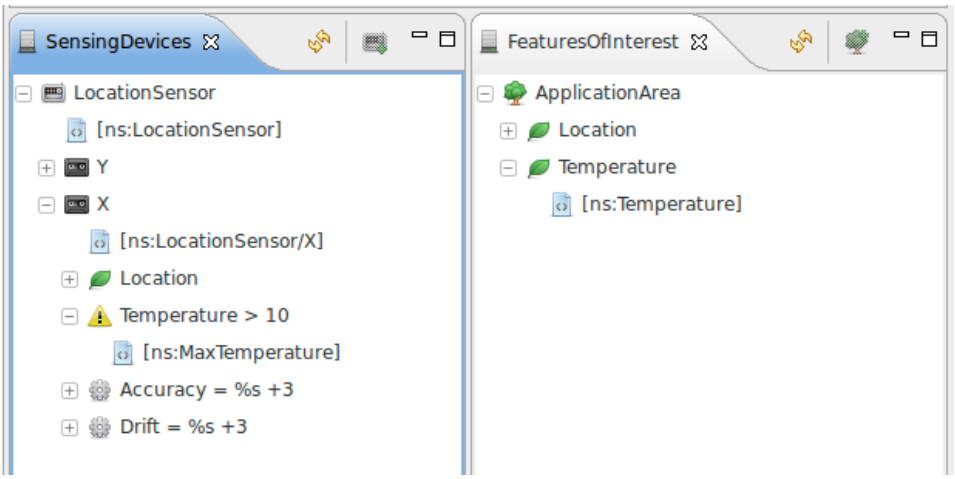


Abbildung 6.7: *GUI-Element zur Anzeige von Sensoren, Messmöglichkeiten und Qualitätseinschränkungen*

durch ein anderes Ontologieframework ausgetauscht werden kann. Für die Verwendung der SSN-Ontologie als Wissensdatenbank für die Verknüpfung von Sensoren und ihren Messwerten wird in der aktuellen Implementierung die Apache Jena Bibliothek[2] verwendet.

Client

Über die Client-Komponente können innerhalb der Odysseus-GUI neue Sensoren, Einsatzgebiete und Messmöglichkeiten mit ihren Qualitätseigenschaften definiert werden. Des Weiteren gibt sie dem Anwender eine Übersicht über bereits modellierte Sensoren und ihre Messmöglichkeiten.

In Abbildung 6.7 ist hierzu die grafische Oberfläche zur Verwaltung der Sensoren und ihren Messmöglichkeiten dargestellt. Hierbei wurde ein Sensor mit dem Bezeichner *LocationSensor* mit den zwei Messmöglichkeiten *X* und *Y* erstellt, die eine Eigenschaft *Location* in dem Einsatzgebiet *ApplicationArea* wahrnehmen. Des Weiteren sind für die Messmöglichkeit *X* zwei Qualitätseigenschaften *Accuracy* und *Drift* mit je einem mathematischen Ausdruck definiert. Diese zwei Qualitätseigenschaften gelten für die Messmöglichkeit wenn die Eigenschaft *Temperature* einen Wert von 10 übersteigt. Durch die grafische Oberfläche kann der Anwender jederzeit neue Sensoren, aber auch neue Messmöglichkeit mit dazugehörigen Bedingungen und Qualitätseigenschaften erstellen. Die dabei eingegebenen Daten werden direkt in der Ontologie abgelegt, von wo aus sie wiederum bei der Erstellung von Verarbeitungsanfragen abgerufen werden können.

[2] http://jena.apache.org

Common

In diesem Teil befinden sich Klassen und Schnittstellen, die sowohl von Client-, wie auch von Server-Seite verwendet werden. Hierzu zählen die Modellklassen zur Serialisierung von Java-Objektklassen auf Klassen innerhalb der Ontologie und die verwendeten Vokabeln innerhalb der Ontologie.

6.3 Anwendungen

Im Folgenden betrachten wir die in Kapitel 1.2 vorgestellten Anwendugnsszenarien und wie diese durch die vorgenommenen Erweiterungen des Odysseus-Datenstrommanagements realisiert werden konnten.

6.3.1 Fahrerlose Transportsysteme

Das erste Szenario ist dabei aus dem Bereich der fahrerlosen Transportsysteme und wurde im Rahmen des Forschungsprojekts SALSA [TEK+12] realisiert. Hierbei nehmen die fahrerlosen Transportsysteme ihre Umgebung über fest verdrahtete Sensoren am Transportsystem, wie etwa Laserscanner oder Kameras, wahr. Zur Vermeidung von Kollisionen mit anderen Objekten werden Umgebungsinformationen durch interne Routinen für die lokal angebrachte Sensorik verarbeitet und im Falle einer kritischen Situation ein Nothalt ausgelöst. Im Folgenden wird gezeigt, wie durch die Erweiterungen des Datenstrommanagementsystems die Sensoren in der Umgebung fusioniert werden konnten und als dynamisches Kontextmodell dem fahrerlosen Transportsystem zur Verfügung gestellt wurden.

Die Umgebung des Fahrzeugs wird dabei von zwei Laserscannern des Typs LMS100 und einem Laserscanner des Typs LMS511 der Firma SICK[3] überwacht. Die drei Laserscanner messen kontinuierliche die Distanzen zu Objekten in einem Winkel von $270\,^\circ$ und einer Winkelauflösung von $0.25\,^\circ$ bei einer Frequenz von 25Hz. Diese Messungen entsprechen einer Punktwolke mit 1080 Distanzmessungen alle 40 Millisekunden. Diese Punktwolke wird dabei über Ethernet in dem herstellereigenen Format zu dem Datenstrommanagementsystem übermittelt und verarbeitet.

Das Demonstrationsfahrzeug selbst ist ebenfalls mit zwei Laserscannern ausgestattet, die jeweils vorne am Fahrzeug (vgl. Abb. 6.8) montiert sind. Zudem verfügt das Demonstrationsfahrzeug über eine DGPS-Antenne, die eine sehr exakte Positionierung erlaubt. Auf Basis der Laserscanner am Fahrzeug wird zunächst lokal eine Belegungskarte erstellt. Die Laserscanner in der Umgebung werden innerhalb einer Basisstation zu einer globalen Belegungskarte aggregiert. Zu Verknüpfung der beiden Kontextmodelle sendet das Fahrzeug die aktuelle Position über eine drahtlose Verbindung an die Basisstation und erhält

[3] www.sick.com

Abbildung 6.8: *Demonstrator in dem Projekt SALSA*

daraufhin einen Teilausschnitt der globalen Belegungskarte um die aktuelle Position des Fahrzeugs. Anschließend werden die beiden Kontextmodelle im Fahrzeug kombiniert, wobei die Informationen aus dem lokalen Kontextmodell in dem Sinne dominieren, als dass Objekte im lokalen Kontextmodell nicht durch andere Informationen im globalen Kontextmodell entfernt werden können. Einen Teil der Anfrage, welche innerhalb des Datenstrommanagementsystems in der Basisstation ausgeführt wird findet sich in Listing 6.1. Hierbei werden zunächst die Daten aller Laserscanner über den Vereinigungsoperator in einen Strom zusammen geführt. Anschließend wird der Strom durch das Belegungsgitter aus dem Kontextspeicher *myStore* erweitert. Dieses Kontextmodell wird daraufhin auf den aktuellen Stromzeitpunkt mit Hilfe der *spread*-Methode prädiziert. Sollte hierbei noch kein Belegungsgitter in dem Kontextspeicher vorhanden gewesen sein oder veraltet sein, wird ein neues Belegungsgitter erstellt. Die Überprüfung hierfür findet über die *eif*-Methode statt. Aufbauend auf den Ergebnissen aus der *spread*-Methode werden mit der *merge*-Methode die Messungen in das globale Belegungsgitter integriert. Hierzu wurde als Parameter für die Methode eine Standardabweichung der Laserscannermessungen von 4.2 Millimeter angenommen. Anschließend wird dieses Gitter zurück in den Kontextspeicher geschrieben und zusätzlich mit der aktuellen Position des Fahrzeugs verknüpft (nicht mehr im Teil der dargestellten Anfrage) um den Bereich um die Fahrzeugposition herauszuschneiden. Dieses nun auf die Position des Fahrzeugs reduzierte Belegungsgitter wird anschließend serialisiert und an das Fahrzeug übertragen.

Zur Realisierung dieser Anwendung war es zudem notwendig das Kommunikations- und Steuerungsprotokoll der Laserscanner zu implementieren und ein Kommunikationsprotokoll zur Übertragung der Belegungskarte an das fahrerlose Transportfahrzeug zu ent-

```
Grid = MAP({EXPRESSIONS = [['streamtime()','timestamp'],['
   merge(grid, cellsize, distanceMatrix, x, y, radius, 4.2)'
   ,'grid']]},
    MAP({EXPRESSIONS = [['spread(eif(!isNull(grid),grid,
       toSpatialGrid(width, height)), eif(!isNull(timestamp)
       ,timestamp,streamtime()), velocity)','grid'],['
       distanceMatrix','distanceMatrix'],['x','x'],['y','y'
       ]]},
      CONTEXTENRICH({STORE = 'myStore', OUTER = true},
        UNION(
          LMS1,LMS2,LMS3
        )
      )
    )
)
ContextStore = STORE({STORE = 'myStore'},
  Grid
)
```

Quelltext 6.1: *Bestimmung der Belegungskarte für das Szenario autonome Fahrzeuge*

werfen. Die Verarbeitung und die Visualisierung der Belegungskarte wurde auf der 6ten ACM Distributed Event-Based Systems Konferenz [KGS+12] mit Konferenzteilnehmern erprobt und mit dem Best Demonstration Award ausgezeichnet.

6.3.2 Sichere Offshore-Operation

Bei Offshore-Operationen, wie etwa der Konstruktion von Offshore-Windrädern, kommt es immer wieder zu Unfällen, weil sich die beteiligte Mitarbeiter in Gefahrenbereichen, wie etwa unter einer schwebenden Last, aufhalten. Zur Sicherstellung von Offshore-Operationen wird daher die Position von Mitarbeitern und Gütern erfasst und kritische Situationen durch ein Assistenzsystem frühzeitig erkannt und entsprechend im Falle einer kritischen Situation die beteiligten Personen gewarnt. Die hier vorgestellten Erweiterungen des Systems wurden im Rahmen des Forschungsprojekts SOOP [SBL+12] realisiert. Die Erweiterungen werden dazu verwendet eine qualitätssensitive Verarbeitung zu ermöglichen und die Qualität der Verarbeitungsergebnisse bei der Bestimmung von kritischen Situationen innerhalb des Assistenzsystems zu nutzen. Hierzu wurde das Deck eines Schiffs in Abbildung 6.9 genutzt um eine Verladeoperation zu simulieren. Die Teilnehmer trugen dabei Positionsfunksensoren, die jeweils ihre aktuelle Position an das Assistenzsystem übermitteln.

Abbildung 6.9: *Demonstrator in dem Projekt SOOP*

Die Verarbeitungsanfrage in Listing 6.2 bestimmt zunächst mit Hilfe des Erwartungswertmaximierungsoperators das stochastische Modell aus den Positionssensormessungen der Teilnehmer. Das stochastische Modell der Position wird dabei auf den letzten 100 Messungen durchgeführt. Als Parameter wurde der Erwartungswertmaximierungsoperators so konfiguriert, dass er eine zwei-komponentige Mischverteilung auf den Attributen x und y des Eingangsstroms bestimmt und dabei einen Schwellwert für die Log-Likelihood von 0.001 und eine maximale Anzahl von 30 Iterationen durchläuft. Darauf aufbauend wird eine kritische Situation durch die Verwendung des Selektionsoperators ermittelt. Die Funktion *as2DVector* transformiert dabei die beiden Attribute mit probabilistischen Werten in einen Vektor umso eine Selektion im zweidimensionalen Raum zu ermöglichen.

Für dieses Szenario war es notwendig, das von den Sensoren für die Kommunikation genutzte Protokoll zu implementieren. Die Verarbeitung und die Visualisierung der stochastischen Modelle wurde auf der 8ten ACM Distributed Event-Based Systems Konferenz [KN14a] mit den Konferenzteilnehmern durch die Verwendung eines Ultraschallsensors statt eines Funkknotens demonstriert.

6.4 Zusammenfassung

In diesem Kapitel wurde gezeigt, wie die entwickelten Konzepte aus den Kapiteln 3, 4 und 5 in dem Datenstrommanagementsystem Odysseus integriert wurden. Die entwickelten

```
Supervisor = EM({ATTRIBUTES = ['x','y'], MIXTURES = 2,
   ITERATIONS=30, THRESHOLD=0.001},
    ELEMENTWINDOW({size = 100},
       UWB
    )
)
Hazard = SELECT({predicate =
   ProbabilisticRelationalPredicate('as2DVector(x,y) > [
   MIN_X,MIN_Y] && as2DVector(x,y) < [MAX_X,MAX_Y]')},
    Supervisor
)
```

Quelltext 6.2: *Erkennung von kritischen Situationen für das Szenario Sichere Offshore-Operationen*

Komponenten für die Verarbeitung von Existenzwahrscheinlichkeiten und der indirekten Bestimmung von Qualitäten durch eine Ontologie sind dabei getrennt voneinander implementiert worden und erlauben so eine höhere Flexibilität in möglichen Anwendungen.

Des Weiteren sind die Implementierungen der beiden Konzepte jeweils in Client- und Server-Implementierung aufgeteilt, um sie so auch getrennt voneinander in einer verteilten Umgebung zu verwenden. Zusätzlich zu den entwickelten Konzepten wurden Protokolle für die Kommunikation zwischen dem Datenstrommanagementsystem und den in den Szenarien verwendeten Sensoren implementiert, die die Umwandlung der Messdaten in die interne Struktur eines Tupels übernehmen bzw. dieses wieder in ein anwendungsspezifisches Format überführen. Anschließend wurden die zwei betrachteten Szenarien aufgezeigt, in denen die Implementierung der Konzepte verwendet wurde. Eine genauere Evaluation der einzelnen Konzepte und implementierten Verfahren erfolgt nun in Kapitel 7.

7 Evaluation

In diesem Kapitel findet eine Evaluation der einzelnen Konzepte statt, welche im Rahmen dieser Arbeit entwickelten wurden. Hierzu werden zunächst die Implementierungen der einzelnen Konzepte in einem Datenstrommanagementsystem (DSMS) mit synthetischen Daten, als auch mit realen Sensordaten auf ihre Latenz und die Qualität der Verarbeitung geprüft. Anschließend werden die entwickelten Konzepte im Rahmen der Fallstudie Sichere Offshore-Operationen in ihrem Gesamtbild auf ihre Anwendbarkeit hin evaluiert.

7.1 Evaluierung einzelner Konzepte

Im Folgenden werden zunächst die entwickelten Erweiterungen der temporalen relationalen Operatoren für die Verarbeitung von stochastischen Modellen in Form von Mischverteilungen geprüft. Der Fokus der Evaluation liegt hierbei auf der Latenz der Operatoren in Abhängigkeit zu der Anzahl an Komponenten in den zu verarbeiteten Mischverteilungen. Hierzu wird ein Datenstrom simuliert, der jeweils aus Mischverteilungen mit 1, 2, 4 oder 8 Komponenten besteht. Während in den meisten Anwendungen, in denen Positionsdaten verarbeitet werden, das Rauschen der Sensoren durch eine Gauß-Verteilung adäquat modelliert werden kann [MM88], dienen die Datenströme mit 4- und 8-komponentigen Mischverteilungen als obere Abschätzung des entwickelten Verfahrens. Soweit die Operatoren besondere Konfigurationen aufweisen, werden diese im Folgenden an gegebener Stelle genannt.

In einem weiteren Schritt werden die Operatoren zur Bestimmung des stochastischen Modells hinsichtlich ihrer Latenz, aber auch hinsichtlich der Güte des stochastischen Modells evaluiert. Zu diesem Zweck werden 10.000 Stichproben aus einer Normalverteilung und aus einer logarithmischen Normalverteilung verwendet um einen Datenstrom aus Messwerten zu simulieren. Die Evaluation der Latenz und der Güte des Modells betrachtet dabei drei Szenarien mit Datenfenstern der Größe 10, 100 und 1000. Das Datenfenster definiert dabei die gültigen Messwerte, auf denen die Operatoren das stochastische Modell bestimmen sollen. Die Güte des Modells betrachtet das aktuell bestimmte stochastische Modell im Hinblick auf alle 10.000 Stichproben. Als Qualitätskriterium wird hierzu das Akaike-Informationskriterium verwendet. Das Akaike-Informationskriterium (AIC) ist ein Maß für die relative Qualität eines stochastischen Modells für eine gegebene Datenmenge und ist definiert als:

$$AIC = 2k - 2\ln(L) \tag{7.1}$$

Der Parameter k repräsentiert hierbei die Anzahl der freien Parameter in dem stochastischen Modell und der Parameter L gibt die Log-Likelihood zwischen dem stochastischen Modell und der gegebenen Datenmenge wieder. Der AIC ist also somit eine Kombination aus der Nähe des Modells zu der gegebenen Datenmenge und der Komplexität des stochas-

tischen Modells. In diesem konkreten Fall sind die freien Parameter der Erwartungswertvektor und die Kovarianzmatrix der Komponenten innerhalb der Mischverteilung.

Dieses Informationskriterium ist für die Evaluation deshalb gut geeignet, da es sowohl die Nähe der generierten Mischverteilung aus den drei Verfahren zu den tatsächlichen Daten bewertet und zudem die Anzahl der Komponenten innerhalb der Mischverteilungen in die Bewertung mit einfließen lässt. Die Nähe zu den tatsächlichen Daten ist wichtig für die Qualität der Verarbeitungsergebnisse und die Anzahl der Komponenten innerhalb der Mischverteilung hat eine Auswirkung auf die Latenz der Verarbeitung, da für jede Komponente innerhalb einer Mischverteilung bei relationalen Operationen wie der Selektion oder dem Verbund mit einem Selektionskriterium bei einer probabilistischen Verarbeitung das Integral gebildet werden muss.

Jede Evaluation wurde dabei 10-mal wiederholt um mögliche Ausreißer zu minimieren. Als Testsystem diente ein Lenovo Thinkpad X240 mit Intel Core i7 und 8GB RAM. Die verwendete Java Laufzeitumgebung war ein OpenJDK Runtime Environment (IcedTea 2.5.2) (7u65-2.5.2-2) mit einer OpenJDK 64-Bit Server VM (build 24.65-b04, mixed mode). Bei dem Betriebssystem handelte es sich um ein Debian GNU/Linux mit einem 3.14 Kernel.

7.1.1 Probabilistische Verarbeitungsoperatoren

Für die Evaluation der beschriebenen Verarbeitungsoperatoren wurde jeweils die Latenz der einzelnen Tupel vom Systemeingang bis hin zur Ausgabe gemessen.

7.1.1.1 Selektion

Für die Evaluation des Selektionsoperators wurden die Messung mit der zuvor genannten Anzahl an Komponenten pro Mischverteilung mit jeweils einer univariaten Verteilung und einer zweidimensionalen multivariaten Verteilung durchgeführt. Im Falle einer multivariaten Mischverteilung wird dabei die Existenzwahrscheinlichkeit eines Tupels über den Genz-Algorithmus [Gen92] angenähert. Als Parameter für die Anzahl der Stichproben innerhalb des Genz-Algorithmus wurde der Wert 5000 gewählt, wie er auch in dem bereitgestellten Quelltext des Autors verwendet wird. In der Abbildung 7.1 sehen wir das Latenzverhalten über die Zeit bei univariaten Mischverteilungen mit 1, 2, 4 und 8 Komponenten. Da der Selektionsoperator ein zustandsloser Operator ist, bleibt hier die Latenz über die Zeit konstant. In Abbildung 7.2 sind alle 4 evaluierten Fälle sowohl für eine univariate als auch für eine multivariate Mischverteilung gegenüber gestellt. Hierbei ist klar zu erkennen, dass die Latenz des Selektionsoperators linear mit der Anzahl an Komponenten der Mischverteilung steigt. Die durchschnittliche Latenz des Selektionsoperators bei der Verarbeitung einer multivariaten Mischverteilung durch Anwendung des Genz-Algorithmus verhält sich ebenfalls linear zu der Anzahl an Komponenten innerhalb der Mischverteilung.

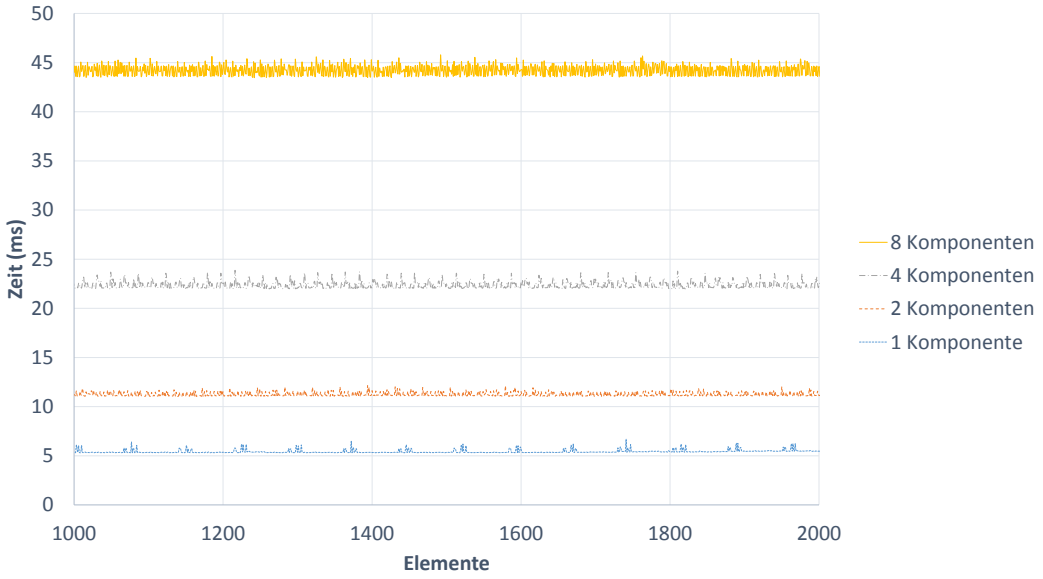


Abbildung 7.1: *Latenz des Selektionsoperators bei einer univariaten Mischverteilung mit 1, 2, 4 und 8 Komponenten*

Aus den Ergebnissen der Evaluation des Selektionsoperators folgt zunächst, dass die Latenz des Selektionsoperators es verbietet, diesen wie einen relationalen Selektionsoperator für deterministische Attribute zu handhaben. Dies gilt insbesondere bei der Optimierung und Restrukturierung von Anfragen, bei der versucht wird eine Selektion in Richtung der Quelle zu verschieben. Stattdessen muss der Selektionsoperator für probabilistische Datenströme vielmehr wie ein Abbildungsoperator betrachtet werden, da bei einer Filterung auf diskrete oder kontinuierliche unsichere Attribute keine Tupel verworfen werden, wie dies bei relationalen Selektionsoperationen für deterministische Attribute der Fall ist, sondern vielmehr eine Abbildung auf ein Tupel mit geringerer Existenzwahrscheinlichkeit stattfindet. Des Weiteren bietet es sich an, aus Gründen der Optimierung und Reduzierung der Datenmenge, resultierende Filterergebnisse mit einer sehr geringen Existenzwahrscheinlichkeit aus dem Ausgabestrom zu entfernen.

7.1.1.2 Projektion

Für die Evaluation des Projektionsoperators wurde ein Tupel mit einer zweidimensionalen multivariaten Mischverteilung erzeugt, welches durch die Projektionsoperation auf eine univariate Mischverteilung projiziert wird. Auch hier wurde jeweils die Latenz für jede der genannten Komponentenanzahl der Mischverteilung evaluiert.

Die Latenz des probabilistischen Projektionsoperators steigt dabei linear mit der Anzahl an Komponenten in der Mischverteilung (vgl. Abbildung 7.3). Dieses Latenzerhalten ist da-

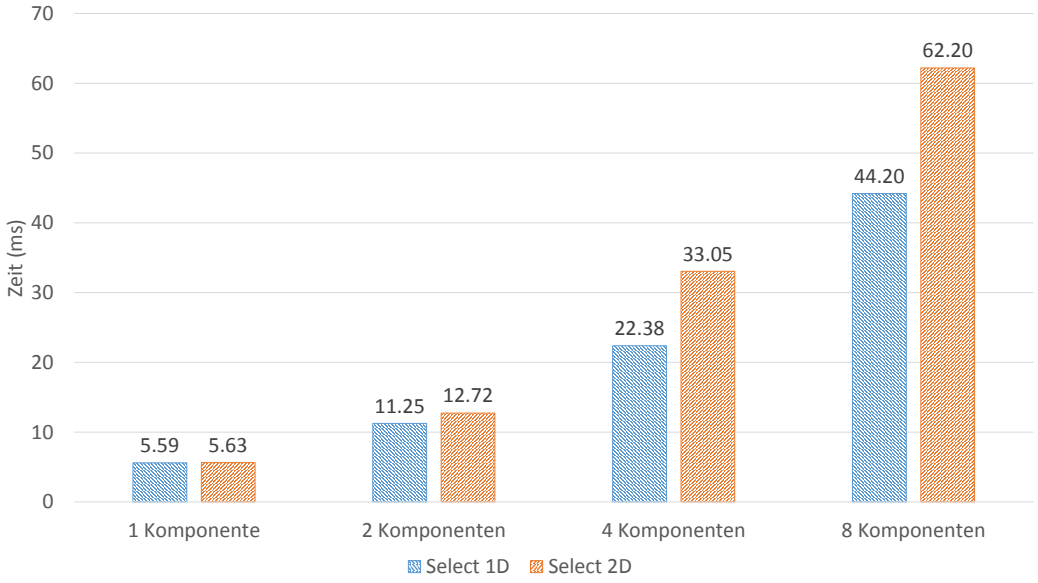


Abbildung 7.2: *Vergleich der durchschnittlichen Latenz des Selektionsoperators in Abhängigkeit zu der Komponentenanzahl einer univariaten/multivariaten Mischverteilung*

durch zu erklären, dass die Projektion auf jeder Komponente der Mischverteilung durchgeführt werden muss, um anschließend eine neue Mischverteilung mit der gewünschten Anzahl an Dimensionen zu erstellen.

7.1.1.3 Verbund

Für die Evaluation des Verbundoperators wurden zwei Datenströme mit jeweils 1-, 2-, 4- und 8-komponentigen Mischverteilungen simuliert Die beiden Ströme werden innerhalb des Verbundoperators über das Verbundkriterium $x1 < 0$ AND $x2 < 0$ konkatiniert, wobei $x1$ eine Mischverteilung aus dem ersten Datenstrom referenziert und $x2$ entsprechend eine Mischverteilung aus dem zweiten Datenstrom referenziert. In Abbildung 7.4 zeigt sich, dass im Gegensatz zum Verhalten des zuvor evaluierten Selektionsoperators, die Latenz exponentiell zur Anzahl der Komponenten der Mischverteilung steigt. Der Grund für die exponentielle Latenz in Abhängigkeit zur der Anzahl der Komponenten liegt darin begründet, dass die resultierende Verteilung innerhalb des Verbundkriteriums über die Konkatenation der Elemente aus beiden Strömen bestimmt werden muss und nicht für sich genommen berechnet werden kann.

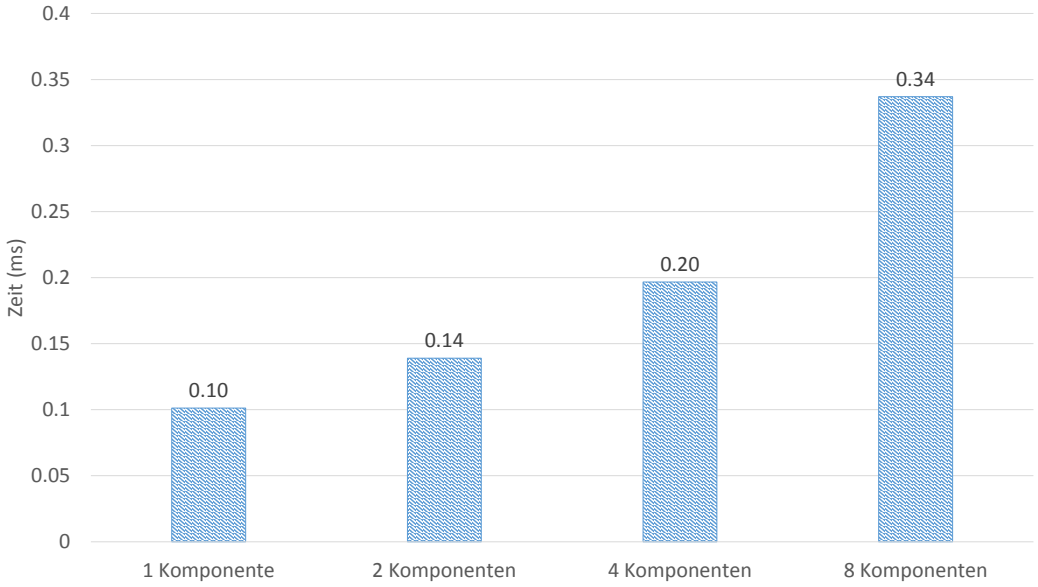


Abbildung 7.3: *Vergleich der durchschnittlichen Latenz des Projektionsoperators in Abhängigkeit zu der Komponentenanzahl einer multivariaten Mischverteilung*

7.1.2 Indirekte Qualitätsbestimmung

Bei der indirekten Qualitätsbestimmung wird der logische Anfrageplan durch zusätzliche Quellen, welche eine Qualitätsauskunft geben können, angereichert. Die Quellen, sowie die Ausdrücke zur Bestimmung der Qualität, stammen dabei aus der Semantic-Sensor-Network-Ontologie (SSN). Da der zusätzliche Anfrageplan nur aus Operatoren der temporalen relationalen Algebra besteht und die Operatoren bereits zuvor evaluiert wurden, wird im Folgenden der Frage nachgegangen, wie viele Verknüpfungen innerhalb der Ontologie repräsentiert werden können und in vertretbarer Zeit abgefragt und zu einem Anfrageplan transformiert werden können. Hierzu wurde schrittweise jeweils ein Hauptsensor, welcher die eigentlich Quelle der Anfrage repräsentiert, sowie drei weitere Sensoren (Sekundarsensoren), die die Umwelteigenschaften *Temperatur*, *Luftdruck* und *Luftfeuchtigkeit* messen, in der Ontologie angelegt. Die Messmöglichkeit des Hauptsensors hängt dabei von allen drei Umwelteigenschaften ab, so dass in der Ontologie die Messmöglichkeit des Hauptsensors mit drei Bedingungen, die jeweils von den Umwelteigenschaften abgeleitet sind, verknüpft ist. Im Folgenden wurde dabei die Zeit gemessen, die benötigt wird um den auswertbaren Ausdruck zur Bestimmung der Qualitätsdimension zu konstruieren. Dieser Aufbau wurde entsprechend für 1 bis 60 Hauptsensoren mit je 3 zusätzlichen Sekundarsensoren für die Umwelteigenschaften und entsprechend 3 Bedingungen pro Messmöglichkeit durchgeführt.

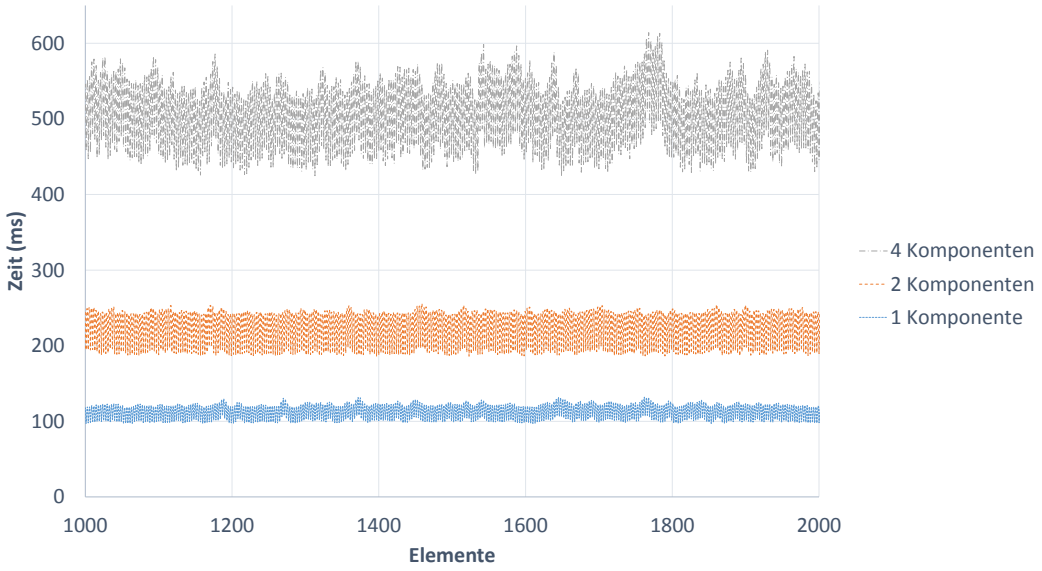


Abbildung 7.4: *Latenz des Verbundoperators für 1,2 und 4-komponentige Mischverteilungen*

Wir sehen in Abbildung 7.5 deutlich wie die Latenz der Bestimmung des Ausdrucks exponentiell mit der Anzahl an Bedingungen steigt und bei 184 Sensoren (46 Hauptsensoren und jeweils 3 Sekundarsensoren für die Umwelteigenschaften) einen für die gewöhnliche Nutzung nicht mehr vertretbare Dauer für die Generierung der Qualitätsdimensionsausdrücke von 120 Sekunden aufweist. Dieser Wert mag zunächst gering erscheinen, allerdings gilt zu bedenken, dass dies bedeutet, dass eine Anwendung dabei 46 unterschiedliche Sensoren aufweist, die jeweils von 3 Umwelteigenschaften abhängen und diese Umwelteigenschaften für jeden Sensor einzeln von einem anderen Sensor wahrgenommen werden. Die resultierende Datenstromverarbeitungsanfrage hätte dabei also 184 Quellen. In den meisten Anwendungen ist eher zu erwarten, dass diese Umwelteigenschaften von weitaus weniger Sensoren wahrgenommen werden können, also beispielsweise ein Temperatursensor, der die aktuelle Temperatur für mehrere Hauptsensoren wahrnimmt. In den betrachteten Szenarien etwa wird die zu überwachende Fläche von 3 Laserscannern überwacht, wobei unter der Annahme, dass es keine gravierenden Temperaturunterschiede zwischen den Platzierungen der Sensoren gibt, ein einziger Temperatursensor für die Bestimmung der Betriebstemperatur innerhalb der Anwendung ausreichen würde.

7.1.3 Direkte Qualitätsbestimmung

Im Folgenden werden die drei Ansätze zur direkten Qualitätsbestimmung von Elementen in einem Datenstrom evaluiert. Hierbei wird zum einen die Latenz der Operatoren aber auch die Güte der entstehenden Mischverteilungen im Bezug zu den existierenden Daten-

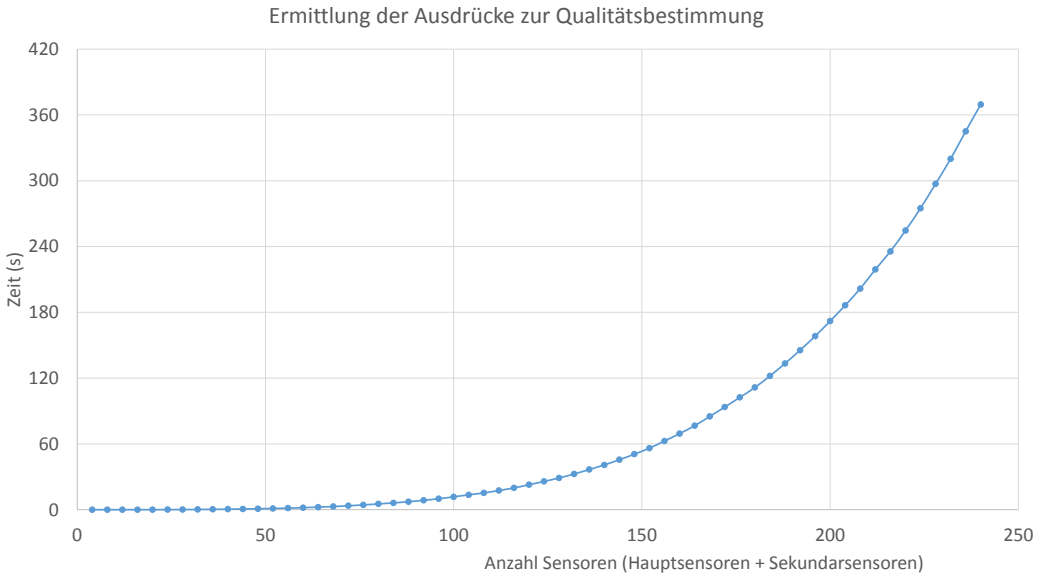


Abbildung 7.5: *Benötigte Zeit zur Ermittlung des Qualitätsdimensionsausdrucks in Abhängigkeit zur Anzahl an verwendeten Sensoren*

werten verglichen. Zu den evaluierten Ansätze zählen die in Abschnitt 4.5 beschriebenen Verfahren zur Erwartungswertmaximierung, zur Kerndichteschätzung und die Kombination aus Kerndichteschätzverfahren und Bregman-Hard Clustering zur Komprimierung der resultierenden Mischverteilungen aus dem Kerndichteschätzverfahren.

Das Erwartungswertmaximierungsverfahren versucht ein stochastisches Modell an die eingehenden Daten anzupassen. Dabei spielen neben der Datenfenstergröße die Anzahl der Iterationen, der Konvergenzschwellwert für die Veränderung der Log-Likelihood in jeder Iteration, sowie die Anzahl an Komponenten der Mischverteilungen eine Rolle für die Latenz dieses Operators. Für die Evaluation wurde der Konvergenzschwellwert auf 0.001 gesetzt, die Anzahl an Iterationen auf 30 und die Zahl der Komponenten auf 2. Die gleiche Anzahl an Iterationen wird ebenfalls in der von V. Garcia bereitgestellten Java Bibliothek jMEF[1] verwendet. Das Kerndichteschätzverfahren bestimmt für jeden Datenwert eine eigene Komponente in der resultierenden Mischverteilung. Der entwickelte Operator verwendet hierzu die Scott-Regel zur Bestimmung der Bandbreite der Kovarianzmatrix der Komponenten. Das Bregman-Hard Clustering, welches in einem weiteren Schritt verwendet wird um die Anzahl an Komponenten auf die gewünschte Zahl zu minimieren, wurde mit einer maximalen Anzahl von 30 Iterationen konfiguriert. Um die Resultate vergleichbar zu halten wurde der Operator so konfiguriert, dass er ebenfalls eine 2-komponentige Mischverteilung ermittelt – also zwei Cluster bildet. Der hier verwendete Konvergenzschwellwert für das Erwartungswertmaximierungsverfahren liegt oberhalb des, in der ver-

[1] http://vincentfpgarcia.github.io/jMEF/

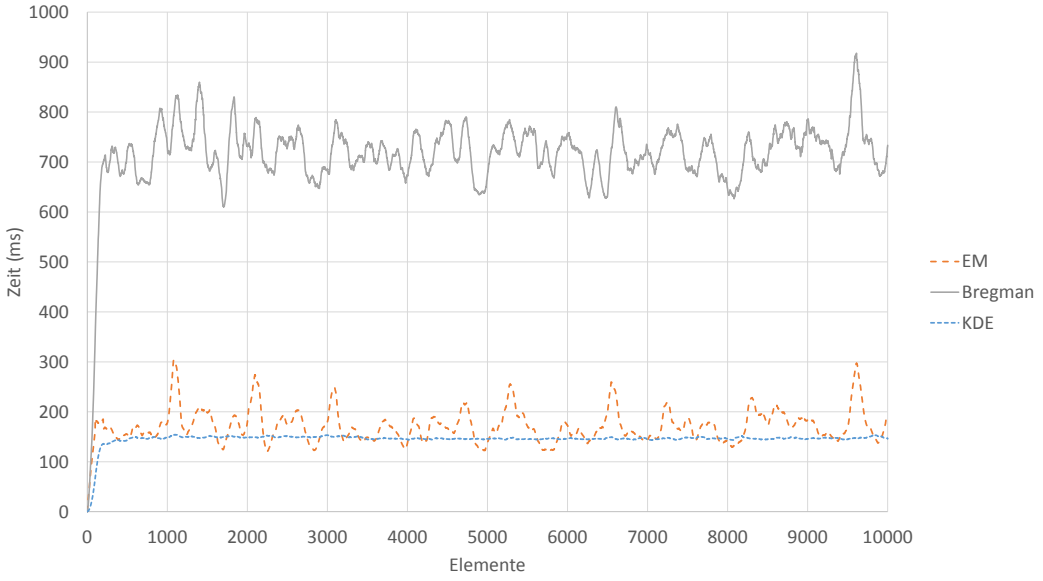


Abbildung 7.6: *Latenz der Operatoren bei einem Datenfenster der Größe 100 für Daten aus einer Normalverteilung*

wendeten Apache Commons Math3 Bibliothek[2] als Standardwert festgelegten, Wertes von 0.00001, da sich in den Versuchen zeigte, dass bereits ein höherer Konvergenzschwellwert ausreichte um die Verfahren hinsichtlich der Güte des stochastischen Modells und der gemessenen Latenz miteinander zu vergleichen.

Die Evaluation wurde dabei sowohl auf synthetischen Daten als auch auf Daten aus einem Ultrabreitband-Positionierungssystem [WJKC12], welches im Rahmen der Fallstudie Sichere Offshore-Operationen verwendet wird, durchgeführt.

7.1.3.1 Synthetische Daten

Zur Evaluation wurde die Verarbeitung auf Datenfenstern der Größe 10, 100 und 1000 angewendet. Die Datensätze wurden dabei aus einer Normalverteilung und einer logarithmischen Normalverteilungen generiert. Das Latenzverhalten der einzelnen Verfahren ist in Abbildung 7.6 und Abbildung 7.7 jeweils für Daten aus einer Normalverteilung und Daten aus einer logarithmischen Normalverteilung für ein Datenfenster der Größe 100 dargestellt. Weitere Messungen für die Fenstergrößen 10 und 1000 finden sich im Anhang dieser Arbeit.

Das Erwartungswertmaximierungsverfahren weist in beiden Fällen eine ähnliche und stabile Latenz von durchschnittlich ca. 200 Millisekunden auf. Dies ist durch die mehrmali-

[2] http://commons.apache.org/proper/commons-math

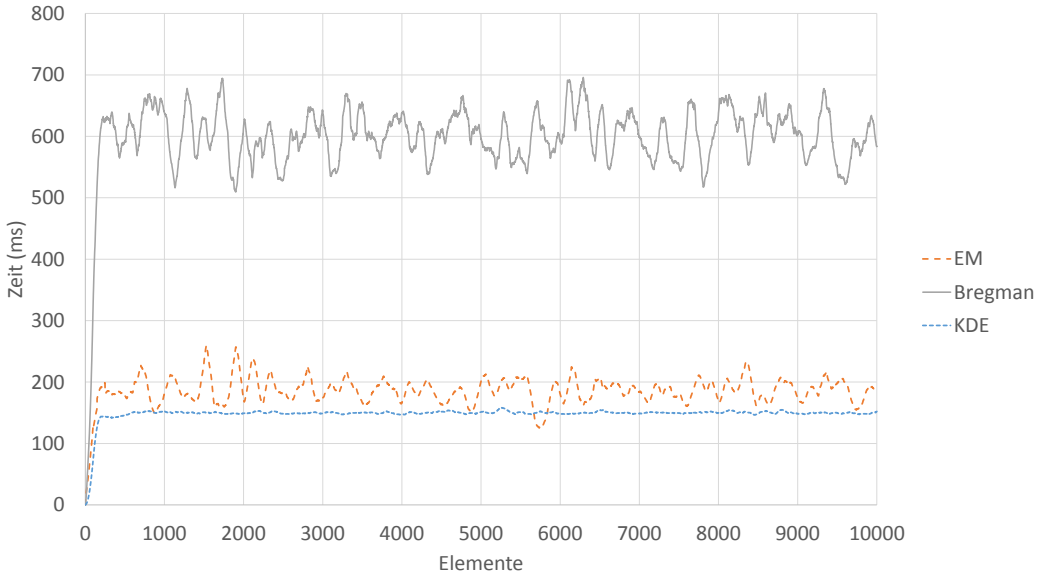


Abbildung 7.7: *Latenz der Operatoren bei einem Datenfenster der Größe 100 für Daten aus einer logarithmischen Normalverteilung*

ge Iteration über die aktuell gültigen Daten zur Bestimmung der Log-Likelihood zwischen dem jeweils temporären stochastischen Modell und den Daten geschuldet. Im Gegensatz zum Erwartungswertmaximierungsverfahren kann das Band bei der Kerndichteschätzung kontinuierlich bestimmt werden, wodurch nicht jedes Element im aktuellen Gültigkeitsfenster zusätzlich iteriert werden muss. Des Weiteren findet kein Minimierungsverfahren zur Bestimmung der Mischverteilung statt. Es fällt auf, dass trotz mehrmaliger Wiederholung der Messung das Verfahren zum Bregman-Hard Clustering eine deutlich höhere Latenz aufweist. Dieses Verhalten ist dabei unabhängig von der Art der Verteilung. Dies ist vor allem auf die Tatsache zurück zu führen, dass das Bregman-Hard Clustering Verfahren in jeder Iteration die Bregman-Divergenz zwischen den Clusterzentren und den einzelnen Komponenten bestimmen muss und zusätzlich noch den Zentroiden aus jedem Cluster in jeder Iteration neu ermitteln muss. Beim Vergleich der durchschnittlichen Latenz bei unterschiedlichen Größen von Datenfenstern zeigt sich, dass die Latenz des Erwartungswertmaximierungsverfahrens konstant bleibt, während die Latenz des Bregman-Hard Clusterings stark ansteigt. Bei der Betrachtung der Qualität des ermittelten stochastischen Modells fällt auf, dass die Qualität des Erwartungsmaximierungsverfahrens im Sinne des AIC bei Werten aus einer logarithmischen Normalverteilung deutlich besser abschneidet als das Kerndichteschätzverfahren in Kombination mit dem Bregman-Hard Clustering. Bei Werten aus einer Normalverteilung dagegen unterscheidet sich der AIC Wert nur geringfügig bei den beiden Verfahren. Ein gleiches Verhalten lässt sich auch bei Datensatzfenstern der Größe 1.000 beobachten.

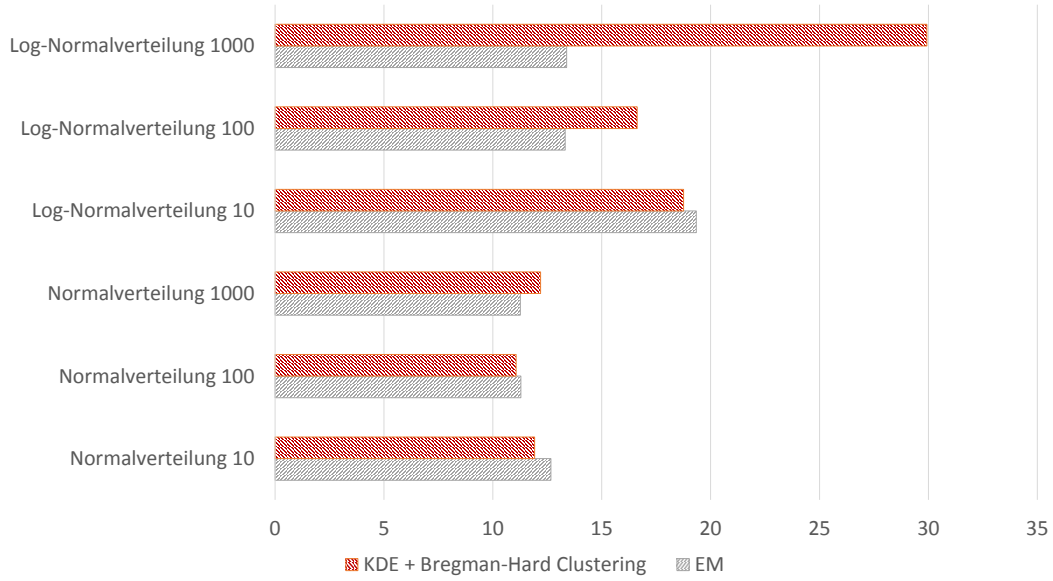


Abbildung 7.8: *Vergleich des AIC zwischen Erwartungsmaximierungsverfahren und Kerndichteschätzung mit Bregman-Hard Clustering bei unterschiedlichen Datensatzfenstergrößen für Werte aus einer Normalverteilung und einer logarithmischen Normalverteilung*

Dieses Verhalten ändert sich allerdings bei einer geringen Anzahl von Datensätzen (vgl. Abbildung 7.8). Bei einem Datensatzfenster der Größe 10 zeigt sich unabhängig von dem zugrunde liegenden stochastischen Modell der Daten, dass die Kombination aus Kerndichteschätzung und Bregman-Hard Clustering das bessere Modell liefert. Zudem unterscheiden sich die Latenzen bei dieser Datenmenge zwischen den beiden Verfahren nur gering, wie dies beispielhaft bei Daten aus einer logarithmischen Normalverteilung in Abbildung 7.9 zu sehen ist.

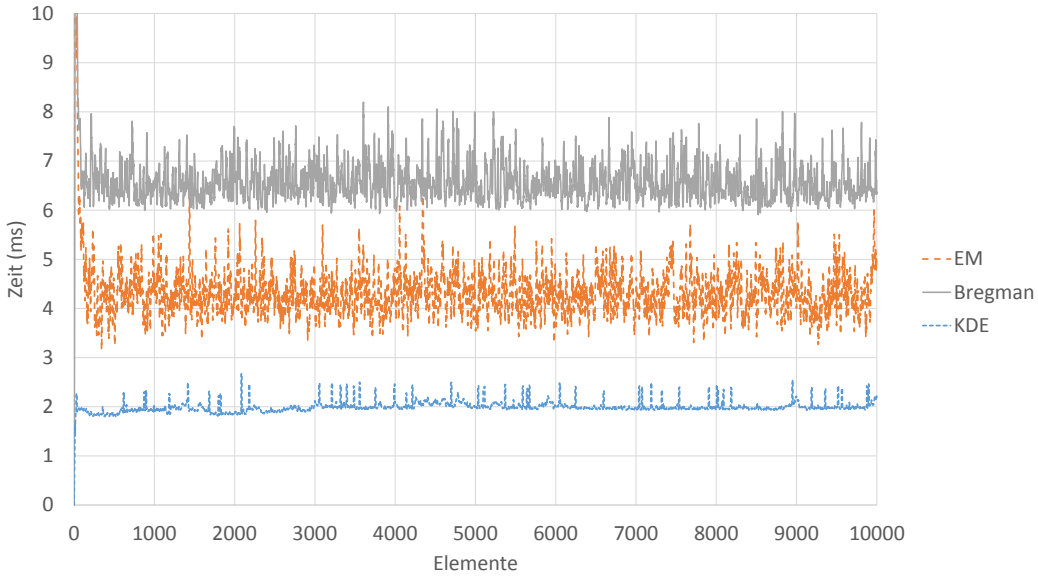
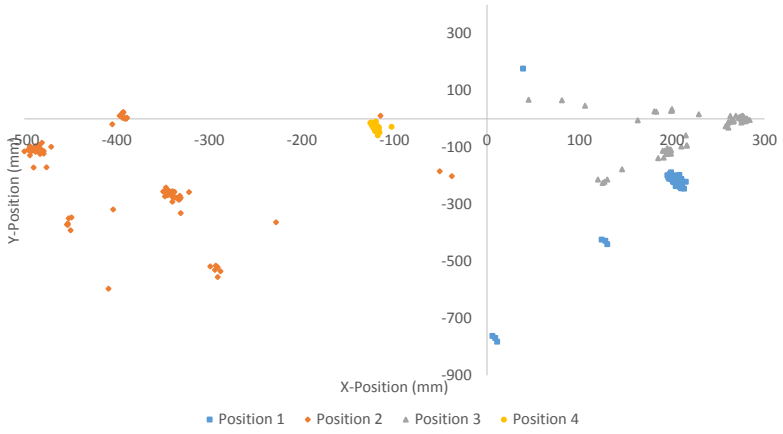


Abbildung 7.9: *Latenz der Operatoren bei einer logarithmischen Normalverteilung mit einem Datenfenster der Größe 10*



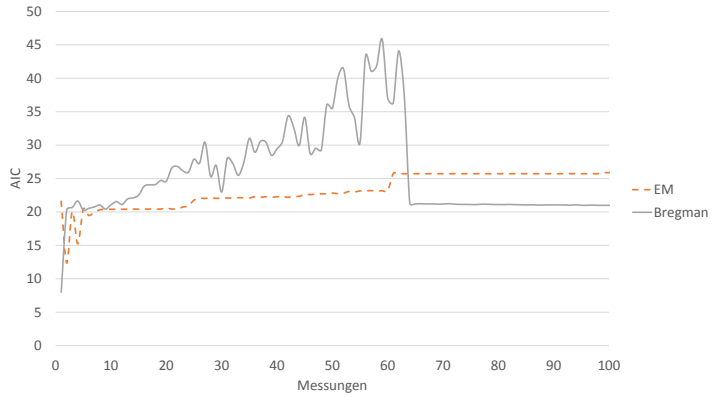
(a) Messwerte der Positionen 1–4

(b) Messwerte der Positionen 5–8

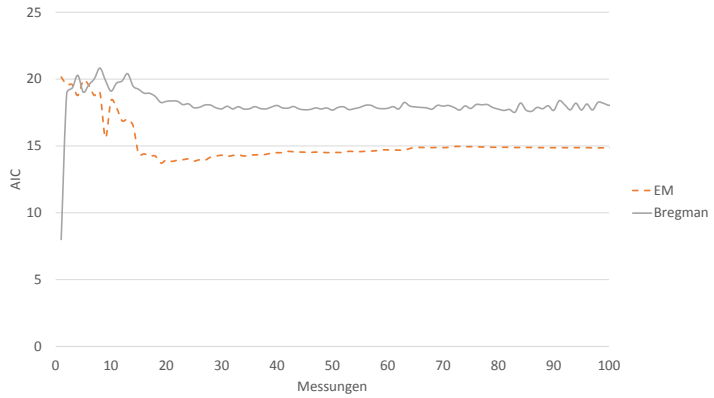
Abbildung 7.10: *Messwerte der Positionsbestimmung für die Positionen 1–8*

7.1.3.2 Reale Sensordaten

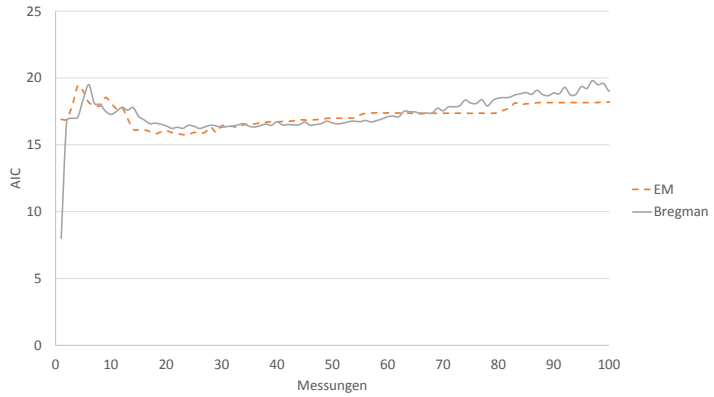
Um zu zeigen, dass die Verfahren auch stochastische Modelle von echten Sensordaten innerhalb eines DSMS ermitteln können, wurden die Operatoren auf Sensordatenaufzeichnungen eines Ultrabreitband-Positionierungssystem [WJKC12] angewendet. Insgesamt wurden 8 Positionen (vgl. Abbildung 7.10) bestimmt, von denen im Folgenden die Positionen 1, 6 und 7 als repräsentative Positionen näher betrachtet werden. Hierbei wurde das stochastische Modell jeder Position mit dem Erwartungswertmaximierungsverfahren und der Kombination aus Kerndichteschätzung und Bregman-Hard Clustering auf einem Datensatzfenster der Größe 10 und einem Datensatzfenster der Größe 100 bestimmt.



(a) Qualität des stochastischen Modells von Position 1



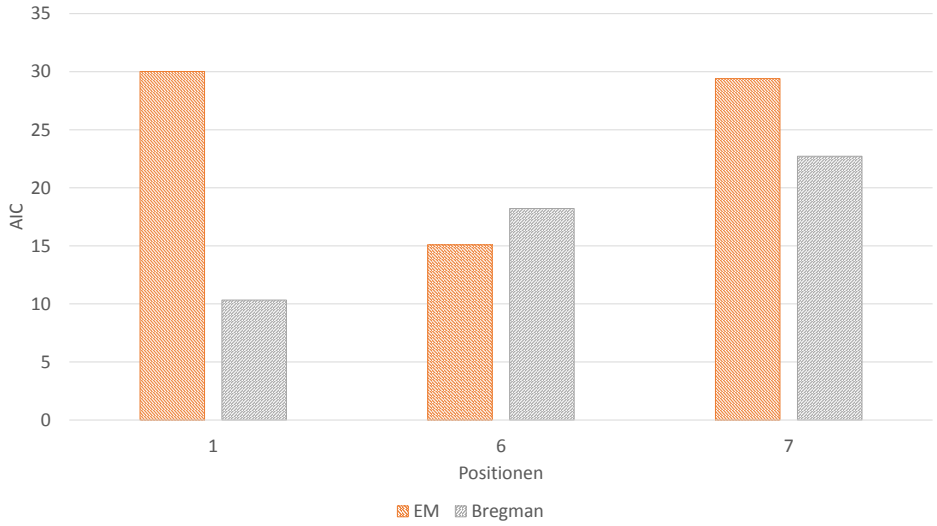
(b) Qualität des stochastischen Modells von Position 6



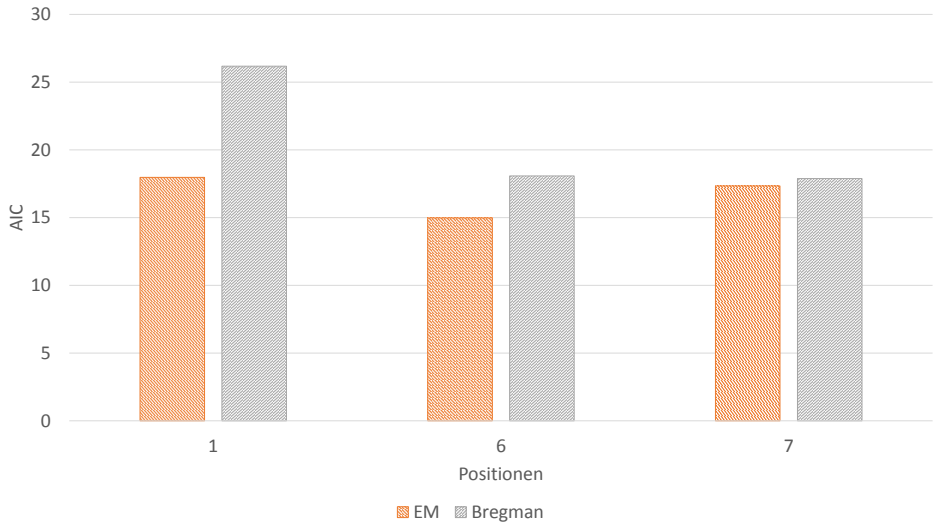
(c) Qualität des stochastischen Modells von Position 7

Abbildung 7.11: *Qualitäten der stochastischen Modelle der Positionsbestimmungen*

Bei Betrachtung der zeitlichen Bestimmung des stochastischen Modells in der Abbildung 7.11 für die Positionen 1, 6 und 7 fallen zunächst für die Positionen 1 und 6 anfängliche Ausreißer bei der Nähe zum Modell auf. Dies deutet auf eine anfängliche Anpassung der Positionierungsknoten der Anwendung hin. In den darauf folgenden Messungen bleiben sowohl die Modellqualität des Erwartungswertmaximierungsverfahrens als auch das resultierende Modell des Bregman-Hard Clustering stabil.



(a) Qualität des stochastischen Modells bei einer Datenfenstergröße von 10 Messungen



(b) Qualität des stochastischen Modells bei einer Datenfenstergröße von 100 Messungen

Abbildung 7.12: *Qualitäten der stochastischen Modelle im Sinne des AIC für die Positionen 1, 6 und 7*

Wie bereits bei den synthetischen Daten zu sehen war, ist auch bei realen Sensordaten in Abbildung 7.12a das Phänomen erkennbar, dass die Kombination aus Kerndichteschätzung mit Bregman-Hard Clustering bei kleinen Datensatzfenstern im Vergleich zum Erwartungswertmaximierungsverfahren bessere stochastische Modelle im Sinne des AIC ermittelt. Dagegen ist bei größeren Datensatzfenstern, wie in Abbildung 7.12b zu sehen ist, das Erwartungswertmaximierungsverfahren besser geeignet um gute stochastische Modell im Sinne des AIC zu bestimmen.

7.1.4 Evaluation der Funktionen zur Erstellung der probabilistischen Belegungskarte

Zur Evaluation der Funktionen zur Ausbreitung und Aktualisierung der Belegungskarten aus Kapitel 3 wurden die Daten eines Laserscanners simuliert, welcher eine leere Fläche der Größe $144\text{m} \times 144\text{m}$ überwacht. Der Laserscanner wurde in Anlehnung an einen Laserscanner für den Außenbereich vom Typ LMS 511 der Firma SICK simuliert. Der Sensor besitzt eine Winkelauflösung von 0.25°, einen Erfassungsbereich von 270° und eine maximal detektierbare Distanz von 72 Metern. Die Überwachung einer leeren Fläche bedeutet insbesondere für den Aktualisierungsoperator, dass die maximale Anzahl an Zellen verarbeitet werden müssen. Da bei Hindernissen oder Objekten in der Umgebung der Algorithmus früher abbrechen kann, bedeutet eine leere Fläche die obere Grenze des Algorithmus hinsichtlich seiner Latenz. Für die Latenzevaluation des Ausbreitungsoperators wurde jeweils die Latenz für die Verarbeitung von Belegungskarten mit Zellengrößen von 500, 250 und 100 Millimeter untersucht.

Das Laufzeitverhalten des Ausbreitungsoperators ist in Abbildung 7.13 dargestellt. Die Latenz des Ausbreitungsoperators steigt hierbei exponentiell zur Größe der zu berechnenden Zellen. Da die Latenz des Operators selbst bei einer Zellengröße von 100mm noch unterhalb von 90ms liegt, kann der Ausbreitungsoperator auch in Anwendung eingesetzt werden, in denen Sensoren mit einer höheren Frequenz verwendet werden.

Für die Latenzevaluation des Aktualisierungsoperators wurden die gleichen Werte für die Zellengröße verwendet und für die radiale Größe des Polargitters die Größen 500, 250 und 100 Millimeter gewählt. Als Streuung hinsichtlich der wahren Distanz wurde für den Sensor der Wert 4.2 Millimeter als Standardabweichung angewendet. Wichtig für die Latenz ist dabei die Anzahl an parallelen Prozessen, da jeder Strahl getrennt von den anderen Strahlen verarbeitet werden kann und nur der Zugriff auf die zu aktualisierende Zelle im kartesischen Belegungsgitter synchronisiert werden muss. Für die Evaluation wurden hierbei 9 parallele Prozesse verwendet, die jeweils einen Kreisabschnitt von 30° verarbeiteten.

Selbst bei Zellengrößen von 100×100 Millimetern kann eine Berechnung der Belegungswahrscheinlichkeit der Zellen in einem Bereich von $144\text{m} \times 144\text{m}$ unterhalb von 300 Millisekunden durchgeführt werden. Da allerdings die verwendeten Sensoren eine Fre-

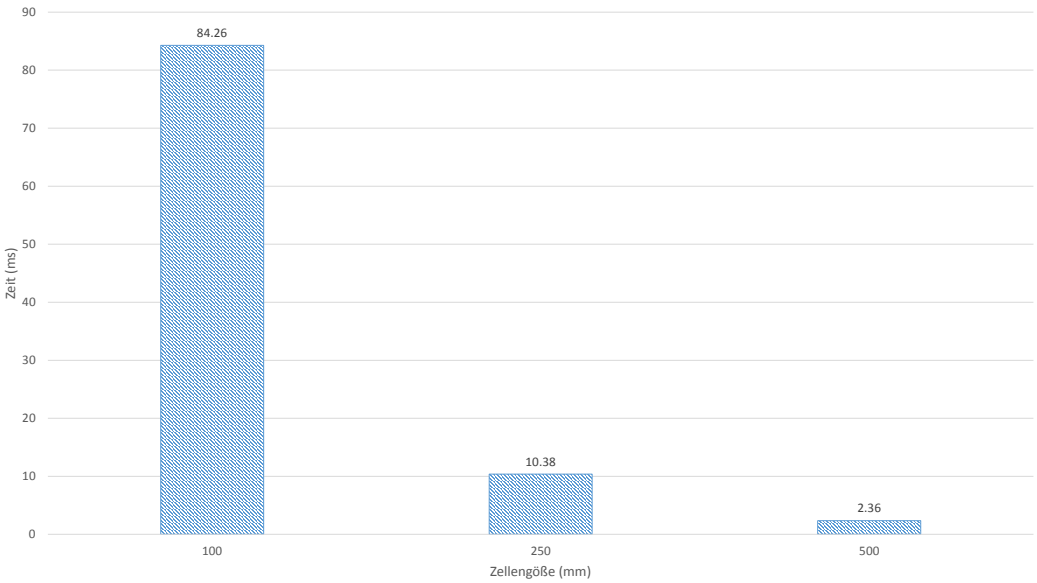


Abbildung 7.13: *Latenz des Ausbreitungsoperators für Zellen mit Kantenlängen 100mm, 250mm und 500mm*

quenz von 25Hz aufweisen und in den betrachteten Szenarien die beteiligten Akteure in Schrittgeschwindigkeit von ca. 10km/h operieren ist die Latenz bei einer Zellengröße von 100mm zwar nicht akzeptabel im Hinblick auf die Aktualisierungsfrequenz der Sensoren aber in dem betrachteten Szenario auch nicht zwingend notwendig, so dass die Auslösung bei der Belegungskarte geringer gewählt werden kann.

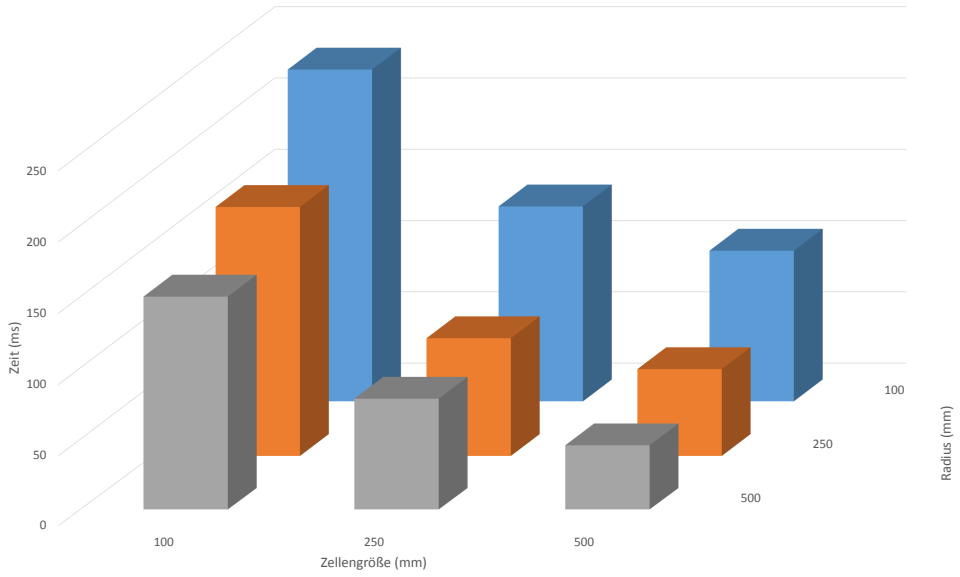


Abbildung 7.14: *Latenz des Aktualisierungsoperators in Abhängigkeit zur Zellengröße im kartesischen Gitter und zum Radius des Polargitters*

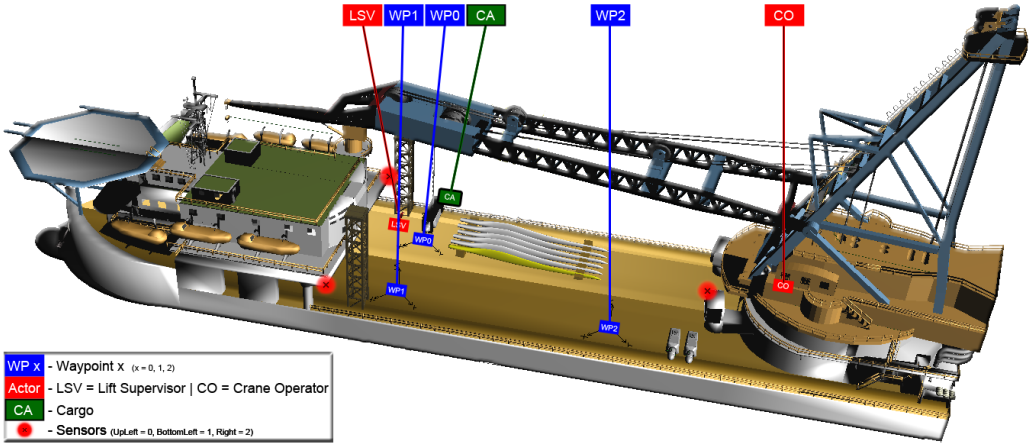


Abbildung 7.15: *Aufbau des SOOP-Szenarios*

7.2 Fallstudie: Sichere Offshore-Operationen

Um die Anwendbarkeit der entwickelten Verfahren zu evaluieren wird im Folgenden das Konzept auf das Anwendungsszenario der Sicheren Offshore-Operationen (SOOP) [SBL+12] angewendet. In diesem Szenario wird der Kranführer bei Verladeoperation von großen oder schweren Lasten durch einen Lift-Supervisor, der die Verladeoperation vom Deck aus überwacht und mit dem Kranführer kommuniziert, unterstützt. Die Aufgabe des Assistenzsystems ist es hierbei frühzeitig zu warnen wenn:

- Eine Person zu nah an der Last bzw. unter der Last steht, oder
- Der Lift-Supervisor nicht genug Abstand zur Last hält

Zur Evaluation auf realen Sensordaten wurden Sensordatenaufzeichnungen aus dem Anwendungsszenario verwendet, bei der ein Verladeszenario auf einem Schiff nachgestellt wurde. Hierbei wurde die Umgebung mit 3 Laserscannern erfasst und die Position der Personen mit Ultrabreitband-Positionssensoren [WJKC12, JBS+13] erfasst. Die Anordnung der Laserscanner und der beteiligten Personen ist in Abbildung 7.15 dargestellt. Die drei Laserscanner sind so angeordnet, dass sie die komplette Verladefläche erfassen können. Während der Aufzeichnung folgt der Lift-Supervisor (LSV) den blau eingezeichneten Wegpunkten. Die Position der Last wird in der folgenden Evaluation als ein fester Bereich auf der Verladefläche angesehen. Da das verwendete Schiff in dem Verladeszenario nur ein kleines Deck besitzt, das Anwendungsgebiet aber Operation wie etwa Containerverladung und Montage von Offshore-Windkraftanlagen betrachtet, wird im Folgenden daher eine Fläche von 20cm $\times$ 20cm als kritischer Bereich verwendet. Ziel der Verarbeitung ist die Erkennung von zwei kritischen Situationen, welche im Folgenden als Hazard bezeichnet

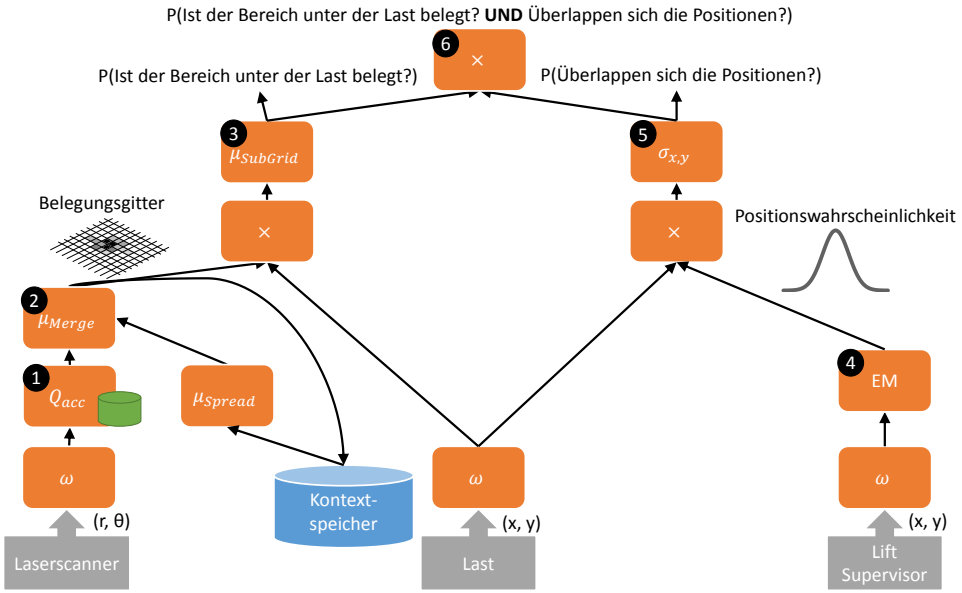


Abbildung 7.16: *Logischer Anfrageplan zur Erstellung des dynamischen Kontextmodells für die Fallstudie Sichere Offshore-Operationen*

werden. Der erste Hazard beschreibt die Situation, dass sich etwas unterhalb der schwebenden Last befindet. Dabei ist es nicht notwendig ein bestimmtes Objekt zu detektieren, sondern lediglich die Frage zu beantworten ob sich etwas im kritischen Bereich befindet. Der zweite Hazard beschreibt die Situation, dass sich der Lift-Supervisor unterhalb der Last befindet. In diesem Fall handelt es sich um ein konkretes Objekt, welches durch den vom Lift-Supervisor getragenen Positionssensor ermittelt wird.

Die, für die Erkennung von kritischen Situationen, entwickelte Anfrage für die Sensordatenverarbeitung in einem DSMS ist in Abbildung 7.16 als Graph dargestellt. Zunächst wird in Punkt 1 das Rauschen der Laserscanner durch die Informationen aus der Ontologie mit Hilfe des Qualitätsoperators (Q_{acc}) aus Kapitel 4.4 bestimmt. Hierzu ist innerhalb der Ontologie eine Standardabweichung von 4.2mm als Qualität für den Sensor bei einer Betriebsumgebungstemperatur von 0° – 50℃ definiert. Der um die Standardabweichung angereicherte Datenstrom wird anschließend an den nächsten Operator zur Ermittlung der Existenzwahrscheinlichkeit von Objekten in der überwachten Umgebung weitergeleitet. Hierbei wird über die entwickelten Ausbreitungsfunktion (μ_{Spread}) und Aktualisierungsfunktion (μ_{Merge}) aus Kapitel 3 ein dynamisches Kontextmodell in Form einer Belegungskarte durch die Laserscanner an Punkt 2 im Graphen erstellt. Anschließend wird an Punkt 3 die Belegungskarte mit der aktuellen Position der Last kombiniert und an der Position der Last ein Teil der Belegungskarte durch den Abbildungsoperator ($\mu_{SubGrid}$) herausgeschnitten und die maximale Belegungswahrscheinlichkeit aller Zellen in dieser

lokalen Belegungskarte ermittelt. Dieses Ergebnis entspricht der Wahrscheinlichkeit, dass der erste Hazard eingetreten ist.

Parallel dazu wird die Qualität der Positionsmessungen des Positionierungssystems kontinuierlich durch die direkte Qualitätsbestimmung aus den Messungen des Positionssensors an Punkt 4 im Anfragegraphen ermittelt. Hierbei kommt der entwickelte Operator (EM) zur Erwartungswertmaximierung aus Kapitel 4.5 zum Einsatz. Das so ermittelte stochastische Modell wird daraufhin an den Folgeoperator weitergeleitet. Dieser Operator verknüpft die eingehenden Daten mit der aktuellen Position der Last. Das so kombinierte Tupel wird daraufhin in Punkt 5 durch die Anwendung des Selektionsoperators gefiltert. Das Selektionskriterium lautet hierbei:

$$\text{Cargo.x} - 10 \leq \text{LiftSupervisor.x} \leq \text{Cargo.x} + 10 \text{ AND}$$
$$\text{Cargo.y} - 10 \leq \text{LiftSupervisor.y} \leq \text{Cargo.y} + 10$$

Es wird also geprüft, ob sich der Lift-Supervisor innerhalb der 20cm $\times$ 20cm Fläche der Last aufhält. Da es sich hierbei um die Anwendung eines Selektionsoperators auf ein stochastisches Modell handelt, wird das Integral über das stochastische Modell innerhalb der Schranken, welche durch die Fläche der Last festgelegt werden, gebildet. Das resultierende Tupel weist also nach dieser Operation eine Existenzwahrscheinlichkeit auf, die sich aus der Fläche der integrierten Verteilung ergibt. Dieses Ergebnis entspricht somit der Wahrscheinlichkeit, dass der zweite Hazard eingetreten ist.

Auf Basis der beiden Ergebnisse aus den jeweiligen Verarbeitungspfaden wird zudem eine weitere Ausgabe durch den Verbundoperator in Punkt 6 berechnet. Die dritte Ausgabe kann dabei als Konfidenz der Verarbeitung angesehen werden und entspricht der Verknüpfung der beiden Ausgaben und ihrer Existenzwahrscheinlichkeit. Die Aussage eines Stromelements aus dieser Ausgabe beantwortet die Frage, zu welchem Grad die Fläche unter der Last frei ist und die Position des Lift-Supervisors und der Last sich überlappen. Die konkrete Verarbeitungsanfrage in Form einer PQL-Anfrage für das Odysseus DSMS findet sich im Anhang dieser Arbeit.

7.2.1 Konfiguration der Evaluationsoperatoren

In dem betrachteten Szenario wurde eine Belegungskarte der Größe 20m$\times$20m überwacht. Als Zellengröße wurde eine 100mm $\times$ 100mm Fläche verwendet. Die Geschwindigkeit mit der sich Objekte innerhalb dieses Szenarios fortbewegten wurde mit 10km/h festgesetzt. Der Radius für das Polargitter, welches innerhalb des Aktualisierungsoperators verwendet wird, hatte eine Länge von 25mm. Zur Bestimmung des stochastischen Modells wurde eine Datenfenstergröße von 10 Messungen verwendet. Für den kritischen Bereich der Last wurde aufgrund der Größe des Decks des Schiffes in dem evaluierten Szenario eine Fläche von 20cm $\times$ 20cm festgelegt. Ein Schnappschuss der Belegungskarte, wie sie von dem Aktualisierungsoperator auf Basis der Laserscanner erstellt wird, sehen wir in Abbildung 7.17. Hierbei ist deutlich zu sehen, wie die Messungen aller drei Sensoren zu einem

Abbildung 7.17: *Generierte Belegungskarte in der Evaluation der Fallstudie Sichere Offshore-Operationen*

dynamischen Kontextmodell vereinigt werden. Auch ist zu erkennen, wie die Belegungswahrscheinlichkeit insbesondere an den Rändern, aber auch um Objekte herum in Form von Grauwerten anwächst.

7.2.2 Ergebnisse der Evaluation

Die Evaluation startete zunächst mit dem Lift-Supervisor innerhalb der kritischen Region, aus der der Lift-Supervisor im Laufe der Messungen heraustritt. In Abbildung 7.18 sehen wir hierzu zunächst die Existenzwahrscheinlichkeit einer detektierten kritischen Situation durch die probabilistische Verarbeitung der Ultrabreitbandsensoren. Zu Beginn der Messung liegt die Existenzwahrscheinlichkeit für den Hazard 2 bei ca. 80% und steigt zum Zeitpunkt 52:20 auf 100%. Dies bedeutet, dass das anfängliche stochastische Modell noch eine zu hohe Streuung aufwies, welche durch die zusätzlichen Messungen präzisiert werden konnte und so die komplette Wahrscheinlichkeitsmasse des stochastischen Modells innerhalb der kritischen Fläche liegt.

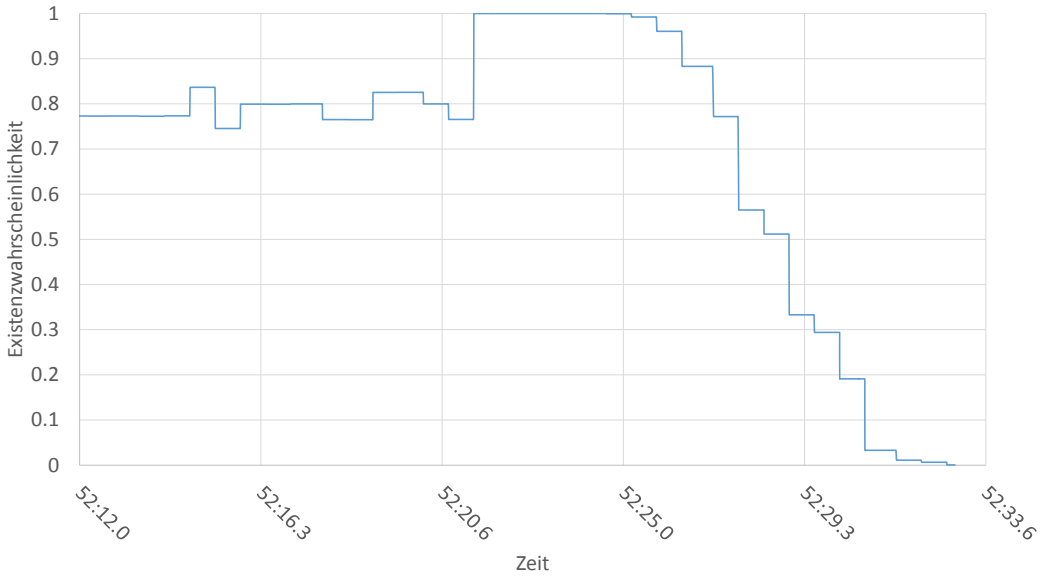


Abbildung 7.18: *Existenzwahrscheinlichkeit der Positionsverarbeitung zur Detektion von Hazard 1 auf Basis der Ultrabreitband-Sensoren*

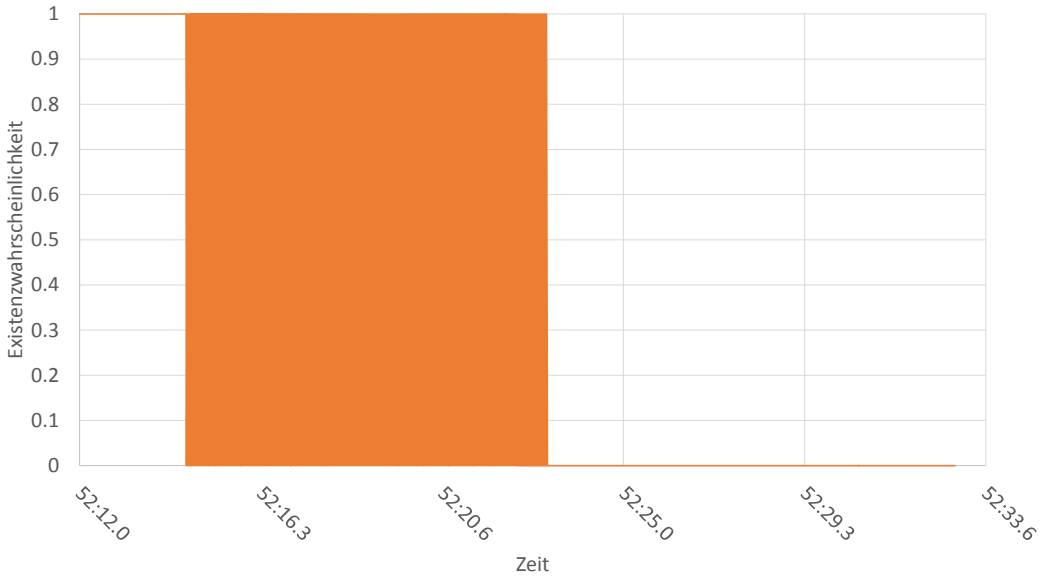


Abbildung 7.19: *Existenzwahrscheinlichkeit der Detektion von Hazard 2 auf Basis der Belegungskarte*

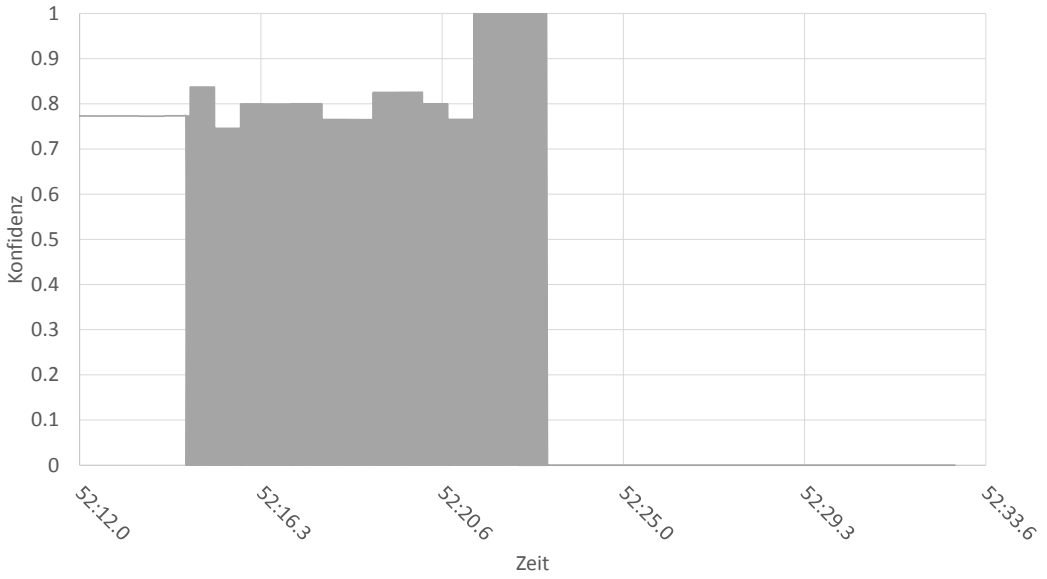


Abbildung 7.20: *Konfidenz der Verarbeitungsergebnisse aus den Anfragen zur Detektion von Hazard 1 und 2*

Bei der Verarbeitung durch die Belegungskarte sieht man in Abbildung 7.19 wie bereits zum Beginn die Existenzwahrscheinlichkeit von Hazard 1 fast 100% beträgt und anschließend die Wahrscheinlichkeit zwischen drei Werten oszilliert. Dieser Effekt tritt aus zwei Gründen auf. Zum einen ist die Streuung der Messwerte der Laserscanner sehr gering, so dass lediglich eine Nachbarzelle neben einem Objekt einen Wert zwischen „belegt“ und „frei“ annimmt. Zum anderen oszillieren die Belegungen von zwei gegenüberliegenden Zellen zusätzlich durch die Quantisierung der Fläche und der nicht absolut exakten Ausrichtung der Laserscanner bei Übergängen des Lift Supervisors zwischen zwei Zellen.

In Abbildung 7.20 ist das Ergebnis der dritten Ausgabe zu sehen. Die Ausgabe spiegelt die Konfidenz der Existenz beider Hazards wieder. Hier zeigt sich, dass obwohl der Positionssensor ein hohes Rauschen aufweist, die Existenzwahrscheinlichkeit der Ergebnisse aus der Erkennung von Hazard 1 die Konfidenz der Ergebnisse aus beiden Verarbeitungspfaden auf den Wert 0 herunter zieht. Hier zeigt sich ein Problem der direkten Qualitätsbestimmung. Zwar ist es möglich das stochastische Modell auf Basis der Messwerte anzunähern, allerdings werden besonders bei beweglichen Objekten mehr Messwerte benötigt, die das Objekt in einem nahe Stillstand erfassen. Dies war in dieser Fallstudie nicht möglich, weshalb das Ende einer kritischen Situation erst zu einem späteren Zeitpunkt festgestellt werden konnte. Im Umkehrschluss bedeutet dies auch, dass eine kritische Situation auf Basis der Messungen in diesem Anwendungsszenario zu spät erfolgen würde. Eine zeitnahe Erkennung benötigt eine höhere Messfrequenz der verwendeten Sensorik. Dennoch entsteht durch den Verbund beider Verarbeitungspfade eine Sicht von oben und

von unten auf die kritische Situation und erlaubt es daher einer Anwendung zu entscheiden, ob es die Erkennung von Hazard 2 weiter verarbeiten oder die Erkennung verwerfen möchte.

Dies führt zu den drei Problemen bisher existierender Arbeiten, welche zu Beginn dieser Arbeit aufgezeigt wurden:

1. Probleme der Flexibilität von sensordatenverarbeitenden Systemen. Viele Systeme die in den betrachteten Anwendungen verwendet werden, können nur mit einer Klasse von Sensoren oder sogar nur mit Sensoren eines einzigen Herstellers arbeiten.

2. Probleme der Integration von Sensordatenqualitäten. Die Systeme betrachten lediglich den Messwert eines Sensors, jedoch nicht die Qualität der Messung, wenn diese nicht direkt von dem Sensor selbst geliefert wird.

3. Probleme der Verarbeitung von Qualitätsinformationen. Selbst eine nachträgliche Integration von Qualitätsinformationen in ein System bedeutet noch nicht, dass diese Qualitätsinformationen bei der Verarbeitung berücksichtigt und durch das komplette System korrekt propagiert werden.

Am Beispiel der Fallstudie Sichere Offshore-Operationen wurde gezeigt, wie eine qualitätssensitive Verarbeitung von Sensormesswerten zur Erstellung eines dynamischen Kontextmodells stattfinden kann. Hierzu wurden die entwickelten Ansätze zur Lösung der drei Teilprobleme wie folgt aufgezeigt:

1. Das Problem der Flexibilität von sensorverarbeitenden Systemen wurde durch den Einsatz eines Datenstrommanagementsystems zur kontinuierlichen Verarbeitung von Sensormessungen gelöst. Durch die Nutzung einer Anfragesprache zur Beschreibung der Verarbeitung, statt einer fest implementierten Verarbeitung von Sensormesswerten, kann eine sehr flexible und komplexe Verarbeitung von Sensormesswerten stattfinden.

2. Die Integration von Sensorqualitäten wurde auf zwei Arten gelöst. Zum einen werden die Ungenauigkeiten des Ultrabreitband-Positionierungssystems durch die Bestimmung des stochastischen Modells mit den zusätzlichen Operatoren behandeln. Zum anderen werden das Rauschen der Laserscanner durch die verwendete Ontologie bestimmt und innerhalb der Operatoren zur Aktualisierung der Belegungskarte berücksichtigt.

3. Das Problem der Verarbeitung von Qualitätsinformationen wurde durch die Erweiterung der temporalen relationalen Algebra gelöst, in der nun alle Operatoren des Systems die Qualitätsinformationen, zusätzlich zu dem Gültigkeitsintervall des relationalen Tupels, in Form von multivariaten Mischverteilungen in den Metadaten tragen. Dabei werden von allen Operatoren die Qualitätsinformationen bei der Verarbeitung in den zu erzeugenden Ausgabestrom überführt. Hierdurch ist es nun auch möglich die

Konfidenz von mehreren Verarbeitungspfaden durch den Verbundoperator zu ermitteln.

7.3 Zusammenfassung

In diesem Kapitel wurden zunächst die entwickelten Konzepte für sich genommen hinsichtlich ihrer Latenz evaluiert und gezeigt, dass sie sich grundsätzlich zur Verarbeitung von hochfrequenten Sensordaten eignen. Allerdings zeigten sich auch einige Einschränkungen, so können Filteroperationen durch die Notwendigkeit der Integration von multivariaten Mischverteilungen die Latenz der Verarbeitung erheblich erhöhen. Das gleiche gilt für den Verbundoperator, bei dem zwei Datenströme durch ein Verbundkriterium zu einem neuen Datenstrom verbunden werden. Hierbei stieg die Latenz exponentiell mit der Anzahl an Komponenten in den zu verarbeiteten Mischverteilungen. Andere Operatoren, wie die Projektion, können dagegen ohne zusätzliche Kosten für die Verarbeitung verwendet werden, da sich ihrer Latenzverhalten nur geringfügig von den bisherigen Operatoren in einem DSMS unterscheiden .

Des Weiteren wurden das Konzept zur indirekten Qualitätsbestimmung und die unterschiedlichen Konzepte zur direkten Qualitätsbestimmung evaluiert. Bei der indirekten Qualitätsbestimmung wurde die Anzahl an Sensoren gemessen, die zur Ermittlung des Qualitätsausdrucks in der Ontologie miteinander verknüpft werden können. Die Qualitätsindikatoren Flüchtigkeit, Vollständigkeit und Konsistenz aus Kapitel 4.4 wurden im Rahmen der Fallstudie nicht betrachtet, da sie für die Bestimmung der kritischen Situation nicht notwendig waren. Die einzelnen Funktionen zur Verarbeitung der Qualitätsindikatoren stehen aber in der Implementierung zur Verfügung. Bei der direkten Qualitätsbestimmung wurde die Güte der entstehenden Mischverteilungen aus den unterschiedlichen Verfahren verglichen. Hierbei zeigte sich, dass bei kleinen Datensätzen das Kerndichteschätzverfahren in Kombination mit dem Bregman-Hard Clustering das bessere Ergebnis hinsichtlich der Metrik aufweist, die Erwartungsmaximierungsmethode wies dagegen bei einer hohen Datenmenge bessere Ergebnisse auf.

Anschließend wurden die entwickelten Ansätze in dem Anwendungsszenario Sichere Offshore-Operation evaluiert. Dabei wurden zwei kritische Situationen getrennt voneinander verarbeitet, wobei der erste Verarbeitungspfad auf Basis von Daten aus 3 Laserscannern eine Belegungskarte der Umgebung erstellt und meldet sobald ein Objekt sich in einem kritischen Bereich befindet. Hierzu wurden die Konzepte des dynamischen Kontextmodells aus Kapitel 3 mit der indirekten Qualitätsbestimmung aus Kapitel 4.4 verknüpft. Im zweiten Verarbeitungspfad wurde die Qualität von Positionsmessungen eines Ultrabreitband-Positionssystem durch die Verwendung der in Kapitel 4.5 entwickelten direkten Qualitätsbestimmung ermittelt und durch den erweiterten temporalen relationalen Verbundoperator mit der Position der Last verknüpft. Hierbei wurde auch gezeigt, wie Widersprüche in der Verarbeitung einer Applikation mitgeteilt werden können, indem

die Ergebnisse der zwei Verarbeitungspfade über einen weiteren Verbundoperator verknüpft werden können. Das Ergebnis dieser Verknüpfung enthielt dabei die gemeinsame Wahrscheinlichkeit der beiden Teilanfragen. Dabei zeigte sich allerdings auch, dass die Frequenz der Positionssensoren für die Anwendung zu gering ist. Hierdurch wurde eine kritische Situation zu spät erkannt bzw. das Ende einer kritischen Situation zu spät ermittelt. Dies zeigte sich insbesondere bei dem Verbund der beiden Verarbeitungspfade. Obwohl die ermittelte kritische Situation durch die Laserscanner bereits beendet war, lag die Existenzwahrscheinlichkeit einer kritischen Situation durch die Verarbeitung der Positionssensoren noch bei 100% lag.

Am Beispiel der Fallstudie Sichere Offshore-Operationen wurde somit gezeigt, wie die drei Teilprobleme der Forschungsfrage gelöst werden konnten. Indem zum einen ein Datenstrommanagementsystem zur kontinuierlichen Verarbeitung von Sensormessungen genutzt wurde, bei dem statt einer fest implementierten Verarbeitung eine Anfragesprache zur flexiblen Beschreibung der Verarbeitung eingesetzt wird. Zur Integration von Qualitätsinformationen in der Verarbeitung wurden gleich zwei Möglichkeiten vorgestellt. Zum einen findet eine indirekte durch eine, in einer Ontologie abgelegten, Relation zwischen Sensoren und Qualitäten statt. Zum anderen findet eine direkte Qualitätsbestimmung durch die Ermittlung des, den Sensordaten unterliegenden, stochastischen Modells statt. Das dritte Teilproblem der ganzheitlichen Verarbeitung von Qualitätsinformationen wurde durch die Erweiterung der temporalen relationalen Operatoren gelöst, so dass die zuvor bestimmten Qualitäten auch in jedem Operator des Systems berücksichtigt und korrekt bis zu einer Anwendung propagiert werden. Hierdurch ist es nun auch möglich die Konfidenz von mehreren Verarbeitungspfaden durch den Verbundoperator zu ermitteln.

8 Zusammenfassung und Ausblick

In diesem Kapitel werden abschließend die einzelnen Forschungsergebnisse dieser Arbeit zusammengefasst und bewertet. Anschließend folgt ein Ausblick auf weitere Einsatzszenarien und Erweiterungsmöglichkeiten der entwickelten Ansätze.

8.1 Zusammenfassung

Auf Basis der beschriebenen Anwendungsszenarien wurden die Probleme der flexiblen Sensordatenverarbeitung, sowie der Integration von Qualitätsinformationen und der korrekten Verarbeitung von qualitätsbehafteten Daten aufgezeigt. Durch die Verwendung eines Datenstrommanagementsystems als flexible Sensorfusionseinheit wurde bereits zu Beginn dieser Arbeit gezeigt, wie das erste Teilproblem gelöst werden könnte. Jedoch konnte so noch keine qualitätssensitive Sensorverarbeitung erzielt werden, da die existierenden Systeme eine solche Verarbeitung nicht unterstützen. Dies führte in Folge dessen zu der Forschungsfrage, die in dieser Arbeitet zu beantworten war:

Wie kann eine qualitätssensitive Verarbeitung von Sensordaten in einem Datenstrommanagementsystem in einer allgemeinen und weitestgehend automatisierten Weise durchgeführt werden?

Auf die wesentlichen Punkte zur Beantwortung dieser Frage wird im Folgenden eingegangen.

Dynamische Kontextmodelle

Zunächst wurde die grundlegende Idee eines dynamischen Kontextmodells, welches einer Anwendung alle notwendigen Informationen für die Erfüllung ihrer Aufgabe bereitstellt, erörtert. Zu diesem Zweck wurde der Begriff des Kontextes analysiert und existierende Ansätze, Architekturen und Systeme aufgezeigt. Anschließend wurde das Kontextmodell in die drei Kontextebenen, Signalebene, Merkmalsebene und Objektebene unterteilt und erläutert, welche Funktion die einzelnen Ebenen erfüllen und wie die darin enthaltenen Kontextinformationen charakterisiert werden können. In Folge dessen wurden die beiden Anwendungsszenarien Fahrerlose Transportsysteme und Sichere Offshore-Operationen für dynamische Kontextmodelle näher beleuchtet und die, in diesen Anwendungen notwendigen, Kontextinformationen aufgelistet. Aufbauend darauf wurden exemplarisch am Beispiel einer probabilistischen Belegungskarte im Kontext von fahrerlosen Transportsystemen und am Beispiel der Distanzbestimmung zwischen Mitarbeitern und schwebenden Lasten eines Assistenzsystems für Verladeoperation gezeigt, wie ein solches dynamisches Kontextmodell innerhalb eines Datenstrommanagementsystems mit Hilfe der dortigen temporalen relationalen Operatoren und die hierfür entwickelte Abbildungsfunktionen als Anfrage hinterlegt und berechnet werden kann. Hier wurde davon ausgegangen,

dass die Qualität der Sensorwahrnehmungen zuvor bekannt ist und sich zur Laufzeit auch konstant verhält. Dies ist bei realen Anwendungen allerdings nicht immer der Fall, so dass zunächst eine Qualitätsbestimmung stattfinden muss.

Qualitätsbestimmung in Datenströmen

Zur Bestimmung der Qualität von Sensorwahrnehmungen wurden zunächst die verwendeten Qualitätsdimensionen und Qualitätsklassen aus existierenden Arbeiten gesammelt, sowie ihre Definition aufgezeigt und verglichen. Zur Modellierung von Sensoren, ihren Messmöglichkeiten und deren Qualitäten, sowie die Bedingungen unter welchen diese Qualitäten gültig sind, wurde eine Ontologie eingesetzt. Die Ontologie diente dazu, die für eine Qualitätsbestimmung notwendigen Sensoren zu bestimmen und Anfragepläne zur Bestimmung der Qualität der Sensorwahrnehmungen zu generieren. Konkret wurde hierzu die Semantic-Sensor-Network-Ontologie (SSN) [CBB+12] verwendet, die bereits eine hohe Verbreitung in anderen Projekten genießt.

Da allerdings die reine Modellierung von Beziehungen und Qualitäten in einer Ontologie nicht ausreicht um die aktuelle Genauigkeit einer Sensorwahrnehmung zu beschreiben, wurde zudem aufgezeigt, wie direkt auf Basis der Messwerte deren Qualität bestimmt werden kann. Zu diesem Zweck wurden Verfahren vorgestellt, welche die Möglichkeit bietet, ein stochastisches Modell in Form von Mischverteilungen an die aktuellen Daten anzunähern und so die Qualität der Messwerte in Form des statistischen Fehlers zu beschreiben. Zur Anreicherung von Daten in einem Datenstrommanagementsystem wurden hierzu spezielle Operator definiert, welche einen Datenstrom durch Qualitätsinformationen über die enthaltenen Elemente erweitern. Die vorgestellten Operatoren wurden dabei in logische und physische Operatoren aufgeteilt um eine Trennung der Operatorsemantik und der konkreten Verarbeitung zu erzielen.

Qualitätssensitive Datenstromverarbeitung

Aufbauend auf der Anreicherung von Sensorwahrnehmungen wurden die Operatoren der temporalen relationalen Algebra innerhalb des Datenstrommanagementsystems dahingehend erweitert, damit diese in der Lage sind qualitätsbehaftete Daten zu verarbeiten. Zur Abbildung von Qualitätsinformationen wurde hierzu das Datenmodell von [Krä07] um das Mischtyp-Modell [TPD+12] erweitert. Dies erlaubt es nun zusätzlich zu den deterministischen Daten auch Ungenauigkeiten der Werte in Form von Mischtyp-Verteilungen darzustellen und mittels der definierten Operatoren zu verarbeiten. Hierzu wurde aufgezeigt, wie die Informationen innerhalb eines Stromelementes dargestellt und zwischen Operatoren ausgetauscht werden können, sowie die Semantik der logischen Operatoren definiert. In einem zweiten Schritt wurde gezeigt, wie eine mögliche physische Realisierung dieser logischen Operatoren implementiert werden kann. Hierbei wurde allerdings darauf geachtet, dass diese Erweiterung eines Datenstrommanagementsystems keinen Einfluss auf die Anfragesprache solcher Systeme stellt. Somit können also Anfragen auf determinis-

tische Daten direkt auf probabilistische Daten portiert werden ohne Änderungen an der Anfragesprache bzw. der entsprechenden Anwendung zu tätigen.

Implementierung

Die entwickelten Konzepte wurden in dem Datenstrommanagementsystem Odysseus prototypisch umgesetzt. Die entwickelten Komponenten für die Verarbeitung von Existenzwahrscheinlichkeiten und der indirekten Bestimmung von Qualitäten durch eine Ontologie sind dabei getrennt voneinander und erlauben so eine höhere Flexibilität in möglichen Anwendungen. Des Weiteren sind die Implementierungen der beiden Konzepte jeweils in Client- und Server-Implementierung aufgeteilt um sie so auch getrennt voneinander in verteilten Umgebungen zu verwenden. Die Umsetzbarkeit des Ansatzes zur qualitätssensitiven Verarbeitung von Datenströmen wurde dabei in den beiden Forschungsprojekten SALSA [TEK+12] und SOOP [SBL+12] gezeigt. Des Weiteren wurde die Verarbeitung und die Visualisierung des dynamischen Kontextmodells auf der 6ten ACM Distributed Event-Based Systems Konferenz [KGS+12] mit Konferenzteilnehmern erprobt und die direkte Qualitätsbestimmung zusammen mit der probabilistischen Verarbeitung auf der 8ten ACM Distributed Event-Based Systems Konferenz [KN14a] den Teilnehmern demonstriert.

Evaluation

Zum Beweis der Machbarkeit wurden die entwickelten Konzepte für sich genommen hinsichtlich ihrer Latenz evaluiert und gezeigt, dass sie sich grundsätzlich zur Verarbeitung von hochfrequenten Sensordaten eignen. Des Weiteren wurden das Konzept zur indirekten Qualitätsbestimmung und die unterschiedlichen integrierten Verfahren zur direkten Qualitätsbestimmung evaluiert. Bei der indirekten Qualitätsbestimmung wurde die Anzahl an Sensoren gemessen, die zur Ermittlung des Qualitätsausdrucks in der Ontologie miteinander verknüpft werden können. Hierbei zeigte sich, dass die Anzahl an Sensoren innerhalb der Ontologie stark begrenzt ist, aber für die hier betrachteten Anwendungen ausreichend ist. Bei der direkten Qualitätsbestimmung wurde die Güte der entstehenden Mischverteilungen aus den unterschiedlichen Verfahren verglichen. Dabei stellte sich heraus, dass das Erwartungswertmaximierungsverfahren in den meisten der betrachteten Verarbeitungskonfigurationen die geringere Latenz und das bessere stochastische Modell lieferte.

Anschließend wurden die entwickelten Ansätze in der Fallstudie Sichere Offshore-Operation evaluiert. Dabei wurden zwei kritische Situationen getrennt voneinander verarbeitet. Im ersten Verarbeitungspfad wurde auf Basis von Daten von mehreren Laserscannern eine Belegungskarte der Umgebung erstellt und eine Meldung generiert, sobald ein Objekt sich in einem kritischen Bereich befand. Hierzu wurden die Konzepte des dynamischen Kontextmodells aus Kapitel 3 mit der indirekten Qualitätsbestimmung aus Abschnitt 4.4 verknüpft. Im zweiten Verarbeitungspfad wurde die Qualität von Positionsmessungen eines Ultrabreitband-Positionssystem [WJKC12] durch die Verwendung der in Abschnitt 4.5 entwickelten direkten Qualitätsbestimmung ermittelt und durch den erweiterten tempora-

len relationalen Verbundoperator aus Kapitel 5 mit der Position der Last verknüpft. Hierbei wurde auch gezeigt, wie Widersprüche in der Verarbeitung einer Applikation mitgeteilt werden können, indem die Ergebnisse der zwei Verarbeitungspfade über einen Verbundoperator verknüpft werden können. Das Ergebnis dieser Verknüpfung enthielt dabei die gemeinsame Existenzwahrscheinlichkeit der beiden Teilanfragen.

8.2 Bewertung

Am Beispiel der Fallstudie Sichere Offshore-Operationen wurde gezeigt, wie die drei Teilprobleme der Forschungsfrage gelöst werden konnten. Statt wie bei bisherigen Anwendungen eine fest implementierte Verarbeitung von Sensormessungen zu verwenden, wurde ein Datenstrommanagementsystem genutzt um die Verarbeitung flexibel durch eine Anfragesprache zur beschreiben. Durch die Verwendung der SSN-Ontologie für die indirekte Qualitätsbestimmung wurde eine bereits etablierte Ontologie zur Sensorbeschreibung verwendet. Es kann daher davon ausgegangen werden, dass in Zukunft auf bereits existierende Sensorbeschreibung zugegriffen werden kann und ein Anwender nicht für jede Anwendung explizit die Abhängigkeiten zu unterschiedlichen Umwelteigenschaften und ihren Einfluss auf die Qualität der Sensorwahrnehmung modellieren muss. Bei der direkten Qualitätsbestimmung sind die integrierten Verfahren zur Annäherung des stochastischen Modells noch sehr zeitintensiv und wirken sich daher negativ auf die Latenz der Verarbeitung aus. Hier gilt es bessere Verfahren zu integrieren und zu evaluieren, die bei einer geringeren Latenz ein gleich gutes oder besseres stochastischen Modell ermitteln können. Die Erweiterung der temporalen relationalen Algebra zur Repräsentation von Qualitäten ließ sich ohne große Änderungen an der Gesamtarchitektur des verwendeten Datenstrommanagementsystems integrieren. Allerdings sind die Laufzeiten des Selektionsoperators und des Verbundoperators auf Grund der Integration der verwendeten mehrdimensionalen Mischverteilungen sehr hoch. In den betrachteten Anwendungsszenarien ist diese Latenz noch vertretbar, allerdings nur unter der Annahme, dass sich Objekte in der Umgebung mit Schrittgeschwindigkeit fortbewegen. In dynamischeren Anwendungen muss daher die Frage gestellt werden, welche Sensorwahrnehmungen durch eine probabilistische Verarbeitung fusioniert werden müssen und welche Sensorwahrnehmungen zunächst auf eine deterministische Weise verarbeitet werden können. Durch die Trennung der einzelnen Konzepte bei der Implementierung des Prototyps und dem Erhalt der existierenden Funktionalität des genutzten Datenstrommanagementsystems sind beide Arten der Verarbeitung innerhalb einer Anfrage möglich.

8.3 Ausblick

Die entwickelten Konzepte in dieser Arbeit bieten eine Grundlage zur qualitätssensitiven Datenstromverarbeitung. Bei der Verarbeitung und Nutzung von qualitätsannotierten Daten entstehen allerdings auch neue Fragestellungen. Eine der zentralen Fragen ist dabei

die Interpretation von qualitätsannotierten Daten durch einen Nutzer oder eine Anwendung. Auch muss die Art der Repräsentation untersucht werden. Hier ist vor allem entscheidend, ab welcher Qualität die Verarbeitungsergebnisse einem Anwender als Entscheidungsgrundlage dargebracht werden sollten und welche Kriterien dafür ausschlaggebend sind.

Eine weitere Frage, die es zu beantworten gilt, ist die Art und Verwendung der gewählten Verteilungen. Die in dieser Arbeit gewählte Form der Mischtyp-Verteilung ist für die Repräsentation von Unsicherheiten im Bereich der Sensorfusion angebracht, jedoch gilt es zu untersuchen welche Art der Verteilung in anderen Anwendungsdomänen, wie etwa der Verarbeitung von Finanzströmen oder der Verarbeitung von medizinischen Daten, geeignet ist, um die Qualität adäquat darzustellen.

Zusätzlich zu dem hier betrachteten Datenmodell existieren gerade auch im Bereich der probabilistischen Datenbanken Ansätze zur Verarbeitung von Unsicherheiten in semi-strukturierten und XML-Daten [NJ02]. Da bereits Ansätze zur Verarbeitung von geschachtelten Daten im Bereich der Datenstromverarbeitung existieren, können diese Techniken möglicherweise auf diese Datenmodelle übertragen werden. Eine weitere Frage betrifft die Verknüpfung der hier entwickelten Verarbeitungskonzepte mit anderen Ansätzen der kontinuierlichen Datenstromverarbeitung. So könnten etwa Ansätze zum maschinellen Lernen [Gee13] mit der qualitätssensitiven Verarbeitung kombiniert werden um etwa Klassifizierungsergebnisse als diskrete Wahrscheinlichkeitsverteilung innerhalb des Datenstrommanagementsystems darzustellen und durch die entwickelten Operatoren weiter zu verarbeiten. Auch die Integration von Unsicherheiten bezüglich des Zeitpunktes einer Wahrnehmung, wie er in [ZDI12] betrachtet wird, ist ein Punkt der bei der qualitätssensitiven Verarbeitung eine wichtige Rolle spielt.

A Anhänge

An dieser Stelle wird nochmals auf die Evaluation der direkten Qualitätsbestimmung mit unterschiedlichen Datenfenstergrößen eingegangen und die verwendete Anfrage für die Evaluation der Fallstudie Sichere Offshore-Operationen im Detail erläutert.

A.1 Direkte Qualitätsbestimmung

In Absatz 7.1.3 wurden die Verfahren zur direkten Qualitätsbestimmung durch das Erwartungswertmaximierungsverfahren, der Kerndichteschätzung und der Kombination aus Kerndichteschätzung und Bregman-Hard Clustering verglichen. Bei einem Datenfenster der Größe 100 war hierbei das Erwartungswertmaximierungsverfahren sowohl hinsichtlich der Latenz als auch hinsichtlich der Güte auf Basis des Akaike-Informationskriterium (AIC) besser für die kontinuierliche Verarbeitung von Sensordaten geeignet. Bei einem Datenfenster der Größe 10 lieferte allerdings das Bregman-Hard Clustering, im Vergleich zum Erwartungswertmaximierungsverfahren, die besseren stochastischen Modelle. In Abbildung A.1 und A.2 sind hierzu die Latenzen der drei Verfahren abgetragen. Es zeigt sich, dass die Latenz des Bregman-Hard Clustering in diesem Fall nur minimal höher ist als die Latenz des Erwartungswertmaximierungsverfahrens. Bei einem Datenfenster der Größe 1000 ist die Latenz des Erwartungswertmaximierungsverfahrens allerdings deutlich geringer (vgl. Abbildung A.3 und A.4).

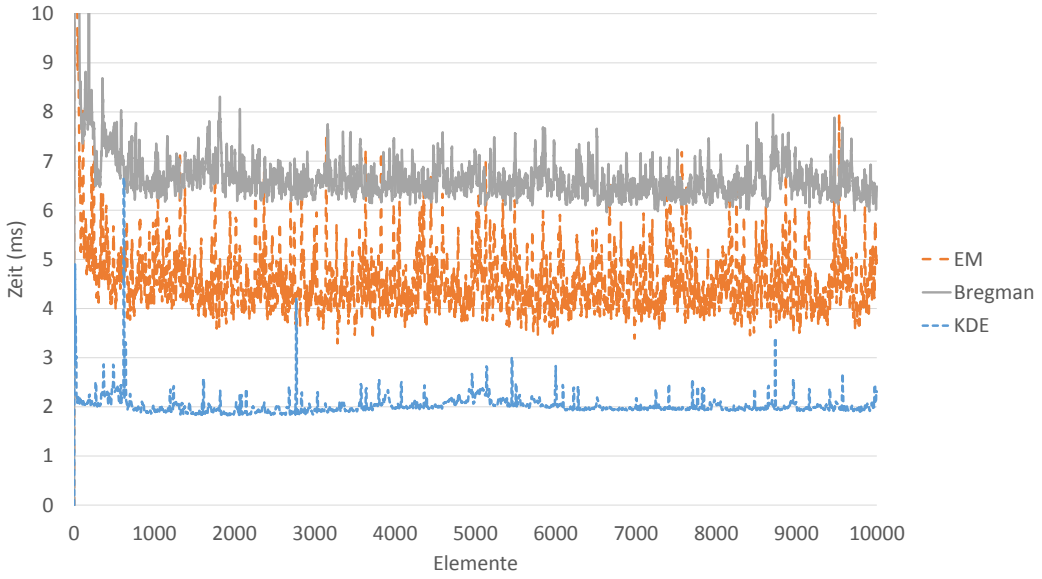


Abbildung A.1: *Latenz der Operatoren bei einem Datenfenster der Größe 10 für Daten aus einer Normalverteilung*

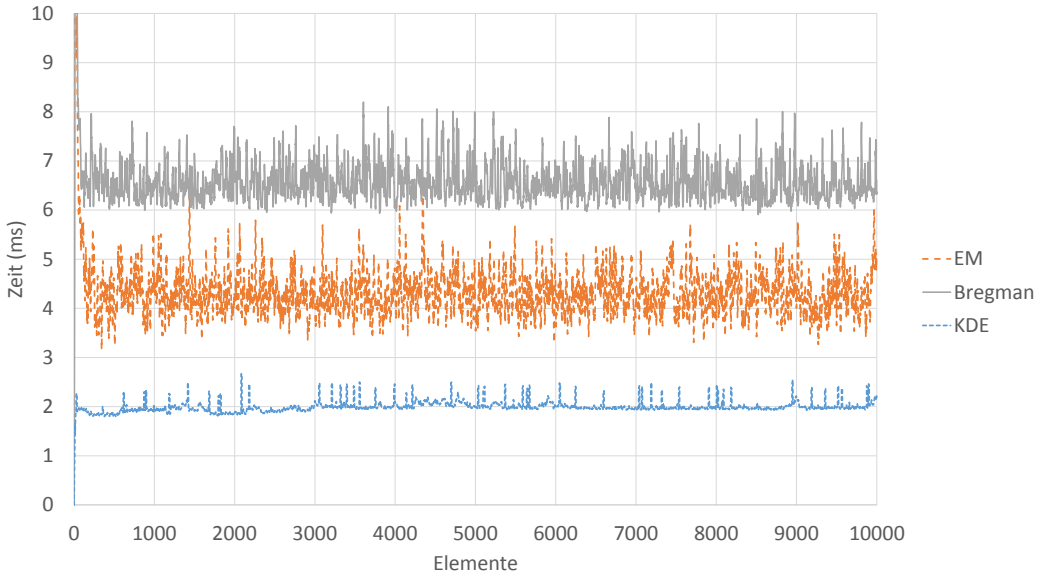


Abbildung A.2: *Latenz der Operatoren bei einem Datenfenster der Größe 10 für Daten aus einer logarithmischen Normalverteilung*

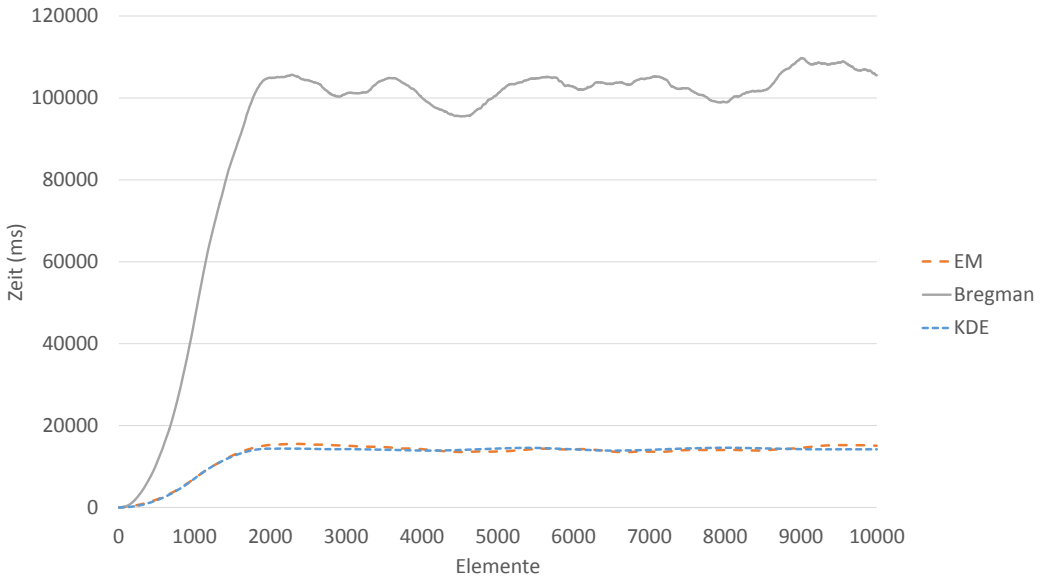


Abbildung A.3: *Latenz der Operatoren bei einem Datenfenster der Größe 1000 für Daten aus einer Normalverteilung*

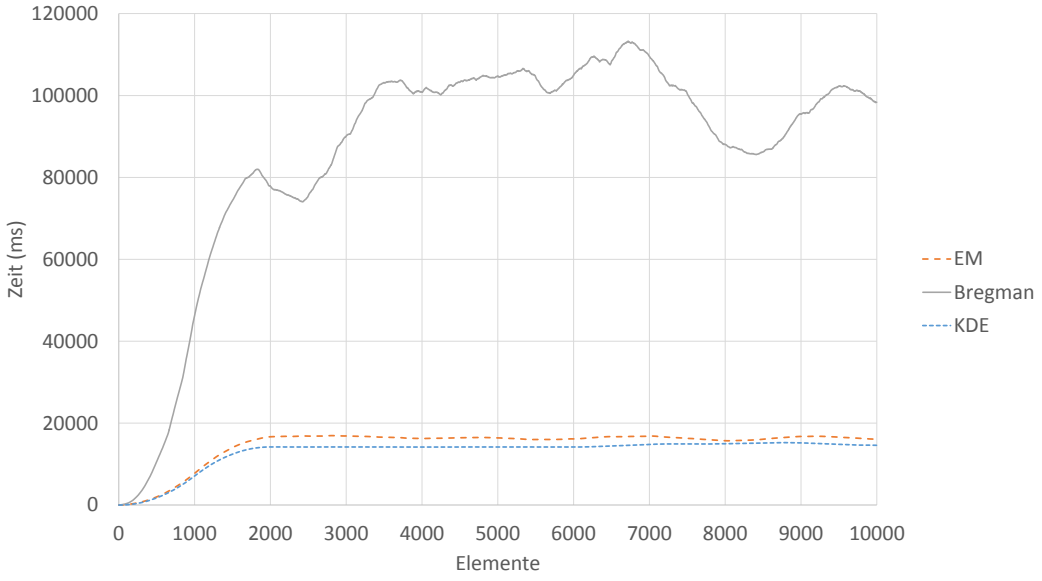


Abbildung A.4: *Latenz der Operatoren bei einem Datenfenster der Größe 1000 für Daten aus einer logarithmischen Normalverteilung*

A.2 Fallstudie: Sichere Offshore-Operationen

In Listing A.1 findet sich die verwendete Anfrage in der Procedural Query Language (PQL) für das Datenstrommanagementsystem (DSMS) Odysseus zur Evaluation der Fallstudie Sichere Offshore-Operation. In den Zeilen 2-4 werden zunächst die verwendeten Metadaten zur Nutzung des Intervall-Ansatzes und der Tupelexistenzwahrscheinlichkeit, sowie eines Metadatums zur Berechnung der Verarbeitungslatenz hinzugefügt. In den Zeilen 7-22 werden anwendungsspezifische Parameter, wie die Größe der überwachten Fläche und die Geschwindigkeit von Objekten in der Anwendung, als globale Konstanten für die kontinuierliche Verarbeitung festlegen.

```
#PARSER PQL
#METADATA TimeInterval
#METADATA Probabilistic
#METADATA Latency
#ADDQUERY

/// Size of observed area
#DEFINE DISTANCE 10000
/// Size of one cell
#DEFINE CELLSIZE 100
/// Radius of polar grid cell
#DEFINE RADIUS 25
/// Velocity of objects (10km/h)
#DEFINE VELOCITY 0.027778
/// Size of element window
#DEFINE WINDOWSIZE 10

/// Location of cargo area
#DEFINE CARGO_MIN_X 3800.0
#DEFINE CARGO_MAX_X 4000.0
#DEFINE CARGO_MIN_Y 3600.0
#DEFINE CARGO_MAX_Y 3800.0

/// Setup laserscanner
LMS1 = MAP({EXPRESSIONS = [['rotateDistanceMatrix(
   DISTANCE16BIT, toRadians(-181.5))','DISTANCE16BIT'],['
   1061','x'],['14454.0','y']]},
  LMS151
)

```

```
LMS2 = MAP({EXPRESSIONS = [['rotateDistanceMatrix(
   DISTANCE16BIT, toRadians(106.5))','DISTANCE16BIT'],['
   7717.0','x'],['14810.0','y']]},
  LMS100_1
)

LMS3 = MAP({EXPRESSIONS = [['rotateDistanceMatrix(
   DISTANCE16BIT, toRadians(-45.0))','DISTANCE16BIT'],['
   5000.0','x'],['0.0','y']]},
  LMS100_2
)

/// Merge laserscanner data into global context model
Grid = MAP({EXPRESSIONS = [['streamtime()','timestamp'],['
   merge(grid, ${CELLSIZE}, DISTANCE16BIT, x, y, ${RADIUS},
   accuracy)','grid']]},
    /// Predict context model to current timestamp using
       object velocity
    MAP({EXPRESSIONS = [['spread(eif(!isNull(grid),grid,
       toSpatialGrid(${DISTANCE}/${CELLSIZE} * 2 + 2, ${
       DISTANCE}/${CELLSIZE} * 2 + 2)), eif(!isNull(
       timestamp),timestamp,streamtime()), ${VELOCITY})','
       grid'],['DISTANCE16BIT','DISTANCE16BIT'],['x','x'],['
       y','y']]},
      /// Enrich stream with latest context model
      CONTEXTENRICH({STORE = 'myStore', OUTER = true},
        UNION(
          /// Use of sensor accuracy information stored
             within the ontology
          QUALITY(LMS1, "Accuracy"),
          QUALITY(LMS2, "Accuracy"),
          QUALITY(LMS3, "Accuracy")
        )
      )
    )
)

/// Write current context model back to context store
ContextStore = STORE({STORE = 'myStore'},
  Grid
)
```

```
/// Setup supervisor position
Supervisor = EM({ATTRIBUTES = ['x','y'], MIXTURES = 2,
   ITERATIONS=30, THRESHOLD=0.001},
    ELEMENTWINDOW({size = ${WINDOWSIZE}},
        MAP({EXPRESSIONS = [['-1* posX -1363.6 +5000.0', 'x'
           ],['-1* posY +2942.3','y']]},
            UWB3
        )
    )
)

/// Hazard 1: Supervisor position under cargo
Hazard1 = SELECT({predicate =
   ProbabilisticRelationalPredicate('as2DVector(x,y) > [${
   CARGO_MIN_X},${CARGO_MIN_Y}] && as2DVector(x,y) < [${
   CARGO_MAX_X},${CARGO_MAX_Y}]')},
    Supervisor
)

/// Hazard 2: Occupy probability of cells under cargo
Hazard2 = Probabilistic({ATTRIBUTE='p'},
    MAP({EXPRESSIONS = [['1.0 - isFree(grid,${CARGO_MIN_X}/$
       {CELLSIZE},${CARGO_MIN_Y}/${CELLSIZE},${CARGO_MAX_X}/
       ${CELLSIZE},${CARGO_MAX_Y}/${CELLSIZE})','p']]} ,
        Grid
    )
)


/// Confidence of both hazards
Confidence = JOIN(
  ELEMENTWINDOW({size = 1},
    Hazard1
  ),
  ELEMENTWINDOW({size = 1},
    Hazard2
  )
)


#PARSER CQL
#METADATA TimeInterval
```

```
#METADATA Probabilistic
#METADATA Latency
#RUNQUERY

/// Create context store for context models
DROP CONTEXT STORE myStore IF EXISTS
CREATE CONTEXT STORE myStore (timestamp LONG, grid Grid) AS
   SINGLE
```

Quelltext A.1: *Anfrage zur Ermittlung der Hazards 1 und 2 aus der Fallstudie Sichere Offshore-Operationen*

Die Zeilen 24-35 binden die verwendeten Laserscanner ein und transformieren ihre Messungen in das globale Koordinatensystem. In den Zeilen 37-51 werden ihre Messungen anschließend in die globale Belegungskarte eingearbeitet. Hierzu wird zunächst in den Zeilen 44-47 der Datenstrom der Laserscanner durch den Wert der Standardabweichung aus der Ontologie über den *QUALITY*-Operator erweitert. Durch den *UNION*-Operator werden die Ströme der einzelnen Laserscanner in einen Datenstrom zusammengefasst und über den *CONTEXTENRICH*-Operator mit der letzten erstellten Belegungskarte aus dem Kontextspeicher erweitert. Zeile 40 prädiziert die Belegungskarte auf den aktuellen Zeitpunkt. Der aktuelle Zeitpunkt wird dabei über die Funktion *streamtime()* ermittelt. Abschließend werden in Zeile 38 die aktuellen Messungen in die prädizierte Belegungskarte integriert. In Zeile 54 wird die so aktualisierte Belegungskarte mit dem Zeitstempel der letzten Messung wieder in den Kontextspeicher mit dem Bezeichner *myStore* geschrieben.

Zur Ermittlung des Rauschens der Positionsmessungen wird in Zeile 59 durch den *EM*-Operator das stochastische Modell angenähert. Hierzu werden zuvor die Positionsmessungen durch den *MAP*-Operator in das globale Koordinatensystem transformiert und über den *ELEMENTWINDOW*-Operator die zeitliche Gültigkeit der Messungen festgelegt.

Die beiden Hazards werden anschließend in den Zeilen 68 und 73 erkannt. Hierzu wird zur Ermittlung von Hazard 1 die Position des Lift-Supervisors mit der Position der Last in einem *SELECT*-Operator verglichen, wobei ein Integral über das zuvor ermittelte stochastische Modell gebildet wird. Der zweite Hazard wird durch die maximale Wahrscheinlichkeitsmasse in den belegten Zellen der Belegungskarte durch die Funktion *isFree()* ermittelt. Der so resultierende Wahrscheinlichkeitswert wird über den *Probabilistic*-Operator in die Metadatenebene als Existenzwahrscheinlichkeit kopiert. In Zeile 81 wird abschließend ein Verbund aus den beiden Strömen gebildet, wodurch die Existenzwahrscheinlichkeiten der Elemente aus beiden Strömen multipliziert werden.

Im zweiten Teil der Anfrage wird in Zeile 99 in der Continuous Query Language (CQL) ein Kontextspeicher definiert, in dem die Belegungskarte zwischen zwei Verarbeitungszyklen temporär gespeichert wird.

Glossar

Nachfolgend sind noch einmal wesentliche Begriffe dieser Arbeit zusammengefasst und erläutert. Eine ausführliche Erklärung findet sich jeweils in den einführenden Abschnitten, sowie der jeweils darin angegebenen Literatur.

Bregman-Hard Clustering

Das $\sim$ ist ein Clustering-Verfahren, bei dem die Bregman Divergenz als Distanzmaß verwendet wird um Repräsentanten für ähnliche Verteilungen innerhalb einer ↑ Mischverteilung zu bestimmen. 73

Erwartungswertmaximierungsverfahren

Das $\sim$ stellt ein Verfahren dar, um die Parameter eines stochastischen Modells in einem iterativen Verfahren anzunähern. 70

Kerndichteschätzungsverfahren

Das $\sim$ dient dazu, das stochastische Modell von Daten durch eine Menge von Gauß-Verteilungen mit gleicher Kovarianzmatrix und gleicher Gewichtung zu repräsentieren. 72

Logischer Anfragegraph

Ein $\sim$ ist ein gerichteter Graph (V, E) bestehend aus einer Menge von logischen ↑ Operatoren V, welche über die gerichteten Kanten E miteinander verbunden sind. Die Richtung der Kanten gibt dabei die Flussrichtung der Daten im Anfragegraphen vor. 41, 46

Logischer Datenstrom

Ein $\sim$ ist eine möglicherweise unendliche Sequenz von ↑ Tupeln, die jeweils zu einem Zeitpunkt t gültig sind und n-mal zu diesem Zeitpunkt existieren. 15

Mischtyp-Verteilung

Eine $\sim$ ist ein Paar (p, f) mit Tupelexistenzwahrscheinlichkeit $p \in [0, 1]$ und Dichtefunktion f. 88

Mischverteilung

Eine $\sim$ ist eine Kombination aus gewichteten Verteilungen gleicher Art. 92

Odysseus

$\sim$ ist ein DSMS, das in der Abteilung Informationssysteme der Carl von Ossietzky Universität Oldenburg entwickelt wurde. 101–108, 111, 112, 115, 136, 145, 152, 159, 160

Odysseus Script

$\sim$ ist eine Konfigurationssprache zur Konfiguration und Steuerung des Datenstrommanagementsystems ↑ Odysseus. 103, 159

Operator

Unter einem $\sim$ versteht man eine Verarbeitungseinheit, die ein oder mehrere Datenströme konsumiert und ein oder mehrere Datenströme produziert. 18

Physischer Datenstrom

Ein $\sim$ ist eine möglicherweise unendliche Sequenz von ↑ Tupeln, die jeweils über eine Zeitspanne $[t_S, t_E)$ gültig sind. 17

SALSA

Das $\sim$ -Projekt ist ein vom Bundeswirtschaftsministerium (BMWi) gefördertes Verbundprojekt zur Entwicklung von autonomen fahrerlosen Transportfahrzeugen. 112, 113, 145, 160

Semantic-Sensor-Network-Ontologie

Die $\sim$ ist eine von der W3C Semantic Sensor Network Incubator Group veröffentlichte Ontologie zur Beschreibung der syntaktischen Interoperabilität und semantische Kompatibilität von Sensoren. 62

Sensor

Unter einem $\sim$ versteht man ein technisches Bauteil, das bestimmte physikalische oder chemische Eigenschaften seiner Umwelt qualitativ oder in Messgrößen quantitativ erfassen kann. Ein $\sim$ kann aber auch eine Softwarekomponente sein, die Messwerte über ein empirisches erlerntes oder physikalisches Modell berechnen kann. 7

SOOP

Das $\sim$ -Projekt ist ein vom Europäischen Fonds für Regionale Entwicklung (EFRE) gefördertes Forschungsprojekt mit dem Ziel, neue Verfahren und Werkzeuge zu entwickeln, um kritische Prozesse bei Offshore-Wind-Operationen zu planen, zu simulieren und zu trainieren. 114, 115, 145, 160

Tupel

Ein $\sim$ ist eine Ausprägung eines Datenstromschemas. 16

Abkürzungen

AIC Akaike-Informationskriterium. 117, 125, 126, 128, 149, 160

CQL Continuous Query Language. 23, 155

DOLCE Descriptive Ontology for Linguistic and Cognitive Engineering Ontologie. 63

DSMS Datenstrommanagementsystem. 62, 85, 152

JDL Joint Directors of Laboratories. 7, 11, 12, 24, 25, 159

O&M Observations and Measurements. 63

OSGi Open Services Gateway Specification. 102, 103, 109, 110

OWL Web Ontology Language. 63

PQL Procedural Query Language. 103, 152

RDFS Resource Description Framework Schema. 63

SASE Stream-based And Shared Event processing. 103

SensorML Sensor Model Language. 63

SQL Structured Query Language. 103

SSN Semantic-Sensor-Network-Ontologie. 62–67, 69, 74, 84, 107, 110, 111, 121, 144, 146, 159

Symbole

- $\mathbb{F}_{map}$ Menge aller Abbildungsfunktionen [19]
- μ Abbildungsoperator [19]
- α Aggregationsoperator [20]
- t_A Applikationszeitstempel [57]
- t_O Ausgabezeitstempel [57]
- $\mathcal{T}$ Datenstromschema [15–17]
- t_E Gültigkeitsendzeitstempel [16]
- ω Fensteroperator [21]
- $[t_S, t_E)$ Gültigkeitszeitintervall [16]
- $\mathcal{N}$ Gauß-Verteilung [70]
- $\mathbb{S}^l$ Menge aller logischen Datenströme [16]
- S^l Logischer Datenstrom [15, 16]
- L Log-Likelihood [70]
- Ω Menge aller Nutzdaten [16]
- $\mathbb{S}^p$ Menge aller physischen Datenströme [17]
- S^p Physischer Datenstrom [17]
- π Projektionsoperator [19]
- $\mathbb{P}_{\mathcal{T}}$ Menge aller Selektionskriterien [19]
- σ Selektionsoperator [18]
- t_S Gültigkeitsstartzeitstempel [16]
- $\times$ Verbundoperator [20]
- T Menge aller Zeitstempel [15]

Abbildungen

Algorithmen

Literatur

[ABB+03] Arasu, Arvind ; Babcock, Brian ; Babu, Shivnath ; Datar, Mayur ; Ito, Keith ; Nishizawa, Itaru ; Rosenstein, Justin ; Widom, Jennifer: STREAM: The Stanford Stream Data Manager. In: *Proceedings of the 2003 ACM SIGMOD international conference on on Management of data*. New York, NY, USA : ACM Press, 2003 (SIGMOD '03), S. 665

[ABC+04] Agostini, Alessandra ; Bettini, Claudio ; Cesa-Bianchi, Nicolò ; Maggiorini, Dario ; Riboni, Daniele ; Ruberl, Michele ; Sala, Cristiano ; Vitali, Davide: Towards Highly Adaptive Services for Mobile Computing. In: *Proceedings of IFIP International Federation for Information Processing* Bd. 158, 2004, S. 121–134

[ACc+03] Abadi, Daniel J. ; Carney, Don ; Çetintemel, Ugur ; Cherniack, Mitch ; Convey, Christian ; Lee, Sangdon ; Stonebraker, Michael ; Tatbul, Nesime ; Zdonik, Stan: Aurora: a new model and architecture for data stream management. In: *The VLDB Journal* 12 (2003), August, Nr. 2, S. 120–139

[AGG+12] Appelrath, H.-J. ; Geesen, Dennis ; Grawunder, Marco ; Michelsen, Timo ; Nicklas, Daniela: Odysseus: a highly customizable framework for creating efficient event stream management systems. In: *Proceedings of the 6th ACM International Conference on Distributed Event-Based Systems*. New York, NY, USA : ACM Press, 2012 (DEBS '12), S. 367–368

[BBC+09] Barbieri, Davide F. ; Braga, Daniele ; Ceri, Stefano ; Della Valle, Emanuele ; Grossniklaus, Michael: C-SPARQL: SPARQL for Continuous Querying. In: *Proceedings of the 18th International Conference on World Wide Web*. New York, NY, USA : ACM Press, 2009 (WWW '09), S. 1061–1062

[BBD+02] Babcock, Brian ; Babu, Shivnath ; Datar, Mayur ; Motwani, Rajeev ; Widom, Jennifer: Models and Issues in Data Stream Systems. In: *Proceedings of the 21st ACM SIGMOD-SIGACT-SIGART Symposium on Principles of Database Systems*. New York, NY, USA : ACM Press, 2002 (PODS '02), S. 1–16

[BDR07] Baldauf, Matthias ; Dustdar, Schahram ; Rosenberg, Florian: A survey on context-aware systems. In: *International Journal of Ad Hoc and Ubiquitous Computing* 2 (2007), Nr. 4, S. 263–277

[BE06] Berti-Équille, Laure: Data quality awareness: a case study for cost optimal association rule mining. In: *Knowledge and Information Systems* 11 (2006), März, Nr. 2, S. 191–215

[BKF+07] Botan, I. ; Kossmann, D. ; Fischer, P.M. ; Kraska, T. ; Florescu, D. ; Tamosevicius, R.: Extending Xquery with Window Functions. In: *Proceedings of*

the 33rd International Conference on Very Large Data Bases VLDB Endowment, 2007, S. 75–86

[BKNB11] Busemann, Claas ; Kuka, Christian ; Nicklas, Daniela ; Boll, Susanne: Flexible and Efficient Sensor Data Processing – A Hybrid Approach. In: *Datenbanksysteme für Business, Technologie und Web (BTW), 14. Fachtagung des GI-Fachbereichs "Datenbanken und Informationssysteme"(DBIS)* Bd. 180, GI, 2011 (LNI), S. 123–134

[BKS03] Buchholz, Thomas ; Küpper, Axel ; Schiffers, Michael: Quality of context: What it is and why we need it. In: *Proceedings of the 10th International Workshop of the HP OpenView* (2003)

[BKW+09] Busemann, Claas ; Kuka, Christian ; Westermann, Utz ; Boll, Susanne ; Nicklas, Daniela: SCAMPI-Sensor Configuration and Aggregation Middleware for Multi Platform Interchange. In: *GI Jahrestagung*, 2009, S. 2084–2097

[BMDG05] Banerjee, Arindam ; Merugu, Srujana ; Dhillon, Inderjit S. ; Ghosh, Joydeep: Clustering with Bregman Divergences. In: *Journal of Machine Learning Research* 6 (2005), S. 1705–1749

[BMR07] Bettini, C. ; Maggiorini, D. ; Riboni, D.: Distributed Context Monitoring for the Adaptation of Continuous Services. In: *World Wide Web Journal* 10 (2007), Nr. 4, S. 503–528

[Bol11] Bolles, André: *Ein datenstrombasiertes Framework zur Entwicklung von Objektverfolgungssystemen am Beispiel von Fahrerassistenzsystemen*, Carl von Ossietzky-Universität Oldenburg, Dissertation, 2011

[BRB+14] Boano, Carlo Alberto ; Roemer, Kay ; Brown, James ; Roedig, Utz ; Zúñiga, Marco Antonio: Demo abstract: A testbed infrastructure to study the impact of temperature on WSN. In: *IEEE International Conference on Pervasive Computing and Communications Workshops (PERCOM Workshops)*. Los Alamitos, CA, USA : IEEE Computer Society, 2014, S. 154–156

[BS06] Batini, C. ; Scannapieca, M.: *Data Quality: Concepts, Methodologies and Techniques*. Berlin, Heidelberg : Springer-Verlag, 2006 (Data-Centric Systems and Applications)

[CBB+12] Compton, Michael ; Barnaghi, Payam ; Bermudez, Luis ; García-Castro, Raúl ; Corcho, Oscar ; Cox, Simon ; Graybeal, John ; Hauswirth, Manfred ; Henson, Cory ; Herzog, Arthur ; Huang, Vincent ; Janowicz, Krzysztof ; Kelsey, W. D. ; Le Phuoc, Danh ; Lefort, Laurent ; Leggieri, Myriam ; Neuhaus, Holger ; Nikolov, Andriy ; Page, Kevin ; Passant, Alexandre ; Sheth, Amit ; Taylor, Kerry: The SSN ontology of the W3C semantic sensor

network incubator group. In: *Web Semantics: Science, Services and Agents on the World Wide Web* 17 (2012), Dezember, S. 25–32

[CCMP06] Cappiello, C. ; Comuzzi, M. ; Mussi, E. ; Pernici, B.: Context Management for Adaptive Information Systems. In: *Electronic Notes in Theoretical Computer Science* 146 (2006), Nr. 1, S. 69–84

[CEB+09] Cipriani, Nazario ; Eissele, Mike ; Brodt, Andreas ; Grossmann, Matthias ; Mitschang, Bernhard: NexusDS: a flexible and extensible middleware for distributed stream processing. In: *Proceedings of the 2009 International Database Engineering & Applications*. New York, NY, USA : ACM Press, 2009 (IDEAS '09), S. 152

[CFJ04] Chen, H. ; Finin, T. ; Joshi, A.: Semantic web in the context broker architecture. Los Alamitos, CA, USA : IEEE Computer Society, 2004, S. 277–286

[CPL+06] Coue, C. ; Pradalier, C. ; Laugier, C. ; Fraichard, T. ; Bessiere, P.: Bayesian Occupancy Filtering for Multitarget Tracking: An Automotive Application. In: *The International Journal of Robotics Research* 25 (2006), Januar, Nr. 1, S. 19–30

[Dat82] Date, C. J.: A formal definition of the relational model. In: *SIGMOD Rec.* 13 (1982), Nr. 1, S. 18–29

[Dey01] Dey, Anind K.: Understanding and Using Context. In: *Personal Ubiquitous Comput.* 5 (2001), Januar, Nr. 1, S. 4–7

[DLR77] Dempster, Arthur P. ; Laird, Nan M. ; Rubin, Donald B.: Maximum likelihood from incomplete data via the EM algorithm. In: *Journal of the Royal statistical Society* 39 (1977), Nr. 1, S. 1–38

[DMC00] Drolet, Louis ; Michaud, François ; Côté, Jean: Adaptable sensor fusion using multiple Kalman filters. In: *IROS*, 2000, S. 1434–1439

[DW04] Dias, José G. ; Wedel, Michel: An Empirical Comparison of EM, SEM and MCMC Performance for Problematic Gaussian Mixture Likelihoods. In: *Statistics and Computing* 14 (2004), Oktober, Nr. 4, S. 323–332

[EFG+12] Eilers, Sönke ; Fränzle, Martin ; Gerwinn, Sebastian ; Kuka, Christian ; Schweigert, Sören ; Toben, Tobe: An Autonomous Vehicle Design for Safe Operation in Heterogeneous Environments / School of Computing Science, University of Newcastle upon Tyne. 2012. – Forschungsbericht

[FBK+11] Funk, Alexander ; Busemann, Claas ; Kuka, Christian ; Boll, Susanne ; Nicklas, Daniela: Open Sensor Platforms : The Sensor Web Enablement Framework and Beyond. In: *Proceedings der 6. MMS 2011: Mobile und ubiquitäre Informationssysteme* Bd. 185. Kaiserslautern, Germany : GI, 2011 (LNI), S. 39–52

[FKM+05] Feigenbaum, Joan ; Kannan, Sampath ; McGregor, Andrew ; Suri, Siddharth ; Zhang, Jian: On graph problems in a semi-streaming model. In: *Theoretical Computer Science* 348 (2005), Nr. 2–3, S. 207–216

[FMS+10] Filho, José B. ; Miron, Alina D. ; Satoh, Ichiro ; Gensel, Jérôme ; Martin, Hervé: Modeling and Measuring Quality of Context Information in Pervasive Environments. In: *24th IEEE International Conference on Advanced Information Networking and Applications* (2010), S. 690–697

[FSL07] Fulgenzi, Chiara ; Spalanzani, Anne ; Laugier, Christian: Combining Probabilistic Velocity Obstacles and Occcupancy Grid for safe Navigation in dynamic environments. In: *ICRA 2007 Workshop: Planning, Perception and Navigation for Intellilgent Vehicles*, 2007

[GADI08] Gyllstrom, Daniel ; Agrawal, Jagrati ; Diao, Yanlei ; Immerman, Neil: On Supporting Kleene Closure over Event Streams. In: *2014 IEEE 30th International Conference on Data Engineering* 0 (2008), S. 1391–1393

[GAW+08] Gedik, Bugra ; Andrade, Henrique ; Wu, Kun-Lung ; Yu, Philip S. ; Doo, Myungcheol: SPADE: The System S declarative stream processing engine. In: *The ACM SIGMOD international conference on Management of data*. New York, NY, USA : ACM Press, 2008 (SIGMOD '08), S. 1123

[GBG+12] Geesen, Dennis ; Brell, Melina ; Grawunder, Marco ; Nicklas, Daniela ; Appelrath, Hans-Jürgen: Data Stream Management in the AAL - Universal and Flexible Preprocessing of Continuous Sensor Data. In: *Ambient Assisted Living*. Berlin, Heidelberg : Springer-Verlag, 2012, S. 213–228

[GBH+05] Grossmann, M. ; Bauer, M. ; Honle, N. ; Kappeler, U. P. ; Nicklas, D. ; Schwarz, T.: Efficiently Managing Context Information for Large-Scale Scenarios. In: *Proceedings of Pervasive Computing and Communications*. Los Alamitos, CA, USA : IEEE Computer Society, 2005, S. 331–340

[Gee13] Geesen, Dennis: *Maschinelles Lernen in Datenstrommanagementsystemen*, Carl von Ossietzky-Universität Oldenburg, Dissertation, 2013

[Gen92] Genz, A.: Numerical Computation of Multivariate Normal Probabilities. In: *J. of Comp. Graph. Stat.* 1 (1992), S. 141–149

[GGKL07] Groppe, S. ; Groppe, J. ; Kukulenz, D. ; Linnemann, V.: A SPARQL Engine for Streaming RDF Data. In: *3rd International IEEE Conference on Signal-Image Technologies and Internet-Based System*. Los Alamitos, CA, USA : IEEE Computer Society, Dec 2007, S. 167–174

[GHM+07] Ghanem, T.M. ; Hammad, M.A. ; Mokbel, M.F. ; Aref, W.G. ; Elmagarmid, A.K.: Incremental Evaluation of Sliding-Window Queries over Data Streams.

In: *IEEE Transactions on Knowledge and Data Engineering* 19 (2007), Jan, Nr. 1, S. 57–72

[GQWJ11] Geisler, Sandra ; Quix, Christoph ; Weber, Sven ; Jarke, Matthias: An Ontology-based Framework for Quality-oriented Data Stream Management. In: *Proceedings of the 16th International Conference on Information Quality*, 2011

[GTN11] Gallera, Diana von ; Trujillo, Juan J. ; Nicklas, Daniela: Leistungskennlinienberechnung von Windenergieanlagen unter Einsatz eines Datenstrommanagementsystems. In: *Proceedings BTW 2011 - Workshops und Studierendenprogramm*, 2011, S. 43–51

[HAQJ14] Heinze, Thomas ; Aniello, Leonardo ; Querzoni, Leonardo ; Jerzak, Zbigniew: Tutorial: Cloud-based Data Stream Processing. In: *Proceedings of the 8th ACM International Conference on Distributed Event-Based Systems*. New York, NY, USA : ACM Press, 2014 (DEBS '14)

[Hen03] Henricksen, Karen: *A Framework for Context-Aware Pervasive Computing Applications*, School of Information Technology and Electrical Engineering, The University of Queensland, Diss., September 2003

[HIR08] Hu, Peizhao ; Indulska, Jadwiga ; Robinson, Ricky: An Autonomic Context Management System for Pervasive Computing. In: *Proceedings of the 6th Annual IEEE International Conference on Pervasive Computing and Communications*. Los Alamitos, CA, USA : IEEE Computer Society, 2008 (PerCom '08), S. 213–223

[HJ04] Huang, X ; Jensen, CS: Towards a streams-based framework for defining location-based queries. In: *Proceedings of the 2nd Workshop on Spatial- Temporal Database Management*, 2004, S. 78–85

[HL97] Hall, D.L. ; Llinas, J.: An introduction to multisensor data fusion. In: *Proceedings of the IEEE* 85 (1997), Nr. 1, S. 6–23

[HSE12] HSE: *Offshore Safety Statistics Bulletin 2011/12*. 2012

[HSK13] Hellinger, Ariane ; Stumpf, Veronika ; Kobsda, Christian: Umsetzungsempfehlungen für das Zukunftsprojekt Industrie 4.0. 2013. – Forschungsbericht

[IWM+09] Ilarri, Sergio ; Wolfson, Ouri ; Mena, Eduardo ; Illarramendi, Arantza ; Sistla, Prasad: A Query Processor for Prediction-Based Monitoring of Data Streams. In: *EDBT 2009*. New York, NY, USA : ACM Press, 2009, S. 415–426

[JBG+10] Jacobi, Jonas ; Bolles, Andre ; Grawunder, Marco ; Nicklas, Daniela ; Appelrath, Hans-Jürgen: A physical operator algebra for prioritized elements in data streams. In: *Computer Science - R&D* 25 (2010), Nr. 3–4, S. 235–246

[JBS+13] Janssen, Manuel ; Busboom, Andreas ; Schoon, Udo ; Koch, Carsten ; Cölln, Gerd von: A Hybrid MAC Layer for Localization and Data Communication in Ultra Wide Band Based Wireless Sensor Networks. In: *11th IEEE International Conference on Industrial Informatics (INDIN)*. Los Alamitos, CA, USA : IEEE Computer Society, 2013, S. 474–479

[JG08] Jacobi, Jonas ; Grawunder, Marco: ODYSSEUS: Ein flexibles Framework zum Erstellen anwendungsspezifischer Datenstrommanagementsysteme. In: *Grundlagen von Datenbanken* 1 (2008), S. 86–90

[JM07] Jayram, T. S. ; Muthukrishan, S.: Estimating statistical aggregates on probabilistic data streams. In: *In ACM Symposium on Principles of Database Systems*. New York, NY, USA : ACM Press, 2007, S. 243–252

[Kal60] Kalman, RE: A new approach to linear filtering and prediction problems. In: *Journal of basic Engineering* 82 (1960), Nr. Series D, S. 35–45

[KBNB11] Kuka, Christian ; Busemann, Claas ; Nicklas, Daniela ; Boll, Susanne: Mashups for Community Aware Sensor Processing with SCAMPI. In: *Proceedings BTW 2011 - Workshops und Studierendenprogramm*, Technische Universität Kaiserslautern, 2011, S. 52–57

[KGS+12] Kuka, Christian ; Gerwinn, Sebastian ; Schweigert, Sören ; Eilers, Sönke ; Nicklas, Daniela: Demo: Context-model generation for safe autonomous transport vehicles. In: *Proceedings of the Sixth ACM International Conference on Distributed Event-Based Systems*. New York, NY, USA : ACM Press, 2012 (DEBS '12), S. 365–366

[KKKR11] Khaleghi, Bahador ; Khamis, Alaa ; Karray, Fakhreddine O. ; Razavi, Saiedeh N.: Multisensor data fusion: A review of the state-of-the-art. In: *Information Fusion* (2011), August, Nr. August

[KL09a] Klein, Anja ; Lehner, Wolfgang: How to Optimize the Quality of Sensor Data Streams. In: *2009 Fourth International Multi-Conference on Computing in the Global Information Technology* (2009), S. 13–19

[KL09b] Klein, Anja ; Lehner, Wolfgang: Representing data quality in sensor data streaming environments. In: *Journal of Data and Information Quality (JDIQ)* 1 (2009), Nr. 2, S. 10

[KN12] Kuka, Christian ; Nicklas, Daniela: Approximating Complex Sensor Quality Using Failure Probability Intervals. In: *Proceedings of the 6th International Conference on Scalable Uncertainty Management*. Berlin, Heidelberg : Springer-Verlag, 2012 (SUM'12), S. 287–298

[KN14a] Kuka, Christian ; Nicklas, Daniela: Demo: Supporting quality-aware pervasive applications by probabilistic data stream management. In: *Proceedings of the 8th ACM International Conference on Distributed Event-Based Systems*. New York, NY, USA : ACM Press, 2014 (DEBS '14), S. 330–333

[KN14b] Kuka, Christian ; Nicklas, Daniela: Enriching sensor data processing with quality semantics. In: *2014 IEEE International Conference on Pervasive Computing and Communication Workshops*. Los Alamitos, CA, USA : IEEE Computer Society, 2014, S. 437–442

[KN14c] Kuka, Christian ; Nicklas, Daniela: Quality matters: supporting quality-aware pervasive applications by probabilistic data stream management. In: *Proceedings of the 8th ACM International Conference on Distributed Event-Based Systems*. New York, NY, USA : ACM Press, 2014 (DEBS '14), S. 1–12

[Knu97] Knuth, Donald E.: *Art of Computer Programming, Volume 2: Seminumerical Algorithms (3rd Edition)*. 3. Addison-Wesley Professional, 1997

[Koc09] Koch, Christoph: MayBMS: A System for Managing Large Uncertain and Probabilistic Databases. In: *Managing and Mining Uncertain Data*. Berlin, Heidelberg : Springer-Verlag, 2009

[Krä07] Krämer, Jürgen: *Continuous Queries over Data Streams-Semantics and Implementation*, Philipps-Universität Marburg, Dissertation, 2007

[KS04] Krämer, Jürgen ; Seeger, Bernhard: PIPES - A Public Infrastructure for Processing and Exploring Streams. In: *The ACM SIGMOD international conference on Management of data - SIGMOD '04*. New York, NY, USA : ACM Press, 2004, S. 925

[Kuk12] Kuka, Christian: Processing the uncertainty: Quality-aware data stream processing for dynamic context models. In: *Workshop Proceedings of the 10th Annual IEEE International Conference on Pervasive Computing and Communications*. Los Alamitos, CA, USA : IEEE Computer Society, 2012, S. 560–561

[LBR+04] Llinas, James ; Bowman, Christopher ; Rogova, Galina ; Steinberg, Alan ; Waltz, Ed ; White, Frank: Revisiting the JDL Data Fusion Model II. In: *Proceedings of the 7th International Conference on Information Fusion*, 2004, S. 1218–1230

[MLT+05] Maier, David ; Li, Jin ; Tucker, Peter ; Tufte, Kristin ; Papadimos, Vassilis: Semantics of Data Streams and Operators. In: *Database Theory - ICDT 2005* Bd. 3363. Berlin, Heidelberg : Springer-Verlag, 2005, S. 37–52

[MM88] McKendall, R. ; Mintz, M.: Robust fusion of location information. In: *IEEE International Conference on Robotics and Automation*. Los Alamitos, CA, USA : IEEE Computer Society, Apr 1988, S. 1239–1244 vol.2

[NJ02] Nierman, Andrew ; Jagadish, H. V.: ProTDB: Probabilistic data in XML. In: *In Proceedings of the 28th VLDB Conference.* Berlin, Heidelberg : Springer-Verlag, 2002, S. 646–657

[NN09] Nielsen, Frank ; Nock, Richard: Clustering Multivariate Normal Distributions. In: Nielsen, Frank (Hrsg.): *Emerging Trends in Visual Computing* Bd. 5416. Berlin, Heidelberg : Springer-Verlag, 2009, S. 164–174

[PS04] Patroumpas, K ; Sellis, T: Managing trajectories of moving objects as data streams. In: *Workshop on Spatio-Temporal Database Management (STDBM)*, 2004, S. 41–48

[RHC+02] Roman, Manuel ; Hess, Christopher ; Cerqueira, Renato ; Ranganathan, Anand ; Campbell, Roy H. ; Nahrstedt, Klara: A Middleware Infrastructure for Active Spaces. In: *IEEE Pervasive Computing* 1 (2002), Oct, Nr. 4, S. 74–83

[SBL+12] Sobeich, Cilli ; Boede, Eckard ; Luedtke, Andreas ; Hahn, Axel ; Nicklas, Daniela ; Korte, Holger: Project SOOP: Safe Offshore Operations. In: *ISIS - 9th International Symposium Information on Ships*, DGON (Deutsche Gesellschaft fuür Ortung und Navigation) and German Society for Maritime Technology (STG), 2012

[Sco92] Scott, D.W: *Multivariate Density Estimation: Theory, Practice, and Visualization.* New York, Chicester : John Wiley Sons, 1992

[SFD02] Streller, Daniel ; Furstenberg, K ; Dietmayer, Klaus: Vehicle and object models for robust tracking in traffic scenes using laser range images. In: *Proceedings of the IEEE 5th International Conference on Intelligent Transportation Systems.* Los Alamitos, CA, USA : IEEE Computer Society, 2002, S. 118–123

[Sha76] Shafer, Glenn: *A mathematical theory of evidence.* Bd. 1. Princeton University Press, 1976. – 314 S.

[Sil86] Silverman, B.W.: Density Estimation for Statistics and Data Analysis. In: *Monographs on Statistics and Applied Probability* 26 (1986)

[SWS07] Sheikh, Kamran ; Wegdam, Maarten ; Sinderen, Marten van: Middleware Support for Quality of Context in Pervasive Context-Aware Systems. In: *Fifth Annual IEEE International Conference on Pervasive Computing and Communications Workshops (PerComW'07).* Los Alamitos, CA, USA : IEEE Computer Society, März 2007, S. 461–466

[TCD+05] Thorpe, Chuck ; Carlson, Justin ; Duggins, Dave ; Gowdy, Jay ; MacLachlan, Rob ; Mertz, Christoph ; Suppe, Arne ; Wang, Bob: Safe Robot Driving in Cluttered Environments. In: *Proceedings of the 11th International Symposium on Robotics Research*, 2005, S. 271–280

[TEK+12] Toben, Tobe ; Eilers, Sönke ; Kuka, Christian ; Schweigert, Sören ; Winkelmann, Hannes ; Ruehrup, Stefan: Safe Autonomous Transport Vehicles in Heterogeneous Outdoor Environments. In: *Leveraging Applications of Formal Methods, Verification, and Validation - International Workshops, SARS 2011 and MLSC 2011, Held Under the Auspices of ISoLA 2011* Bd. 336. Berlin, Heidelberg : Springer-Verlag, 2012 (Communications in Computer and Information Science), S. 61–75

[TFPL04] Tao, Yufei ; Faloutsos, Christos ; Papadias, Dimitris ; Liu, Bin: Prediction and Indexing of Moving Objects with Unknown Motion Patterns. In: *Proceedings of the 2004 ACM SIGMOD International Conference on Management of Data*. New York, NY, USA : ACM Press, 2004 (SIGMOD '04), S. 611–622

[TPD+12] Tran, Thanh T. L. ; Peng, Liping ; Diao, Yanlei ; McGregor, Andrew ; Liu, Anna: CLARO: modeling and processing uncertain data streams. In: *The VLDB Journal* 21 (2012), Oktober, Nr. 5, S. 651–676

[TPL+10] Tran, Thanh T. L. ; Peng, Liping ; Li, Boduo ; Diao, Yanlei ; Liu, Anna: PODS: A New Model and Processing Algorithms for Uncertain Data Streams. In: Elmagarmid, Ahmed (Hrsg.) ; Agrawal, Divyakant (Hrsg.): *Proceedings of the 2010 international conference on Management of data - SIGMOD '10*. New York, NY, USA : ACM Press, 2010, S. 159

[W+88] White, F u. a.: A model for data fusion. In: *Proceedings of the 1st National Symposium on Sensor Fusion* Bd. 2, 1988, S. 149–158

[Wal99] Wald, L.: Some terms of reference in data fusion. In: *IEEE Transactions on Geoscience and Remote Sensing* 37 (1999), May, Nr. 3, S. 1190–1193

[WJKC12] Wehs, Thorsten ; Janssen, Manuel ; Koch, Carsten ; Cölln, Gerd von: System architecture for data communication and localization under harsh environmental conditions in maritime automation. In: *Proceedings of the 10th IEEE International Conference on Industrial Informatics (INDIN)*. Los Alamitos, CA, USA : IEEE Computer Society, 2012, S. 1252–1257

[WLLW13] Wang, Yijie ; Li, Xiaoyong ; Li, Xiaoling ; Wang, Yuan: A survey of queries over uncertain data. In: *Knowledge and information systems* 37 (2013), Nr. 3, S. 485–530

[YDM12] Ye, Juan ; Dobson, Simon ; McKeever, Susan: Situation identification techniques in pervasive computing: A review. In: *Pervasive and Mobile Computing* 8 (2012), Nr. 1, S. 36–66

[ZDI12] Zhang, Haopeng ; Diao, Yanlei ; Immerman, Neil: Recognizing patterns in streams with imprecise timestamps. In: *Information Systems* (2012)

[ZDM03] Zaveri, M.A. ; Desai, U.B. ; Merchant, S.N.: Tracking multiple maneuvering point targets using multiple filter bank in infrared image sequence. In: *IEEE International Conference on Multimedia and Expo* 2 (2003), S. 369–372

[Zha94] Zhang, Lianwen: Representation, independence, and combination of evidence in the Dempster-Shafer theory. In: *Advances in the Dempster-Shafer theory of evidence*. New York, NY, USA : John Wiley & Sons, Inc., 1994, S. 51–69

Index