Information Exchange and Fusion in Dynamic and Heterogeneous Distributed Environments

Dissertation
in partial fulfillment of the requirements for the degree

Doktor der Naturwissenschaften (Dr. rer. nat.)

Faculty of Electrical Engineering and Computer Science
University of Kassel, Germany

Kassel 2010

Dipl.-Inf. Roland Reichle

Advisors:
Prof. Dr. Kurt Geihs
Prof. Dr. Stefan Jähnichen

Additional Doctoral Committee Members:
Prof. Dr. Klaus David
Prof. Dr. Claudia Fohry

Date of Defense: 20th October 2010

Contents

List of Figures

List of Tables

Nomenclature

AAS .... Abstract Architecture Specification

CoOL .... Context Ontology Language

CPT .... Conditional Probability Table

CrHMM .... Credal Hidden Markov Model

DAG .... Directed Acyclic Graph

DMC .... Data Management Container

DRC .... Disjunctive Rule of Combination

DST .... Dempster-Shafer Theory of Evidence

FIPA .... Foundation for Intelligent Physical Agents

GBT .... Generalized Bayesian Theorem

GMM .... Gaussian Mixture Model

HMM .... Hidden Markov Model

IRO .... Inter-Representation Operation

MDSD .... Model-Driven Software Development

OWL .... Web Ontology Language

PDF .... Probability Density Function

PEIS .... Physically Embedded Intelligent System

PIM .... Platform-Independent Model

PSM .... Platform-Specific Model

SPICA ML .... SPICA Modelling Language

TBM .... Transferable Belief Model

Abstract

Context awareness, dynamic reconfiguration at runtime and heterogeneity are key characteristics of future distributed systems, particularly in ubiquitous and mobile computing scenarios. The main contributions of this dissertation are theoretical as well as architectural concepts facilitating information exchange and fusion in heterogeneous and dynamic distributed environments. Our main focus is on bridging the heterogeneity issues and, at the same time, considering uncertain, imprecise and unreliable sensor information in information fusion and reasoning approaches. A domain ontology is used to establish a common vocabulary for the exchanged information. We thereby explicitly support different representations for the same kind of information and provide Inter-Representation Operations that convert between them. Special account is taken of the conversion of associated meta-data that express uncertainty and impreciseness. The Unscented Transformation, for example, is applied to propagate Gaussian normal distributions across highly non-linear Inter-Representation Operations. Uncertain sensor information is fused using the Dempster-Shafer Theory of Evidence as it allows explicit modelling of partial and complete ignorance. We also show how to incorporate the Dempster-Shafer Theory of Evidence into probabilistic reasoning schemes such as Hidden Markov Models in order to be able to consider the uncertainty of sensor information when deriving high-level information from low-level data. For all these concepts we provide architectural support as a guideline for developers of innovative information exchange and fusion infrastructures that are particularly targeted at heterogeneous dynamic environments. Two case studies serve as proof of concept. The first case study focuses on heterogeneous autonomous robots that have to spontaneously form a cooperative team in order to achieve a common goal. The second case study is concerned with an approach for user activity recognition which serves as baseline for a context-aware adaptive application. Both case studies demonstrate the viability and strengths of the proposed solution and emphasize that the Dempster-Shafer Theory of Evidence should be preferred to pure probability theory in applications involving non-linear Inter-Representation Operations.

Zusammenfassung

Kontextbewusstsein, dynamische Rekonfiguration zur Laufzeit und Heterogenität sind wesentliche Charakteristiken von zukünftigen verteilten Systemen, insbesondere in den Bereichen Ubiquitous Computing und Mobile Computing. Der Hauptbeitrag dieser Dissertation besteht aus generischen theoretischen Konzepten und den sich daraus ergebenden architekturellen Implikationen, die den Austausch und die Fusion von Informationen in heterogenen und dynamischen verteilten Umgebungen ermöglichen. Unser Hauptaugenmerk liegt hierbei in der Überbrückung von Heterogenität und zugleich in der Berücksichtigung von unsicherer, unpräziser und unzuverlässiger Sensorinformation in verschiedenen Fusions- und Reasoningverfahren. Eine Ontologie wird dazu benutzt, um ein allgemeines Vokabular für die auszutauschende Information zu etablieren. Dabei werden explizit verschiedene Repräsentationen für denselben Informationstyp unterstützt und entsprechende Inter-Repräsentations-Operationen definiert, die eine Konvertierung zwischen den verschiedenen Repräsentationen erlauben. Spezielles Augenmerk erfordert hierbei die Transformation der assoziierten Metadaten, die zur Angabe der Unsicherheit und der Präzision der Information herangezogen werden. Beispielsweise wird die Unscented Transformation dazu verwendet, um multivariate Normalverteilungen über nichtlineare Transformationsfunktionen hinweg zu propagieren. Zur Fusion von unsicheren Informationen wird die Dempster-Shafer-Evidenztheorie eingesetzt, da diese eine explizite Modellierung von partieller und totaler Unwissenheit ermöglicht. Es wird auch gezeigt, wie die Dempster-Shafer-Evidenztheorie in probabilistische Reasoningverfahren, wie zum Beispiel in Hidden-Markov-Modelle, integriert werden kann, um Unsicherheiten auch bei der Ableitung von High-level-Informationen aus Low-level-Informationen berücksichtigen zu können. Für all diese theoretischen Konzepte wird eine abstrakte Architektur beschrieben, die Entwicklern von innovativen Infrastrukturen für den Austausch und die Fusion von Informationen in dynamischen und heterogenen Umgebungen als Richtlinie dienen soll. Zwei Fallstudien dienen als Proof of Concept. Die erste Fallstudie befasst sich mit unabhängig entwickelten und heterogenen autonomen Robotern, die sich zur Laufzeit spontan zu einem kooperierenden Team formieren sollen. Die zweite Fallstudie beschäftigt sich mit einem Ansatz zur Erkennung von Aktivitäten, welche die Grundlage für eine kontextbewusste, adaptive Anwendung darstellt. Beide Fallstudien zeigen die Stärken unseres Ansatzes auf und verdeutlichen, dass die Dempster-Shafer-Theorie der Wahrscheinlichkeitstheorie bei der Modellierung von Unsicherheiten vorgezogen werden sollte, wenn möglicherweise nicht-lineare Inter-Repräsentations-Operationen notwendig sind.

Acknowledgements

The venture of working in the European research projects MADAM and MUSIC, launching a middle size RoboCup team and writing a PhD thesis all at the same time would not have been possible without the help and encouragement of many enthusiastic and devoted people.

I am very grateful to my doctoral advisor *Kurt Geihs* who gave me a lot of opportunities and much freedom to work on different areas of research. Several times I was lost in a forest of too many preliminary ideas, but he has always managed to bring me back on a straight line and never stopped encouraging me. I would also like to thank *Stefan Jähnichen*, who agreed to act as external reviewer of this thesis despite his very packed time table. He also gave me the opportunity to present my work in a great environment at his chair in Berlin with a lot of fruitful discussions.

Working at the Distributed System Group at the University of Kassel was a great experience. My colleagues have not only been a constant source of inspiration for the last five years, but also we got to be friends.

Philipp Baer was the perfect partner for launching a RoboCup team. Without his competence and engagement the Carpe Noctem project would have not been possible. I was really enjoying the great time with you, Philipp! Now, *Hendrik Skubch*, *Dominik Kirchner* and *Daniel Saur* are responsible for the team. They have many great ideas for the further development of the robots and I count myself lucky that I was able to share my fascination on robots with them.

Together with *Michael Wagner*, *Mohammad 'Titu' Khan* and *Christoph Evers*, I struggled side by side to finish the deliverables for MADAM and MUSIC. Only because of the great atmosphere it was bearable to write a series of documents of more than hundred pages each. Besides, Michael got never tired to listen to my ideas and has always provided valuable feedback. Thank you, Michael!

With *Thomas Weise* I found a colleague and friend which showed an incredible dedication to bring forward a project. His meticulousness was a constant motivation for me to improve my work and his jokes often illuminated a cloudy afternoon when I got stuck with my work.

Diana Comes, *Steffen Bleul*, *Jochen Fromm* and *Michael Zapf* were always present when I needed help and were always open for fruitful discussions. Together with our administrative members *Iris Roßbach* and *Thomas Kleppe* they contributed to a familiar atmosphere in our group.

Really big thanks also goes to all the students in the Carpe Noctem Team, who significantly contributed to the success of the project: *Christof Hoppe*, *Jens Wollenhaupt*, *Kai Baumgart*, *Stefan Triller*, *Till Amma*, *Florian Seute*, *Andreas Witsch*, *Andreas Scharf*, *Martin Segatz*, *Stephan Opfer*, *Janosch Henze*, *Stefan Niemczyk*, *Jens Schreiber*, *Timo Heumüller*, *Philipp Burghardt* and *Sven Steenbock*. It was great to observe your capabilities and competences constantly growing over the years, and it was just fun to hang around with you, guys!

Besides, I don't want to forget *Alexander Kohout* and *Michel Helms*, who did a great job in supporting me with my case study. Thanks also to you, Alex and Michel!

It was also a great experience to work with the members of the MADAM and MUSIC consortium. In particular, I would like to thank *Nearchos Paspallis* for many discussions and valuable feedback. It was a great pleasure to work with you on the MUSIC Context Middleware, Nearchos!

Last but not least I am most grateful to my parents, my sister and my girlfriend, who endured many work-through weekends, never stopped to encourage and to believe in me and provided moral support whenever it was needed.

Part I

Foundations

1 Introduction

1.1 Motivation

'Heterogeneity of hardware and software is a fact in most distributed computing environments', as Geihs and Hollberg have written in an article on the DACNOS project in 1990 [47]. The focus of DACNOS was to provide a network operating system that enables resource sharing between heterogeneous autonomous computers in a distributed environment consisting of hardware and software from different vendors. Since then, a number of middleware platforms [96, 44], communication protocols [149, 157] and platform-independent data exchange formats [151, 69] have been developed to efficiently communicate and share resources in heterogeneous distributed environments.

Today, computing environments impose even more challenges compared to the target environment of DACNOS [46, 48]. Information providers and consumers dynamically appear and disappear in the environment and not all of them are known at design time. Besides, the involved software components and systems must be expected to be independently developed with minimal or no interaction between the different development teams. The resulting heterogeneity issues have to be handled at runtime in order to allow the dynamic integration of information providers and consumers into a single system. Furthermore, dynamic reconfiguration, context awareness and incorporation of imprecise, uncertain and unreliable sensor information are additional key characteristics of current distributed computing environments.

For example, *Ubiquitous Computing* [147] involves a potentially high number of diverse computing devices ranging from large computers to small invisible processing units contained in objects we use in activities of our daily life. These devices communicate and exchange data over a wireless network and may utilize services available in the environment. Mobility, context awareness and dynamic reconfiguration are inherent problems in such a scenario. Mobility of the user implies confrontation with different context situations and different properties of the computing environment, the devices and applications have to cope with by dynamic reconfiguration in order to always provide an appropriate quality of service. All the context changes, either with regard to the user or with regard to the computing environment, have to be detected by appropriate sensors or derived by reasoning approaches. Context sensors may be running on the mobile device itself or may be available as services in the environment. Sensor information for real-world entities, however, is often imprecise, uncertain and unreliable. Bridging heterogeneity issues of the involved sensors and reasoning with heterogeneous information is difficult in such dynamic environments, as the involved devices and context services may not be known at design time, and thus, interoperability has to be achieved at runtime. Therefore, an appropriate context management and reasoning system is required which facilitates the dynamic integration of context providers and consumers available in the environment, which is able to cope with the heterogeneity issues at runtime, and which considers the imperfect nature of context information, all at the same time.

Coming from a completely different direction, another example for a heterogeneous distributed computing environment, which also shows the characteristics and requirements identified above, is a cooperating team of heterogeneous autonomous mobile robots for large-scale search and rescue scenarios. In such scenarios a great number of robots with different physical and cognitive capabilities may be required. However, it is impractical to develop large teams or teams of expensive robots at a single site. Thus, we have to assume that a number of different and independently developed robot platforms have to form a team and cooperate in order to achieve the overall goal. Here too, heterogeneity issues have to be handled at runtime as in an emergency situation there may be insufficient time to hand-engineer information exchange and fusion before task execution. In order to successfully cooperate in a team of robots, it is not only necessary to communicate the current goals and tasks performed by a robot but also to establish a common view on the world. Thus, the robots also have to exchange sensor data, e.g. on observed objects. Fusion and reasoning approaches have to be realized that are able to incorporate heterogeneously represented sensor information. Dynamic reconfiguration is also required as communication links may become available or unavailable, robots break down or new robots enter the team. Here too, a framework is needed which facilitates the information exchange between a priori unknown information providers and consumers dynamically appearing in the environment and which enables a fusion of and the reasoning with heterogeneously represented sensor data.

1.2 Problem Statement

The general objective of the work presented in this thesis is to provide a comprehensive approach for information exchange and fusion in dynamic heterogeneous distributed environments which consist of dynamically appearing/disappearing and independently developed information providers and consumers. This comprises theoretical concepts as well as architectural support that can serve as guidelines for the implementation of a corresponding framework. The focus of the work also lies on providing approaches to reason with heterogeneous, imprecise, uncertain and unreliable information. Dynamic reconfiguration has to be addressed due to the dynamic nature of the computing environment, which has impact on the availability or quality of the information required as input for the corresponding reasoning tasks. However, reconfiguration of applications or adaptation of team strategies is out of the scope of this thesis. As a basis for our work, we assume an underlying communication protocol [157] or communication infrastructure [3, 6] that allows the integration of information providers and consumers realized on different platforms and in different programming languages to a distributed application. In this respect, heterogeneity issues with regard to the involved platforms and basic communication are already considered to be resolved. Nevertheless, there are many remaining challenges that form the work plan for this thesis:

- **Semantic discovery and dynamic integration of independently developed information providers and consumers.** The key challenge to deal with in a dynamic heterogeneous distributed computing environment is the dynamic appearance and disappearance of devices and services, which act as information providers and consumers. Not all the devices and services constituting the computing environment may be known at design time. Thus, runtime mechanisms are required to reason about and to perform the mediation tasks that are needed to handle the heterogeneity issues arising from the independent development of the involved services.

- **Heterogeneity of data representations.** The independent development of information providers and consumers implies that each development team utilizes the most suitable platform and technology for its task, but also names and represents the data according to its needs. Even if platform-independent data exchange formats, e.g. based on XML, are used, this results in naming conflicts and in heterogeneous representations of the data to be exchanged. For example, the location of the user in a ubiquitous computing environment may be given in *GPS Coordinates* or as *Address of a Building*. A common vocabulary has to be defined that allows to semantically interpret the meaning and representation of the data. This is a prerequisite to reason about the needed mediation tasks to achieve interoperability. In particular, this also comprises the conversion between different data representations, as for example the conversion of *GPS Coordinates* to an *Address of a Building*.

- **Expressing information offers and needs.** In order to establish communication links between information consumers and providers in a dynamic fashion, information offers and needs also have to be expressed based on a common vocabulary as mentioned in the previous bullet point. Language support is required to allow elaborate specification of information needs, to filter out inappropriate information offers and to establish only those communication links that provide the information actually needed.

- **Competitive fusion of heterogeneous sensor information.** The dynamic appearance and disappearance of information providers not only implies that actually required information may not be available all the time but also that several services may provide the requested information. In the case of sensors for real-world entities, which are often imprecise, uncertain and unreliable, conflicting information has to be expected in such a situation. Competitive sensor fusion schemes based on probabilities and belief functions to resolve conflicts and to come to a coherent value are widely available [115, 125, 124, 89, 82]. However, new challenges arise from the heterogeneity issues of the involved sensors that have to be resolved at runtime. In particular, runtime mechanisms are needed that are able to reason about and to automatically perform the mediation tasks that are required to maintain the meta-data expressing the impreciseness, uncertainty and unreliability of sensor data across the conversion of representations. For example, the impreciseness of an object's location estimation given through a covariance matrix has to be maintained across maybe highly non-linear representation conversions.

- **Reasoning with heterogeneous, imprecise, uncertain and unreliable information.** In many cases, required information cannot be directly sensed but has to be derived from other sensor information by appropriate reasoning mechanisms. For example, consider the situation of a user in a ubiquitous computing environment. There are no sensors available to directly determine whether the user is currently working, shopping or having dinner. Instead, logical or probabilistic reasoning schemes are applied to derive such high-level information from low-level sensor data. Here too, the heterogeneous nature of the involved sensors entails further challenges. Before providing appropriate input for the reasoning schemes, representation conversions and competitive sensor fusion have to be performed. Or viewing it from another perspective, reasoning schemes are required that are able to deal with information resulting from the applied representation conversion and sensor fusion steps.

1.3 Solution Approach

Figure 1.1 provides an overview of the proposed approach. The baseline of the overall approach is an Information Model in terms of an ontology specified in OWL [155]. It defines the basic semantic concepts *Entity Types*, *Scopes* and *Representations*. *Entities* represent real or logical objects of the world that are characterized by a piece of information. As with the usual typing concept, an *Entity Type* abstracts over a set of *Entities*, which are considered as instances or individuals of an *Entity Type*. A *Scope* defines a semantic concept for the type of the provided information and the *Representation* specifies how the information is represented. Apart from these basic concepts, the ontology also defines *Inter-Representation Operations* (IROs) [135] that describe the conversion of one representation of a certain scope to another representation of the same scope. With the help of these concepts, a common vocabulary is established that allows to interpret the meaning and representation of the exchanged data and to reason about the availability of appropriate IROs for the conversion between different representations.

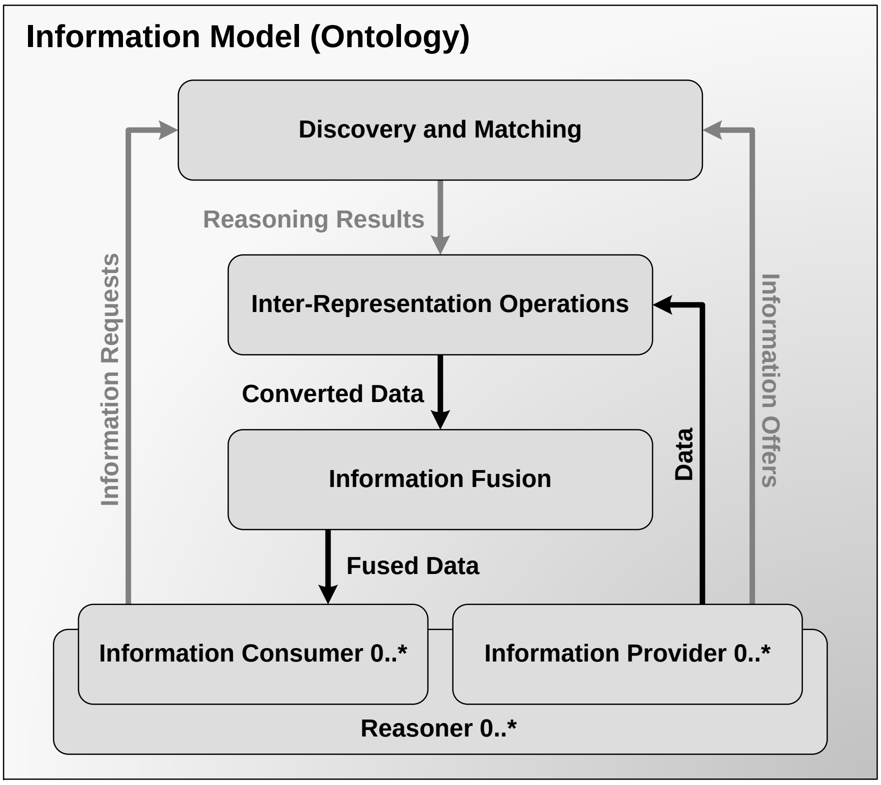

Figure 1.1: Overview of the Solution Approach

As highlighted in the previous section, the focus of our work is also on reasoning with heterogeneous information. We assume that there is an arbitrary number of reasoning tasks to be performed. Each reasoner acts as information consumer and as information provider. The required and offered information is specified on the common vocabulary defined by the ontology utilizing the newly developed *Information Offer and Request Language* (IORL). It supports elaborate filter mechanisms to precisely define and constrain the offered and requested information. The corresponding semantic definitions serve as input for the *Discov-*

ery and Matching approach. Matching of information offers and requests already includes reasoning about potentially required mediation tasks in terms of IROs to be performed for bridging mismatches in the provided and requested representations.

The reasoning results of the Discovery and Matching approach are used to establish communication links to (and maybe activate) the corresponding information providers. There may be multiple providers for the same requested *Scope*, but they are likely to provide the information in a different representation as requested. However, by means of the previous reasoning step it is ensured that only links to such providers are established, where mismatches between offered and requested information can be resolved. When applying the corresponding IROs, special care has to be taken to maintain the meta-data expressing the impreciseness, uncertainty and unreliability. As a developer should not be bothered to deal with such meta-data but should only be required to develop a simple value-to-value conversion routine, we employ sampling mechanisms like the *Unscented Transformation* [70] for maintaining the associated meta-data.

When the different pieces of information are given in the same representation, the *Dempster-Shafer Theory of Evidence* (DST) [126, 27] is utilized to fuse the different sensor inputs. The DST was selected as underlying sensor fusion scheme as it allows to specify partial or complete ignorance and is well-suited to combine information of very different granularity, e.g. *User Location in GPS Coordinates* and *User Location in a Building*.

As Dempster-Shafer belief assignments are the result of the information fusion step, reasoning schemes are required which integrate the DST as well. In scenarios as depicted above, common reasoning schemes are based on logics or on Bayesian probability theory. Examples for the latter case are *Naïve Bayes Classifiers*, *Hidden Markov Models* or *Polytree Bayesian Networks*. These reasoning schemes can be integrated with the DST based on the *Transferable Belief Model* (TBM) [130] in a straight-forward manner.

1.4 Contribution

The research work presented in this document has led to a number of contributions to the state of the art. Although the proposed approach is designed to be generally applicable to dynamic heterogeneous environments, in which fusion of and reasoning with imprecise, uncertain and unreliable sensor information is required, in particular contributions have been made in the area of *Context Management and Reasoning in Ubiquitous Computing* and in the area of *Cooperative Teams of Heterogeneous Mobile Robots*. In this respect, these two target environments have not been selected arbitrarily but from the observation that the two areas can profit from each other by combining the strengths of both to a comprehensive solution approach. Concretely, we see four major contributions:

1. **A comprehensive solution** for information exchange and fusion in dynamic heterogeneous distributed environments has been designed, combining a number of ingredients available in the areas of *Context Management and Reasoning* and *Autonomous Mobile Robots*.
2. **An elaborate context modelling approach with focus on heterogeneity of context sensors and reasoners** is presented in terms of the ontology-based Information Model and the associated IORL. The corresponding concepts form the basis of the context model utilized in the European research project MUSIC [90].

3. **A new generally applicable context aggregation and reasoning method** is proposed which is focused on the incorporation of heterogeneous sensor information and is based on well-established theories.
4. **New concepts for the dynamic formation of cooperative teams of heterogeneous autonomous robots** are presented that allow bridging heterogeneity at the information representation level.

All the claimed contributions will be justified when we analyze related work (Chapter 6) and present the solution approach (Part II).

1.5 Structure of the Thesis

This document is structured into three parts. The foundations of the proposed solution are discussed in the remainder of the first part. This includes information exchange in heterogeneous environments (Chapter 2), fusion of (Chapter 3) and reasoning with imprecise, uncertain and unreliable sensor information (Chapter 4), basic concepts of an ontology-based specification of application domains (Chapter 5) as well as the analysis of related work (Chapter 6).

The main contribution of this thesis is contained in Part II of the document. In Chapter 7, we introduce the underlying Information Model along with the associated IORL. The realization of IROs and the applied techniques to maintain the measures for impreciseness, uncertainty and unreliability across the transformations is described in Chapter 8. Chapter 9 presents our fusion method for heterogeneous sensor information based on the DST. The corresponding reasoning schemes that are able to make use of the results of the sensor fusion step are presented in Chapter 10. The combination of the proposed theoretical concepts to a comprehensive framework leads to a number of implications for its architectural design. These implications are elaborated in Chapter 11.

The general purpose of Part III of the document is to evaluate and to critically assess the proposed concepts. Two case studies are provided to evaluate its practical feasibility. The first real-world case study presented in Chapter 12 is centered around a dynamically composed team of autonomous soccer robots, where the different team members communicate the information of their world models involving heterogeneous representations. We show how this team of robots can come to a coherent estimation of the ball position on the field incorporating imprecise, uncertain and unreliable sensor information. Such a coherent estimation of the ball position is the prerequisite to coordinate team play within a soccer game. Chapter 13 presents the second case study, which is based on a simulated scenario for ubiquitous computing environments. User activity recognition has to be performed to adapt the applications on a mobile device according to the current user situation. In particular, the impact of different quality levels of the incorporated sensor information and the effects of using the DST instead of traditional probabilities on the reasoning result is investigated. Finally, Chapter 14 concludes this dissertation and presents an outlook to future work.

2 Information Exchange in Heterogeneous Environments

2.1 Information and Heterogeneity

The term *information*, derived from Latin *informare* in the meaning *to give form (to the mind)*, is widely used in our daily life and plays an important role in basically all areas of research: physics, computer science, cybernetics, electrical engineering, business economics, social sciences, etc. Dependent on the particular perspective, different characteristics of information are highlighted and consequently its definitions vary to some extent. Thus, the term information is difficult to capture and it is important to clarify our understanding in the context of this thesis. For this purpose, we start with a list of characteristics of *information*, which are collected from [127], [94], and [148] and reflect our understanding.

- information **reduces uncertainty**[1] and thus **leads to new knowledge** about real or logical entities, events, processes or states of the world.
- information **can be transmitted/communicated** in the form of **signals or data**.
- perception of information may lead to a **change of the receiver's state** comprising its knowledge, decisions, actions.

It is noteworthy here that in information theory [127] the mean mutual information of two random variables is always greater or equal to zero and thus reduces uncertainty or has no influence on it. As people usually do not refer to the mean mutual information of two random variables, however, this does not necessarily correspond to the common intuitive understanding, where information contained in a message may also cause an increase of uncertainty. From this consideration and the characteristics above, we derive the following definition of the term information as used in this thesis.

Definition 1 *Information is a metaphor for everything that is encoded in the data of a message transmitted/communicated from a sender to a receiver, influences uncertainty of a certain estimate and may lead to new knowledge at the receiver's site useful to derive decisions, actions or new state estimates. Propagation between different levels of abstraction is also considered as transmission/communication between a sender and receiver.*

Information about real entities, events and processes of the world is usually provided by some kind of sensors. These encode the information contained in (accustic, electromagnetic, etc.) signals in a machine-processable representation. Possible representations range from

[1] A measure of variety which is zero when all elements are in the same category and increases with both the number of categories and their equiprobability [146].

a pure voltage level to more abstract data representations, possibly involving a number of basic preprocessing and feature extraction steps. In our work we focus on information which is already given in such a kind of abstract data representation. This leads us to the following understanding of sensor information as used in this thesis.

Definition 2 *Sensor information is information that characterizes real entities, events and processes of the world and is provided by devices (sensors) which measure physical quantities, transform them in signals and encode the signals in a data representation (possibly performing some basic preprocessing and feature extraction steps).*

In contrast to information, the term *heterogeneity* can be captured much easier. *Heterogeneity* is the noun derived from *heterogeneous* (from Greek *heteros* 'other' and *genos* 'a kind'), which means diverse in character or content.[2] However, for the purpose of this thesis it has to be clarified what kind of heterogeneity with regard to a distributed computing environment we aim to address. In the following list, which is not claimed to be exhaustive, we have identified different characteristics of today's distributed computing environments that impose additional challenges due to heterogeneity.

- A distributed computing environment may comprise a number of different physical devices, which implies heterogeneity with regard to the properties of the involved physical platforms.
- The nodes may run different operating systems and the corresponding software components, modules or services, may be realized using different programming languages, which results in heterogeneity issues with regard to the basic software platform.
- Communication in a distributed computing environment can impose heterogeneity issues due to the usage of various networking technologies.
- Additional challenges may also be implied by heterogeneous communication protocols and serialization/deserialization schemes utilized by the software components, modules and services.
- Independent development of software components, modules and services may result in mismatches in the expected and provided representation of the information. This involves, for example, the names used for entities and/or information types as well as the employed coordinate systems and units of measure.

The focus of this thesis is on bridging heterogeneity of the utilized data representations that result from the independent development of the involved components, modules and services as described in the last bullet point. Heterogeneity issues resulting from different software platforms, network technologies, and communication protocols are assumed to be already solved by an underlying communication framework.

2.2 Communication in Heterogeneous Distributed Environments

Dependent on the type of heterogeneity issues to be addressed, different possibilities are available to realize the communication among the nodes consituting a distributed computing

[2] In Compact Oxford English Dictionary, http://www.askoxford.com/ (accessed 2010-04-23).

environment. Heterogeneity with regard to network technologies, underlying software platforms and programming languages is often bridged by employing a standard communication protocol, which is expected to be supported by all the involved software systems and network technologies.

2.2.1 Web Services

The realization of distributed applications using *Web services* exactly follows this idea. Web services are software applications, which are uniquely identified by a *Uniform Resource Identifier* (URI) and whose interfaces are described in a platform-independent language, usually in WSDL [158]. Direct interaction with other software components, agents or applications is realized through the exchange of messages, which are also represented in a standard platform-independent language, via standard internet protocols. Here, the most frequently used message representation is SOAP [157] and the usually employed protocol is HTTP. In fact, Web services can be realized on top of arbitrary data exchange protocols. However, HTTP and other internet based protocols have the advantage that communication is usually allowed also across networks protected through firewalls without additional configuration effort. In this respect, Web services support the interoperability between different applications realized on various software platforms and frameworks. The usage of heavy-weight XML-based standards, however, requires extensive message parsing and leads to a big message size. Although this is partly avoided by Web services that are based on REST (REpresentational State Transfer) [35] which does not rely on XML, Web services are not the first choice in scenarios where real-time or soft real-time capabilities of the communication framework are required. An example for such a scenario is a team of heterogeneous autonomous mobile robots.

2.2.2 Common Object Request Broker Architecture (CORBA)

Several middleware frameworks for autonomous mobile robots [141, 142, 17] are based on the Common Object Request Broker Architecture (CORBA) [96]. CORBA is a specification for an object-oriented middleware that defines platform-independent protocols and services. One of its central concepts is the so-called Object Request Broker (ORB), through which the application interacts with other locally or remotely available objects. The Interface Definition Language (IDL) allows for a formal specification of the interfaces that a server application provides for local or remote access. The corresponding IDL definitions are then translated by an IDL Compiler in an object model of the used programming language, which results in source code containing stubs and skeletons. The stubs of the generated classes can now be used from the client application in the same way as local objects hiding all the complexity of the remote method calls. The CORBA specification is not bound to a specific programming language or a specific platform. For example, there exist implementations in C, C++ and Java, and standard mappings from IDL to these programming languages are available. Communication within a CORBA implementation was usually realized with a developer-specific protocol. With CORBA 2.0, the General Inter-ORB Protocol (GIOP) was introduced, which supports communication among different CORBA implementations. Here, most widely used is the Internet Inter-ORB Protocol (IIOP), which is a mapping of GIOP to internet protocols. In this respect, CORBA enables the integration of heterogeneous software components written in different programming languages and running on different platforms to a distributed application.

In comparison to Web services, CORBA not only prescribes a standard message representation scheme and communication protocol for bridging heterogeneity issues, but also a rather complete architecture. In a similar way, FIPA specifications (see also Section 6.2.1) define an Abstract Architecture Specification (FIPA-AAS) [37] along with an Agent Communication Language (FIPA-ACL) [38], message content languages and message representation schemes to guarantee interoperability among heterogeneous agents if the corresponding agent frameworks are implemented according to these standards.

2.2.3 SPICA

The *SPICA* development framework [3, 6, 4, 5] is based on the model-driven software development (MDSD) [144] paradigm and allows an automatic generation of communication infrastructures enabling the integration of heterogeneous software modules in a distributed environment from abstract specifications. Here, heterogeneity is caused through independent development of the software modules in different programming languages.

In general, MDSD prescribes the specification of the software component or system to be realized in an abstract and platform-independent model (PIM) using general purpose or domain-specific modelling languages. The PIM is then transformed by appropriate transformation tools in one or more steps to a platform-specific model (PSM) that takes into account the special characteristics and properties of the target platform. In a second transformation, source code for the specified software component/system is generated from the PIM. In this respect, MDSD exhibits a general mean to cope with heterogeneity between various platforms by providing a platform-independent model that can be transformed in different implementations tailored to each of the involved platforms. Here, the type of heterogeneity issues addressed obviously depends on the definition of what is understood by the term platform: e.g. operating system, middleware or programming language. In CORBA, the specification of service interfaces in IDL and the generation of platform-specific stubs and skeletons from these definitions with the help on an IDL compiler can be considered as a kind of MDSD approach as well.

An overview of the SPICA development framework is shown in Figure 2.1. SPICA provides a very lightweight approach and is tailored to support modularity, efficiency, heterogeneity and robustness of the communication infrastructure to be developed. Its central parts are the SPICA modelling language (SPICA ML) and the Aastra model transformation tool. SPICA ML provides specification means to define the messages, the modules constituting the modular architecture of the envisaged system, their data management capabilities and the data flow between the modules. With regard to message definition, SPICA ML allows the modelling of hierarchically structured network messages similar to but much more application domain-oriented than ASN.1 or IDLs. It supports a type concept with strong typing and inheritance. References to an ontology are envisaged in order to base the message and field definitions on a common vocabulary. However, there is no concrete ontology or ontology meta-model provided, and thus an underlying information model is missing.

The modules are defined by their provided and required message types and their data management containers (DMCs). DMCs are for example ring buffers, arrays or queues, and

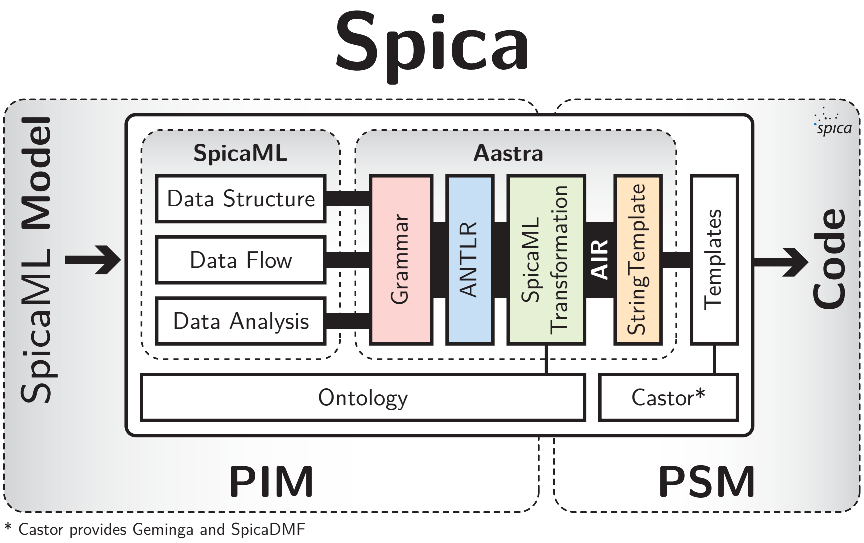

Figure 2.1: Overview of the SPICA Development Framework

used to store asynchronously received messages. Here, SPICA allows to define callback methods, which are invoked when a message is received. DMCs for provided message types can be configured to send the data periodically or a transmission is triggered if a new element is inserted into the corresponding DMC. The data flow between the modules is implicitly defined by the provided and required message types of the modules and established by the resource discovery and matching component *Geminga*. Geminga is a component of the SPICA framework and contained in the *Castor* library. Castor provides implementations of the required communication capabilities as well as the SPICA data management framework (SpicaDMF) and is available in different programming languages such as C++ and C#.

Model transformation and code generation in SPICA are realized with the help of the ANTLR parser generator [1] and the String Template template engine [105]. The code generation results in data structures for the messages along with their serialization and deserialization methods as well as in skeletons for the modules in the desired programming languages. The skeletons serve as proxies on top of existing modules and allow an easy integration of heterogeneous modules with minimal manual programming effort by providing easy-to-use interfaces to exchange data with the DMCs.

In principle, the approach for information exchange and fusion proposed in this dissertation can be realized on top of a Web service-based infrastructure, or using CORBA or SPICA. In Chapter 7, we will shortly discuss how our information model integrates with the SPICA development framework, in particular with SPICA ML. A SPICA-based communication infrastructure will also serve as baseline for our real-world case study (see Chapter 12) that evaluates the feasibility of our approach for a cooperative team of autonomous robots.

3 Fusion of Imprecise and Uncertain Information

3.1 The Imperfect Nature of Sensor Information

Sensors measure physical quantities and convert them into signals serving as input for further processing steps to derive state estimates for the observed entities. Due to physical limitations of the sensors (e.g. the resolution, color depth/format or frame rate of a camera), sensor malfunction, noise, latencies and interferences, sensor information is usually imperfect. In this dissertation, we address the imperfect nature of sensor information with respect to the following four aspects:

- **Uncertainty:** Processing of sensor data often results in several disjunct hypotheses for an entity's state. For example, if a camera is used to locate an arbitrarily colored soccer ball on a soccer field, several regions in the image may show nearly round shapes with approximately the size that can be expected for the ball. In this case, the sensor provides different hypotheses for the ball position on the field. Uncertainty among the hypotheses is often expressed by assigning probabilities to the hypotheses. In this dissertation, we also address the case where these probabilities are partly or totally unknown.
- **Impreciseness:** Even if there is only one hypothesis for an entity's state, the estimate is usually imprecise due to inherent measurement errors, noise and interferences. For example, a temperature sensor may tell 19°C for temperatures from 18.5°C to 19.4°C due to limited resolution and measurement errors. In this case, the impreciseness could be described by a value range. However, it is also very common to express impreciseness with the help of probability density functions, e.g. a Gaussian defined by its mean and covariance matrix.
- **Unreliability:** A sensor may also provide wrong estimates caused by malfunction and/or bad conditions in the environment which disturb the measurements. Therefore, the information provided by a sensor can only be assumed to be reliable to a certain extent. Reliability is often expressed through a confidence value or probability the sensor is expected to work properly and thus to provide a feasible estimate. As argued in [84], the reliability of a sensor may vary in different contexts, i.e. conditionally on different hypotheses regarding the entity state of interest.
- **Outdatedness:** Sensor information may be outdated due to the limited frequency of the performed measurements and/or latencies in signal or data transmission. In particular, if the state of the observed entity, event or process is highly dynamic, the estimates provided by a sensor are more unreliable the older they are.

In Chapter 9, we show how all these characteristics can be considered in an information fusion approach based on Dempster-Shafer belief functions.

3.2 Complementary and Competitive Information Fusion

In dynamic distributed computing environments as envisaged in this dissertation, information providers dynamically appear and disappear. This means that actually required information may not be available all the time, and also that several services may provide the requested information at the same time. In this case, a single information provider has to be selected or information fusion approaches have to be applied. Here, information fusion is understood in the following way:

Definition 3 *Information fusion is concerned with combining information from different sources with the objective to obtain more precise, less uncertain, more reliable or new state estimates.*

In this dissertation, information fusion is mainly considered with regard to data which originate from sensors, i.e. we are mainly concerned with sensor fusion.

Definition 4 *Sensor fusion is a specialization of information fusion where the information to be combined is directly provided by or originates from sensors.*

In the literature, different proposals have been presented for the categorization of sensor fusion approaches [15, 55, 75]. Here, the applied criteria include functionality [15], the addressed information level [55] or architectural properties [75]. From the perspective of functionality, Brooks and Iyengar [15] distinguish between *competitive*, *complementary*, *cooperative* and *independent fusion* approaches.

- **Competitive sensor fusion** schemes combine sensor data that represent the same perspective on or measurement of an entity's state and provide the same type of information with the objective to reduce uncertainty and to resolve conflicts. This is the basic sensor fusion type and is often regarded as the 'traditional' or 'classical' sensor fusion technique.
- **Complementary sensor fusion** methods aim at creating a more complete model by combining sensor information that represent complementary perspectives and do not depend on each other directly.
- **Cooperative sensor fusion** takes into account the fact that no single sensor but only a combination of multiple sensors may be able to provide the required information. The difference to complementary sensor fusion is that in cooperative sensor fusion the sensors directly depend on each other. An example for cooperative sensor fusion is a sterovision system to realize three-dimensional object recognition. Here, the pixels of the two images depend on each other in pairs from which the object distances can be derived.
- **Independent sensor fusion** is performed if information from multiple sensors is used for some purposes in different places in the system. This is a special case and actually does not represent sensor fusion schemes strictly referring to its definition as the information from different sensors is just used in the same system but not necessarily combined.

In conclusion, competitive sensor fusion aims at improving the quality of state estimates, whereas complementary and cooperative sensor fusion schemes are used to derive new, possibly more high-level state estimates. In this dissertation, we restrict ourselves to competitive and complementary sensor fusion. As competitive sensor fusion is considered as the traditional form, the general term sensor fusion is used in the meaning of competitive sensor fusion in the remainder of the document, whereas complementary sensor fusion is captured by the terms *reasoning with*, *aggregation of* or *integration of* sensor information. In the following sections (Section 3.3 to Section 3.5), different approaches for competitive sensor fusion are presented. Methods for reasoning with uncertain, imprecise and unreliable information are presented in Chapter 4.

3.3 Fuzzy Logic

Fuzzy logic [165, 166] has mainly been developed to accommodate impreciseness and uncertainty of descriptive terms used by humans in daily life and to allow reasoning with them. For example, the temperature of a room is often referred to as 'cold', 'warm' or 'hot'. However, there are no crisp boundaries between these three descriptive terms. For example, 19°C can be considered to a certain degree as 'cold', but also interpreted as 'warm'. Thus, fuzzy logic provides methods to deal with (sensor) information which cannot easily be separated into discrete segments, and consequently is difficult to model with conventional mathematical or rule-based systems.

Fuzzy logic is based on fuzzy sets, membership functions that establish a mapping of values to fuzzy sets, as well as appropriate logical operations for these sets and production rules for inference. In practical applications, also methods for fuzzification and de-fuzzification have to be incorporated, i.e. methods to convert values and interrelationships into a fuzzy logic representation and back.

In contrast to sets in the traditional understanding, where an element is either included in the set or not, in fuzzy set theory it is also possible that elements are included in a set only to a certain extent. The degree to which a value is part of a fuzzy set is described by the membership function of the fuzzy set, which assigns each element of the domain a real number of the interval $[0,1]$.

The well-known logical operations 'and', 'or' and 'not' as well as the set operations 'union', 'intersection', and 'complement' are defined for fuzzy sets based on a t-norm and its corresponding t-conorm. A t-norm shows by definition the properties of commutativity, associativity, monotonicity and has the identity element '1'. Usually the logical operations are defined based on the Gödel t-norm [50], which results in the following combination scheme:

$$\begin{aligned}
\text{not } x &= 1 - \text{truth}(x) \\
x \text{ and } y &= \min\{\text{truth}(x), \text{truth}(y)\} \\
x \text{ or } y &= \max\{\text{truth}(x), \text{truth}(y)\} \qquad (3.1)
\end{aligned}$$

where x and y are fuzzy variables and $\text{truth}(x)$ corresponds to the truth value of the variable x. If the logical operators are defined as in Equation 3.1 they are also called Zadeh operators.

Based on fuzzy sets and fuzzy logic, competitive sensor fusion can now be realized by intersecting the state estimates of the involved sensors represented as fuzzy sets. Logic inference is specified based on so-called production rules in the form of IF-THEN statements. This is also often referred to as fuzzy associative matrices.

Fuzzy logic provides a simple sensor fusion method, is well-suited to represent impreciseness and uncertainty inherent to terms used in humans' daily life, and thus also facilitates the integration of knowledge from human experts. However, we do not consider it as first choice for a sensor fusion scheme to be integrated in a generic framework applicable to a wide range of applications, as membership functions, production rules, and methods for fuzzification and de-fuzzification are often highly domain- and problem-specific.

3.4 Bayesian Inference

Bayesian inference is the most widely used sensor fusion scheme and is the starting point for many new methods [159]. Thus, it is often referred to as the 'classical' sensor fusion method. In Bayesian inference, a belief in a hypothesis before evidence has been collected (prior belief) is updated to a belief estimate after evidence has been observed (posterior belief). This process can be repeated when additional evidence is obtained using the posterior belief of the previous step as new prior. Usually, Bayesian inference works on degrees of belief or subjective probabilities in the induction process, and consequently does not necessarily provide an objective method of induction.

Probabilities are adjusted when new evidence becomes available according to the theorem of Bayes:

$$P(H|E) = \frac{P(E|H) \cdot P(H)}{P(E)} \tag{3.2}$$

Here, H represents a certain hypothesis and $P(H)$ the prior probability inferred before the new evidence E was obtained. $P(E|H)$ depicts the conditional probability for observing the evidence E if the hypothesis H happens to be true. If it is considered as a function of H with fixed E it is also referred to as likelihood function.

The marginal distribution of E, $P(E)$, is the a priori probability of obtaining the evidence E considering all possible hypotheses. It can be calculated by summing up the products of the probabilities of a complete set $\mathcal{H}$ of mutually exclusive hypotheses and the conditional probabilities of E given the corresponding hypothesis:

$$P(E) = \sum_{H_i \in \mathcal{H}} P(E|H_i) \cdot P(H_i) \tag{3.3}$$

In practice, however, the calculation of $P(E)$ can often be avoided as the conditional probabilities of the hypotheses given the evidence have to sum up to one. In this case, $1/P(E)$ is often abstracted through a normalization factor α. The conditional probability $P(H|E)$ corresponds to the posterior probability of H after the observed evidence has been considered.

In this update step the factor $P(E|H)/P(E)$ can be interpreted as a measure for the impact of the evidence E on the belief in the hypothesis H. If it is likely that E can be obtained if H is true, but unlikely that E can be observed at all, this factor will be quite large resulting in a higher posterior probability of the hypothesis. Conversely, in the case of a quite unlikely observation of E given H, but a high marginal probability for E, the factor is small and causes a reduction of the posterior probability of H.

As already mentioned above, this update step can be performed iteratively using the posterior of the previous step as new prior, if the evidences are marginally and conditionally independent of each other given the hypotheses. Iteratively applying Bayes' theorem for two pieces of evidence yields:

$$P(H|E_1 \cap E_2) = \frac{P(E_2|H) \cdot P(E_1|H) \cdot P(H)}{\sum_{H_i \in \mathcal{H}} P(E_1|H_i) \cdot P(E_2|H_i) \cdot P(H_i)} \tag{3.4}$$

Although Bayesian inference served as baseline for and has proven its viability in a great number of sensor fusion tasks [159], it still shows a number of disadvantages. For example, it is often difficult to define a priori probabilities or to consider multiple potential hypotheses and multiple conditionally dependent events. Furthermore, the approach requires mutual exclusivity for competing hypotheses and, as it is based on probability theory, it is not able to account for general uncertainty and partial or total ignorance.

3.5 Dempster-Shafer Theory of Evidence

The Dempster-Shafer Theory of Evidence (DST) [126, 27] is a generalization of the classical Bayesian probability theory and is also known as the *theory of belief functions*. In contrast to the classical Bayesian probability theory, DST allows explicit modelling of partial and total ignorance.

Central to DST is the concept of **belief functions**, which are based on **basic belief assignments**. Let Ω be a finite and non-empty set called **frame of discernment**. The mapping $bel : 2^{\Omega} \to [0,1]$ is a belief function iff there exists a basic belief assignment $m : 2^{\Omega} \to [0,1]$ with the following properties:

$$m(\emptyset) = 0 \tag{3.5}$$

$$\sum_{A \subseteq \Omega} m(A) = 1 \tag{3.6}$$

$$bel(A) = \sum_{B \subseteq A, B \neq \emptyset} m(B) \tag{3.7}$$

The values of $m(A)$ ($A \subseteq \Omega$) are called **basic belief masses** (or shortly masses) and are assigned to all elements of the power set of Ω. A mass assigned to a subset A of Ω with $|A| > 1$ is interpreted as belief mass, which can be given to the whole set, but for which no further justification exists to divide the belief mass into masses of more elementary subsets of A. Note that Shafer in his original model asserts $m(\emptyset) = 0$ and hence, $bel(\Omega) = 1$. As described later on, this assertion has consequences for the rules for combination and conditioning of belief functions: the results are normalized by division with appropriate scaling factors. Smets argues that $m(\emptyset) > 0$ can be justified under the open-world assumption and consequently drops the assertion of Shafer's model. The nature of $m(\emptyset) > 0$ and the differences between the two definitions are discussed by Smets in [131].

Belief functions are in one-to-one correspondence with **plausibility functions**, which describe a mapping $pl : 2^{\Omega} \to [0,1]$ where $pl(\emptyset) = 0$ and for all $A \subseteq \Omega$, $A \neq \emptyset$:

$$pl(A) = \sum_{B | A \cap B \neq \emptyset} m(B) \tag{3.8}$$

The relation between belief functions and plausibility functions is also given through:

$$pl(A) = bel(\Omega) - bel(\bar{A}) \tag{3.9}$$

Interpreting the belief of an $A \in \Omega$ as belief masses that have to be assigned to A, and the plausibility of A as the belief masses that can potentially be assigned to A, belief and plausibility form the upper and lower bound of the probability (in the classical sense) of A:

$$bel(A) \leq P(A) \leq pl(A) \tag{3.10}$$

Competitive sensor fusion deals with the problem to fuse competing information for the same question but from different sources. In the Dempster-Shafer model, competitive sensor fusion is realized through **Dempster's Rule of Combination**. It allows to combine two independent evidences given as two independent basic mass assignments m_1 and m_2 on the same frame of discernment. The **joint mass assignment** $m_{1,2}$ is calculated with:

$$m_{1,2}(\emptyset) = 0 \tag{3.11}$$

$$m_{1,2}(A) = \left(m_1 \oplus m_2\right)(A) = \frac{1}{1-K} \sum_{B \cap C = A \neq \emptyset} m_1(B) \cdot m_2(C) \tag{3.12}$$

$$K = \sum_{B \cap C = \emptyset} m_1(B) \cdot m_2(C) \tag{3.13}$$

Dempster's Rule of Combination is based on the idea to minimize conflict by assigning the masses of two subsets B and C of Ω to the intersection $B \cap C$, and to multiply the masses $m_1(B)$ and $m_2(C)$. K corresponds to the sum of masses that have to be assigned to $\emptyset$ and hence provides a measure for the remaining conflict. Forcing Equation 3.11 and maintaining the requirement given through Equation 3.6 results in a normalization with $\frac{1}{1-K}$, which actually means ignoring the conflict. This has led to serious criticism of the rule when a significant conflict is encountered in the mass assignments. As mentioned above, the model of Smets drops the assertion $m(\emptyset) = 0$ and thus does not require the normalization. The corresponding rule without normalization is called **Conjunctive Rule of Combination**.

In Bayesian probability theory the concept of conditional probability is very important. This gives rise for considering the concept of conditioning also within the Dempster-Shafer model and the introduction of **conditional belief functions**. Assume bel quantifies the belief on the frame of discernment Ω and we learn that $\overline{A} \subseteq \Omega$ is false. Then, the resulting conditional mass assignment and the corresponding conditional belief function is obtained by **Dempster's Rule of Conditioning**:

$$m(B|A) = \frac{1}{1-K} \sum_{X \subseteq \overline{A}} m(B \cup X) \qquad \text{if } B \subseteq A \subseteq \Omega \tag{3.14}$$

$$m(B|A) = 0 \qquad \text{otherwise}$$

$$K = bel(\overline{A})$$

Analogously to conditional probabilities, $bel(B|A)$ is considered as the belief of B given A or in a context where A holds. One interpretation of Dempster's Rule of Conditioning is that a mass $m(B)$ given to B is transferred by conditioning on A to $A \cap B$. In this respect, Dempster's Rule of Conditioning can be obtained without the concept of *combination of distinct pieces of evidence* and hence does not require any definition for distinctness, combination or probability. Just as for Dempster's Rule of Combination, in the model of Smets with an open-world assumption the normalization is also omitted for the rule of conditioning.

Dubois and Prade [32] proved the following relations for the combination of a conditional belief or mass assignment given by m_1 and another mass assignment m_2:

$$\begin{aligned} m_{1,2}(A) &= \sum_{B \subseteq \Omega} m_1(A|B)\, m_2(B) \\ bel_{1,2}(A) &= \sum_{B \subseteq \Omega} bel_1(A|B)\, m_2(B) \\ pl_{1,2}(A) &= \sum_{B \subseteq \Omega} pl_1(A|B)\, m_2(B) \end{aligned} \tag{3.15}$$

There are several proposals for decision making using Dempster-Shafer belief functions, e.g. [162, 136]. However, so far the theory of belief functions still lacks a coherent theory to guide the choices of lotteries in which uncertainty is described using belief functions [24]. A possible solution to this problem is to translate a belief function model to a probability model and then to apply the Bayesian decision theory for decision making. Such a transformation calculates a probability value for each element (singleton) $\omega \in \Omega$.

Smets has suggested [129] to apply this strategy and to use the **pignistic transformation** to derive probability functions from belief functions. The pignistic transformation is defined by:

$$BetP_m(\omega) = \sum_{A \subseteq \Omega} \frac{|\omega \cap A|}{|A|} \frac{m(A)}{1 - m(\emptyset)} \tag{3.16}$$

where in case of $m(\emptyset) > 0$ the value of $1 - m(\emptyset)$ acts as normalization coefficient. Intuitively, in the pignistic transformation the masses of subsets of Ω are evenly distributed among its constituting singletons $\omega \in \Omega$.

Although the pignistic transformation is the most commonly used method for translating from belief function models to probability models, Cobb and Shenoy [24] are concerned that this transformation is not consistent with Dempster's Rule of Combination. Hence, they propose the method of the **plausibility transformation**, which is defined in [24] as:

$$Pl\_P_m(\omega) = \frac{pl(\omega)}{\sum_{\upsilon \in \Omega} pl(\upsilon)} \tag{3.17}$$

Bayesian probability theory and Dempster-Shafer Theory of Evidence are the two most popular multi-sensor fusion approaches in a number of different domains. One important reason for this is that both approaches are commutative and associative with respect to the incorporation of competing sensor information.

Both, the Bayesian inference method and the Dempster-Shafer method, support the update of a priori estimates with new observations to obtain a posteriori estimates. However, the Dempster-Shafer method relaxes the restriction on mutually exclusive hypotheses of the Bayesian method as it allows to assign belief to propositions, i.e. subsets of Ω. Partial ignorance can be modelled by assigning masses to subsets of Ω, complete ignorance can be expressed by assigning a mass to the whole frame of discernment.

4 Reasoning with Uncertain Information

Numerous different approaches for reasoning with uncertain information based on fuzzy logic [76, 115], probabilistic logic [115], Bayesian probability theory [115, 124, 125] or DST [89, 82] have been proposed in the literature. In the following sections, we focus on reasoning schemes utilizing probability theory and DST, as these have been identified as the most suitable approaches for competitive sensor fusion in the previous chapter and thus fit best into a comprehensive solution providing support for fusion of and reasoning with heterogeneous and imperfect sensor information.

4.1 Probabilistic Reasoning Schemes

4.1.1 Bayesian Networks

A Bayesian network [109] is a probabilistic graphical model that provides a compact representation of the joint probability density function of a set of random variables via a directed acyclic graph (DAG) by exploiting conditional independencies among the random variables.

The nodes of the DAG represent random variables, e.g. observable quantities, latent variables, hypotheses, and the edges describe the conditional dependencies. Every variable corresponding to a certain node depends on the variables that are represented by the respective parent nodes, i.e. there is a directed edge from each parent node to its child nodes. Conversely, nodes which are not connected through edges depict variables that are conditionally independent of each other. Every node is assigned with a probability density function $P(X|parents(X))$ which is called conditional probability table (CPT) and quantifies the influence of the parent variables on the variable X. If a node has no parent nodes, the probability density function corresponds to the a priori probability of the variable. All variables together represent the full joint probability density function of the variables and thus provide a model for the uncertain knowledge. The full joint probability density function represented by a Bayesian network is given by:

$$P(X_1 = x_1, \ldots, X_n = x_n) = \prod_{i=1}^{n} P(X_i = x_i | X_j = x_j \textit{ for each } X_j \textit{ being parent of } X_i) \tag{4.1}$$

Based on this joint probability density function it is possible to do inference on the data represented by the Bayesian network, i.e. to calculate the conditional probability density function for a set of query variables given certain observations which result in an allocation of values to a set of evidence variables. Here, the simplest method is inference by enumeration, which is explained using the Bayesian network shown in Figure 4.1. Variable elimination

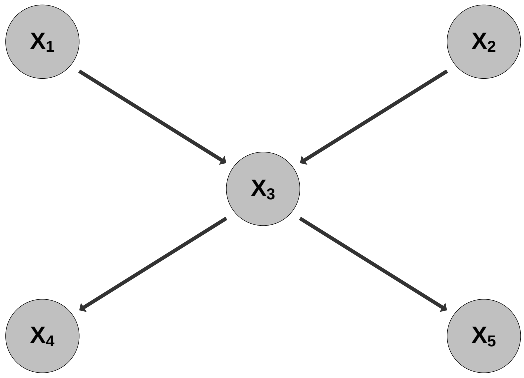

Figure 4.1: Simple Example for a Bayesian Network

is based on the same ideas and can be considered as a more efficient implementation of inference by enumeration.

Consider for example a query on the variable X_1 while X_4 and X_5 have been observed. This means, we would like to calculate the conditional probability $P(X_1 = x_1 | X_4 = x_4, X_5 = x_5)$. The starting point forms the definition of conditional probability which yields to:

$$P(X_1 = x_1 | X_4 = x_4, X_5 = x_5) = \frac{P(X_1 = x_1, X_4 = x_4, X_5 = x_5)}{P(X_4 = x_4, X_5 = x_5)} \tag{4.2}$$

As the denominator on the right side is independent of X_1, i.e. constant for all values of X_1, and it has to be ensured that the conditional probabilities of variable X_1 given the observed values for X_4 and X_5 sum up to 1, the denominator can be subsumed by a normalization factor α. Considering also the marginalization over the hidden variables (non-evidence variables and non-query variables) this results in:

$$P(X_1 = x_1 | X_4 = x_4, X_5 = x_5) = \alpha \sum_{x_2} \sum_{x_3} P(X_1 = x_1, X_2 = x_2, X_3 = x_3, X_4 = x_4, X_5 = x_5) \tag{4.3}$$

Exploiting Equation 4.1 and rewriting the equation using the distributive law yields to:

$$\begin{aligned} P(X_1 = x_1 | X_4 = x_4, X_5 = x_5) &= \alpha P(X_1 = x_1) \sum_{x_2} P(X_2 = x_2) \cdot \\ &\sum_{x_3} P(X_3 = x_3 | X_1 = x_1, X_2 = x_2) P(X_4 = x_4 | X_3 = x_3) P(X_5 = x_5 | X_3 = x_3) \end{aligned} \tag{4.4}$$

Another approach for inference in Bayesian networks is Pearl's belief propagation algorithm [109, 88]. In its original version [108], it allows to calculate the marginal distribution of unobserved nodes given the observed nodes in Bayesian networks that are trees. Later, this algorithm was extended for polytrees [74] and also provides useful approximations for general DAGs, as shown in [109].

The basic idea of the algorithm is to base inference mainly on local computations performed for the nodes of the network and then to propagate belief updates via messages to the corresponding child nodes (π-messages) and to the parent nodes (λ-messages). Here, the observation that a node X separates the network into two disjoint parts is crucial. *Causal* evidence that is accessible through the parents of X (denoted as E_{X^+}) is forwarded 'downwards' in π-messages and *diagnostic* evidence coming from the child nodes of X (denoted as E_{X^-}) is propagated 'upwards' in the λ-messages. Exploiting the independence assumption of the evidences, the marginal distribution for node X can now be calculated as:[1]

$$P(X|E_{X^+}, E_{X^-}) = \alpha P(X|E_{X^+})P(E_{X^-}|X) = \alpha \pi(X)\lambda(X) \tag{4.5}$$

where $\pi(X) = P(X|E_{X^+})$ and $\lambda(X) = P(E_{X^-}|X)$. According to [88], $\pi(X)$ calculates as:

$$\pi_X(x) = \sum_{u_1,\dots,u_p} P(X = x|u_1,\dots,u_p) \prod_{j=1}^{p} \pi_{U_jX}(u_j) \tag{4.6}$$

where $U_1,\dots,U_p$ denote the parents of X and π_{U_jX} are the π-messages sent from node U_j to X. For roots it holds $\pi_X(x) = P(X = x)$, which is just the prior probability independent of the evidence. $\lambda(X)$ can be calculated as (see [88]):

$$\lambda_X(x) = \prod_{j=1}^{c} \lambda_{Y_jX}(x) \tag{4.7}$$

where $Y_1,\dots,Y_c$ are the children of X and λ_{Y_jX} denotes the λ-message from node Y_j to node X. Now the missing part for completion of the algorithm is the computation of the messages passed between the nodes. For the messages we have:

$$\pi_{XY_j}(x) = \alpha \pi_X(x) \prod_{k \neq j} \lambda_{Y_kX}(x) \tag{4.8}$$

$$\lambda_{XU_i}(u_i) = \beta \sum_x \lambda_X(x) \sum_{u_1,\dots u_{i-1},u_{i+1},\dots,u_p} P(X = x|u_1,\dots,u_p) \prod_{k \neq i} \pi_{U_kX}(u_k) \tag{4.9}$$

In an initialization step, $\pi_X(x)$ is assigned with the prior of X, i.e. $P(X = x)$, if X has no parent node, and $\lambda_X(x) = 1$ for all values x if node X has no child nodes. If evidence is available for X, then $\pi_X(x)$ and $\lambda_X(x)$ are initialized with:

$$\begin{aligned} \pi_X(x_i) &= 1 \quad \text{if } x_i \text{ has been observed for } X, \text{ 0 otherwise} \\ \lambda_X(x_i) &= 1 \quad \text{if } x_i \text{ has been observed for } X, \text{ 0 otherwise} \end{aligned} \tag{4.10}$$

[1] A detailed derivation of the following equations is out of the scope of this dissertation, and the reader is referred to [88].

After this initialization step, the following four steps are repeated until no change occurs:

1. $\forall X$, if X has received all π-messages from its parent nodes, $\pi_X(x)$ is calculated.
2. $\forall X$, if X has received all λ-messages from its child nodes, $\lambda_X(x)$ is calculated.
3. $\forall X$, if $\pi_X(x)$ has been calculated and X has received the λ-messages from all children except Y, then $\pi_{XY}(x)$ is calculated and sent to Y.
4. $\forall X$, if $\lambda_X(x)$ has been calculated and X has received the π-messages from all parents except U, then $\lambda_{XU}(x)$ is calculated and sent to U.

Finally, $P(X|E_{X^+}, E_{X^-})$ is calculated by multiplying $\lambda(X)$ and $\pi(X)$ and is normalized with the factor α.

In this dissertation, the belief propagation algorithm described above will serve as baseline for performing inference in polytree Bayesian networks, with Naïve Bayes Classifiers and Hidden Markov Models. Later on, we also consider the propagation of belief using the Transferable Belief Model described in Section 4.2 in order to allow the incorporation of Dempster-Shafer belief functions into these reasoning schemes.

4.1.2 Naïve Bayes Classifiers

A Naïve Bayes Classifier is a simple probabilistic classifier assuming that the features used for discriminating between the classes are independent of each other. This means, the presence or absence of a particular feature of a class is not correlated to the presence or absence of the other features, and thus all features contribute independently to an object's probability for belonging to a class. As apparently over-simplified assumptions are used, the classifier has to be considered as 'naïve' to some extent. Still, Naïve Bayes Classifiers have proved to work quite well for a number of real-world problems [167].

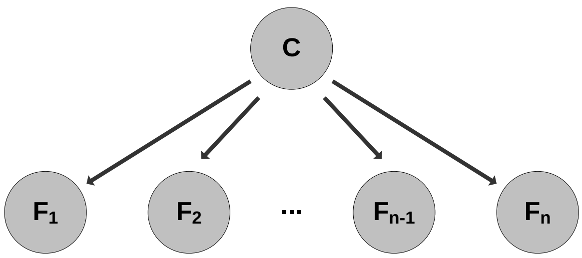

Figure 4.2: Naïve Bayes Classifier as Bayesian Network

Figure 4.2 shows a Naïve Bayes Classifier represented as a Bayesian network. A classification of a feature set consists of calculating the conditional probability $P(C|F_1, \ldots, F_n)$ and assigning the class c to the feature set that maximizes this conditional probability. This decision rule corresponds to a maximum a posteriori classification. Viewing the Naïve Bayes Classifier from the perspective of the belief propagation algorithm described in the previous section, the conditional probability can be calculated as:

$$P(C = c|F_1 = f_1, \ldots, F_n = f_n) = \alpha P(C = c) \prod_{i=1}^{n} P(F_i = f_i|C = c) \tag{4.11}$$

where $\pi_C(c)$ is set to the prior $P(C = c)$ and $\lambda_C(c)$ is calculated as the product of the λ-messages from the feature nodes to the class node, which can be derived from Equation 4.9 simply as $P(F_i = f_i|C = c)$. Equation 4.9 contains also a normalization factor β for each message. However, all normalization factors are subsumed by the single factor α in Equation 4.11.

It is also noteworthy here that training of Naïve Bayes Classifiers, which often consists of determining the means and the variances of the conditional probability distributions $P(F_i|C)$, can be performed with quite a small amount of training data. As independence between the different feature variables is assumed, it is sufficient to estimate the mean and variance for each class. An estimation of the entire covariance matrix is not required.

4.1.3 Hidden Markov Models

Markov models or Markov chains are used to model discrete random processes showing the Markov property. Here, a discrete random process refers to a system which can be in a number of different states and changes its state according to a probability model (transition probabilities) in discrete steps. Due to the Markov property the probability of future states only depends on the current state and is independent of the states of the system in previous steps.

In contrast to a regular Markov model, where the state of the system can directly be observed, in a Hidden Markov Model (HMM) the state is not directly visible. However, the output of the system depends on the state of the system and each state has a probability density function for the possible output symbols. Thus, a sequence of observations generated by the HMM provides some information about the sequence of states.

HMMs are characterized by three parameters: the initial state probabilities at step 1, the transition probabilities describing the conditional probabilities of reaching a state given the previous state and the observation probabilities representing the conditional probabilities of the output symbols given the current state. HMMs can also be considered as very simple dynamic Bayesian networks, i.e. a Bayesian network that models sequences, often time-series of random variables with interdependencies between the different steps. Figure 4.3 shows the representation of a HMM as dynamic Bayesian network.

HMMs have been used for a number of temporal pattern recognition tasks, such as speech recognition [113], activity recognition [123, 168, 98, 114] or human motion analysis [114]. Here, the inference tasks to be performed include calculation of the probability a HMM generates an observation sequence, determining the probability of a system of being in a particular state at a certain step, or finding the state sequence with highest probability.

As a HMM is a simple (dynamic) Bayesian network, the probability density function for the state of a system at step $t = k$ given an observation (evidence) sequence $E_1 = e_1, \ldots, E_n = e_n$, i.e. $P(X_k|E_1 = e_1, \ldots E_n = e_n)$, can be calculated with the belief propagation algorithm introduced in Section 4.1.1. The corresponding instance of the algorithm used for HMMs is called *Forward-Backward-Algorithm*.

For classification of observation sequences, usually a HMM is created for each of the classes under consideration and for each of these models the probability to generate the observation sequence is computed. The sequence is then assigned to the class that corresponds to the

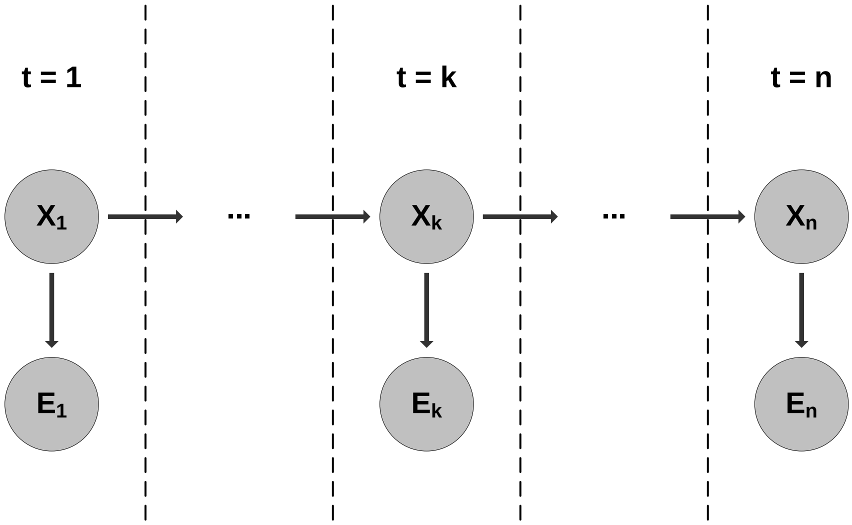

Figure 4.3: Hiddon Markov Model as Dynamic Bayesian Network

model with the highest generation probability. According to [113], $P(E_1 = e_1, \ldots, E_n = e_n|\lambda)$ or $P(E|\lambda)$, i.e. the probability of the observation sequence given the HMM λ calculates as:

$$
\begin{aligned}
P(E|\lambda) &= \sum_q P(E|Q = q, \lambda) \cdot P(Q = q|\lambda) \\
&= \sum_{q_1,\ldots,q_n} P(X_1 = q_1)P(E_1 = e_1|X_1 = q_1)P(X_2 = q_2|X_1 = q_1)P(E_2 = e_2|X_2 = q_2)\ldots \\
&\qquad \cdot P(X_n = q_n|X_{n-1} = q_{n-1})P(E_n = e_n|X_n = q_n) \qquad (4.12)
\end{aligned}
$$

Rewriting this equation using the distributive law, an inductive solution for the calculation of $P(E|\lambda)$ can easily be derived:

- Initialization:

  $\alpha_1(i) = P(X_1 = i)P(E_1 = e_1|X_1 = i) \quad 1 \leq i \leq m$, where m the number of possible states
- Induction:

  $\alpha_{t+1}(j) = P(E_{t+1} = e_{t+1}|X_{t+1} = j) \sum_{i=1}^{m} \alpha_t(i)P(X_{t+1} = j|X_t = i)$

  where $1 \leq t < n$ and $1 \leq j \leq m$.
- Termination:

  $P(E|\lambda) = \sum_{i=1}^{m} \alpha_n(i)$

Within this inductive solution, $\alpha_t(i)$ corresponds to the probability of observing $e_1, \ldots, e_t$ and being in state $X_t = i$ at step t, i.e., $\alpha_t(i) = P(E_1 = e_1, \ldots, E_t = e_t, X_t = i|\lambda)$. Here, it holds:

$$P(X_t = i|E_1 = e_1, \ldots, E_t = e_t, \lambda) = \alpha P(E_1 = e_1, \ldots, E_t = e_t, X_t = i|\lambda) \qquad (4.13)$$

This means the α_t can also be calculated successively by applying the belief propagation algorithm and just ignoring all normalization factors. Indeed, by inspecting Equation 4.5 to Equation 4.9 this direct correspondence becomes obvious. In [113], it is also shown that $P(E|\lambda)$ can be written as:

$$P(E|\lambda) = \prod_{t=1}^{n} \frac{1}{c_t} \quad \text{with} \quad c_t = \frac{1}{\sum_{i=1}^{m} \alpha_t(i)} \tag{4.14}$$

if $\alpha_1(i)$ to $\alpha_{t-1}(i)$ have already been scaled with c_1 to c_{t-1} to sum up to 1. Thus, we can simply calculate the probability by applying the belief propagation algorithm and just multiplying the corresponding normalization factors.

Determining the state sequence with the highest probability is very similar to the inductive solution described above. However, the summation is replaced by a maximization step [122]:

$$\begin{aligned} &\max_{x_1,\ldots,x_t} P(x_1,\ldots x_t, X_{t+1}|e_1,\ldots,e_{t+1}) = \\ &\quad \alpha P(e_{t+1}|X_{t+1}) \max_{x_t}\left(P(X_{t+1}|x_t) \max_{x_1,\ldots,x_{t-1}} P(x_1,\ldots,x_{t-1},x_t|e_1,\ldots,e_t)\right) \end{aligned} \tag{4.15}$$

For retrieving the actual state sequence with the highest probability and not only to calculate its probability, a pointer from a state back to the best previous state has to be stored. This approach for determining the state sequence with the highest probability is also known as the *Viterbi algorithm* [113].

4.2 The Transferable Belief Model

The Transferable Belief Model (TBM) [130, 133] is an elaboration of DST, which is designed as a normative model and based on Dempster-Shafer belief functions, but is built without ever introducing explicitly or implicitly any concept of probability [130, 133]. One of its most important contributions with respect to this are justifictions for Dempster's Rule of Combination and Dempster's Rule of Conditioning without relying on probabilities or a similar concept. As already mentioned above, in [131] also the difference between an open world assumption and the closed world assumption is discussed, and it is shown that an open world assumption leads to the 'unnormalized' versions of these two rules.

The core concepts of the TBM also include the **Disjunctive Rule of Combination** (DRC) and the **Generalized Bayesian Theorem** (GBT) [132], which are important ingredients for establishing reasoning approaches based on DST. Whereas the Conjunctive Rule of Combination allows the combination of belief functions corresponding to two distinct pieces of evidence E_1 and E_2 that hold at the same time, the Disjunctive Rule of Combination enables combination of two belief functions if it is only known that the first or the second piece of evidence holds, i.e. the disjunction $E_1 \cup E_2$. The Disjunctive Rule of Combination is given as:

$$(m_1 \circledcup m_2)(C) = \sum_{A \cup B = C} m_1(A) \cdot m_2(B) \tag{4.16}$$

Furthermore, the Disjunctive Rule of Combination shows the following property:

$$pl_\Omega(x|\theta) = 1 - \prod_{\theta_i \in \theta} 1 - pl_\Omega(x|\theta_i) \quad \forall \theta \subseteq \Theta, \forall x \subseteq \Omega \tag{4.17}$$

where Ω and Θ are two frames of discernment.

As it is obvious from its name, the Generalized Bayesian Theorem constitutes a generalization of the well-known Bayesian theorem for Dempster-Shafer belief functions. It allows to propagate beliefs in a Directed Evidential Network [163] from a node back to its parent node. The Disjunctive Rule of Combination and the Generalized Bayesian Theorem are connected through the relation:[2]

$$pl_\Omega(x|\theta) = pl_\Theta(\theta|x) \quad \forall \theta \subseteq \Theta, \forall x \subseteq \Omega \tag{4.18}$$

Thus, with Equation 4.17 the Generalized Bayesian Theorem is given as:

$$pl_\Theta(\theta|x) = 1 - \prod_{\theta_i \in \theta} 1 - pl_\Omega(x|\theta_i) \quad \forall \theta \subseteq \Theta, \forall x \subseteq \Omega \tag{4.19}$$

In Section 10.1 we show how the Generalized Bayesian Theorem can be exploited to allow DST-based reasoning in the probabilistic schemes described above with the help of a message passing approach according to the belief propagation algorithm.

[2] It is important to note here that this relation only holds if we allow unnormalized belief functions and mass assignments, i.e. the masses not necessarily have to sum up to 1. In particular, we use unnormalized mass assignments if we propagate belief from a node back to its parent node.

5 Ontology-based Specification of Application Domains

The work presented in this thesis targets distributed applications in dynamic heterogeneous environments that need to process the contents of the exchanged information. This has to be seen in contrast to applications which are designed only to store the content of exchanged information or to present it to humans. Whereas in the latter case communication middleware frameworks or standard communication protocols would suffice, in our work a common and formally represented vocabulary on the application domain is required, allowing the involved software components to semantically interpret the exchanged information.

A formal representation of an application domain is usually based on a *conceptualization*, which consists of the objects, concepts, and other entities that are presumed to exist in the area of interest and the relationships that hold between them [49]. Here, a conceptualization is considered to be an abstract, simplified view of the world that we wish to represent for some purpose. In our work, every application, or more precisely, every exchanged message has to be related to a conceptualization.

An explicit specification of such a conceptualization is called an ontology [53]. Here, the term ontology is borrowed from philosophy, where it represents a systematic account of existence. For distributed applications as we target, what *exists* exactly corresponds to what can be represented. If an application domain is represented using a declarative formalism, the set of entities that can be represented is called the *Universe of Discourse* and comprises classes, relations, functions, or other objects.

5.1 Ontology Languages

For the construction of ontologies, formal specification languages are required allowing to encode the knowledge about a specific application domain. As an ontology can be considered as a special case of a knowledge representation, in principle all kinds of formal knowledge representation languages can be used to specify ontologies. Guided by high-level knowledge representations in artificial intelligence which are commonly based on logics, ontology languages are usually declarative languages, which in most cases are generalizations of frame languages, and are commonly based on either first-order logic or on description logic. This allows to utilize the reasoning mechanisms of the underlying logic to check consistency and also to infer new knowledge that can be deduced from the defined objects and their relationships.

In the past a number of ontology languages has been developed. Here, the authors of [26] distinguish between *traditional* ontology languages, like Ontolingua [53], LOOM [79], OCML [86] and FLogic [72, 73], and *Web-based* ontology languages as e.g. RDF/RDF-S

[81, 14], DAML+OIL [62], WSML [138], and OWL [155]. Whereas the traditional ontology languages have mainly been established to achieve interoperability of knowledge-based systems in artificial intelligence, the Web-based ontology languages are designed to facilitate the interchange of ontologies across the World Wide Web (WWW) and the cooperation among heterogeneous agents placed on it. Web-based ontology languages build on Web standards such as XML and RDF and aim at representing the knowledge contained in an ontology in a simple and human-readable way.

In recent years, OWL has been established as de facto standard for the specification of ontologies for a number of applications. Therefore, we also have decided to use OWL for the specification of application domains. However, the general concepts presented in this thesis are not particularly tailored to OWL and can be specified in other ontology languages with a corresponding expressiveness as well.

5.2 Web Ontology Language (OWL)

The Web Ontology Language (OWL) [155] is an ontology language, or more precisely a family of ontology languages, endorsed by the *World Wide Web Consortium (W3C)* [150]. OWL is based on RDF and a revision of the DAML+OIL Web ontology language incorporating lessons learned from the design and application of DAML+OIL. From a formal point of view, OWL semantics are based on description logics, which are a family of logics that are decidable fragments of first-order logic. In this respect, OWL provides a syntax for describing and exchanging ontologies and at the same time has a formally defined semantic corresponding to the $\mathcal{SHOIN}(D)$ description logic. However, these semantics are enhanced by additional concepts, in order to provide compatibility with RDF-S.

Whereas earlier ontology languages have been used to develop tools and ontologies for specific user communities, they were not designed to meet the requirements of an ontology language for the WWW in general. In order to provide a further step to ontologies generally applicable to the WWW, OWL is designed to explicitly facilitate:

- Ability to be distributed across many systems
- Scalability appropriate to WWW needs
- Compatibility with established Web standards for accessibility and internationalization
- Openness and extensibility

The *OWL Use Cases and Requirements Document* [156] of the W3C provides more details on ontologies, presents six use cases in order to further motivate the need for a Web ontology language, and formulates design goals and requirements. One of the six use cases is about *Ubiquitous Computing* and highlights the challenge to achieve interoperability of a priori unknown, dynamically appearing and disappearing devices in an ad-hoc manner. Devices should be enabled to discover and understand other devices on a semantic basis. This reflects quite well part of the challenges subject of our research.

In OWL, ontologies are defined in terms of classes, properties, instances and operations. User-defined **classes** are subclasses of the root class owl:Thing. A class may contain individuals, which are instances of the class, as well as other subclasses. **Properties** are binary relations specifying the characteristics of the classes. They mainly correspond to attributes of class instances and sometimes represent data values or links to other instances. Two types of simple properties are distinguished: datatype and object properties. Datatype properties reflect relations between the instances of classes and RDF literals or XML schema datatypes. Object properties are relations between the instances of two classes. **Instances** are individuals that belong to defined classes, where a class may have any number of instances. Instances are used to define the relationship among different classes through the corresponding properties. **Operations** in OWL express union, intersection and complement of classes and facilitate class enumeration, cardinality and disjointness.

Furthermore, OWL provides a number of language constructs to express **equality/inequality** of classes, properties and instances, to define **property characteristics** (symmetric properties, transitive properties, inverse properties, etc.) and to specify **property restrictions** describing how properties can be used by instances of a class. Examples of these restrictions are *all values from class*, *some values from class*, *maximum cardinality*, *minimum cardinality*.

It is noteworthy here that OWL adopts an *Open World Assumption* in contrast to Prolog [13], for example, which uses a *Closed World Assumption*. This means that if a statement cannot be proved to be true using current knowledge, in OWL the conclusion that the statement is false cannot be drawn.

The OWL ontology language family provides three increasingly expressive (sub-)languages designed for use by specific communities of developers and users. These sub-languages differ with regard to what can be legally expressed and with regard to what can be validly concluded.

- **OWL Lite** was originally designed to support users which primarily need to specify classification hierarchies and simple constraints. While it supports cardinality constraints, it only permits cardinality values of 0 or 1. The rationale behind was the hope that it would be simpler to provide tools for OWL Lite in comparison to the more expressive successors, allowing a quick migration path for systems utilizing thesauri and other taxonomies. In practice, however, most of the constraints available in the successor OWL DL can be expressed using complex combinations of OWL Lite features, and thus the expressiveness constraints imposed on OWL Lite result only in little more than syntactic inconveniences.
- **OWL DL** is designed to provide the maximum expressiveness possible while retaining computational completeness, decidability, and the availability of practical reasoning algorithms. In general, OWL DL already includes all OWL language constructs, but they can only be used under certain restrictions. OWL DL was named due to its correspondence with description logic, a field of research that has studied the logics that form the formal foundation of OWL. As already stated above, OWL DL corresponds to the $\mathcal{SHOIN}(D)$ description logic.
- **OWL Full** is designed to retain some compatibility with RDF-S. In this respect, OWL Full is meant for users who want maximum expressiveness and the syntactic freedom of RDF with no computational guarantees. Consequently, OWL Full is based on a different semantics in comparison to OWL Lite or OWL DL. For example, OWL Full permits that a class can be treated as a collection of individuals and at the same time as an individual in its own right, which is not allowed in OWL DL. OWL Full allows an augmentation of the pre-defined OWL vocabulary. However, it is unlikely that reasoning engines will be able to support the complete reasoning for OWL Full.

At the *'First OWL Experiences and Directions Workshop'* held in Ireland in November 2005, several restrictions of OWL were identified which have caused problems in many applications and have been rendered unnecessary considering the advances in logic-based knowledge representations: syntactical inconveniences, limited expressiveness and no guarantees with regard to reasoning time. Consequently, an extension of OWL-DL, called OWL 1.1, was proposed as result of the workshop and became a *W3C Member Submission* [152]. Since October 2007, a new W3C working group has been concerned with extending OWL by several new features as proposed in the OWL 1.1 Member Submission. This new version, called OWL 2 [153], was announced on 27 October 2009.

OWL 2 enhances OWL-DL with several new features including syntactic enhancements to simplify specifications of some commonly used statements and increased expressive power for properties corresponding to the $\mathcal{SROIQ}(D)$ logic. Besides, extended support for datatypes, simple meta-modelling capabilities, extended annotation capabilities and other minor features have been added. Similar to OWL, OWL 2 also defines language subsets, called *profiles*, which are intended to meet certain performance requirements or to reduce implementation effort. In profiles some expressive power is traded for the efficiency of reasoning. In [154], three OWL 2 profiles are described:

- **OWL 2 EL** is intended to be useful for applications employing ontologies with a very large number of properties and classes. This profile is designed to capture the expressive power used by a number of such ontologies and comprises a subset of OWL 2 features for which reasoning can be performed in polynomial time with respect to the size of the ontology. Corresponding reasoning algorithms for this profile are available, which can be implemented in a highly scalable way. *EL* refers to the basis of the profile in the EL family of description logics, which only provide existential quantification.
- **OWL 2 QL** was defined having applications in mind which use large volumes of instance data and where query answering is the predominant reasoning task. In this profile, conjunctive query answering can be realized with conventional relational database systems. Actually, the QL acronym reflects the fact that in this profile query answering can be realized by rewriting the queries in a standard relational query language. If a suitable algorithm is employed, reasoning can be performed in LOGSPACE with respect to the size of the data. Furthermore, polynomial time algorithms are available for ontology consistency checking and class expression subsumption reasoning. Although the expressive power of the profile is quite limited, it covers the main features of conceptual models like UML class and ER diagrams.
- **OWL 2 RL** is intended for usage in applications which require scalable reasoning and expressive power at the same time, but can trade the full expressiveness of the OWL 2 for efficiency. Besides, RDF-S applications requiring some additional expressiveness are supported as well. Reasoning can be implemented on top of rule-based reasoning engines using a standard Rule Language, which is reflected by the RL acronym. Ontology consistency checking, instance checking, class expression satisfiability, class expression subsumption and conjunctive query answering reasoning can be realized in polynomial time with respect to the size of the ontology.

It is noteworthy here that there are many other possibilities to define profiles of OWL 2. Due to backward compatibility of OWL 2 with respect to OWL, all OWL Lite and OWL DL ontologies are OWL 2 ontologies. Therefore, OWL Lite and OWL DL can be viewed as profiles of OWL 2 as well.

The information model presented in this dissertation (see Chapter 7) was originally specified using OWL DL as it guarantees computational completeness and decidability. Later on, we have additionally used *Property Chain Inclusions* in order to simplify the incorporation of already available ontologies and to fit them to our scheme (see Section 7.2.4). Thus, from a formal perspective we use OWL 2 in a profile corresponding to OWL DL with the extension of property chain inclusions.

6 Related Work

The main objective of this work is to provide a comprehensive approach for information exchange and fusion in heterogeneous distributed environments. This means, we simultaneously target a number of different challenges ranging from semantic discovery of information providers and consumers to fusion of heterogeneous sensor data. Consequently, we touch several different research areas, where valuable work has already been performed. It is not possible to discuss related work in all these areas as part of this thesis. Instead, we focus on *Context Management and Reasoning in Ubiquitous Computing* and *Cooperative Teams of Heterogeneous Mobile Robots* where we see our main contributions to the state of the art.

For a comprehensible discussion of the related works, we first present the criteria that serve as a baseline for our evaluation. These criteria correspond to a list of requirements that can directly be derived from the challenges presented in Section 1.2.

1. **Dynamic integration of a priori unknown information consumers and providers**.

   An approach for information exchange and fusion has to deal with the dynamic nature of the computing environment and to allow integration of independently developed and a priori unknown components, modules or services into a single system.

2. **Establishment of a common vocabulary**.

   Independent development of information consumers and providers implies a kind of common vocabulary established through an onotology or a similar mechanism which allows to resolve naming conflicts.

3. **Mechanisms for the semantic interpretation of data structures**.

   Different development teams may utilize those representations for the data which are most suitable for their purposes. Thus, the representation of the information including the meta-data to describe the imperfect nature of sensor data has to be semantically interpretable.

4. **Support for conversion between data representations**.

   In order to bridge heterogeneity with respect to data structures, elaborate mechanisms are required to convert between different representations of a piece of information.

5. **Expression of information offers and needs**.

   For semantic discovery and matching of information providers and consumers, a kind of language is required that facilitates detailed specification of offered or needed information.

6. **Support of competitive sensor fusion**.

   Due to the imprecise and unreliable nature of sensor information, conflicts in the provided information have to be expected if more than one sensor is available for the same piece of information. Competitive sensor fusion schemes are required in order to resolve the conflicts and to improve the quality of the provided information.

7. **Provisioning of reasoning schemes that consider the imperfect nature of sensor data**.

   In many cases, information is required that cannot directly be sensed by sensors. Instead, reasoning mechanisms are required to aggregate low-level sensor data to high-level information.

In the list presented above, we have focused on the functional requirements that can directly be derived from our envisaged scenarios. At this point, it is important to note that these requirements have not been identified with a particular application in mind as we aim for a generic approach which is applicable for a wide range of problems. Particular applications might have additional requirements. Besides, we are aware that non-functional requirements like *extensibility*, *scalability* or *ease of development* are also of high importance, as for software systems in general. Still, we will base our review of related works mainly on the functional requirements as these form a kind of work plan for the work presented in this thesis.

6.1 Context Management and Reasoning

6.1.1 The ASC Model

Strang and Linnhoff-Popien evaluate in [134] the most relevant context modelling approaches based on the data structures used for representing and exchanging context information: key-value, markup scheme, graphical, object-oriented, logic-based and ontology-based models. According to their evaluation the most promising assets for context modelling for ubiquitous computing environments are found in ontology-based models. In these models, the semantic context information is represented using an ontology markup language, for example OWL (Web Ontology Language) [155]. We share their opinion and consider ontologies as an appropriate way to deal with the heterogeneity implied by ubiquitous computing environments. A corresponding ontology defines a common vocabulary which makes it possible to reason about various context types, thanks to machine-interpretable definitions on basic concepts in the domain and relations among them.

There are several projects that also apply ontologies as a central concept for modelling context information. For instance, Chen et al. [23, 22] defined a context ontology based on OWL to support ubiquitous agents in their *Context Broker Architecture* (CoBrA). Their approach applies sensor information detection and context awareness in a home area intelligent environment as a way of taking decisions, dealing with users' activities, intentions and movements between different home areas.

The *Service-Oriented Context-Aware Middleware* (SOCAM) [145, 160] is an architecture for building context-aware services based on a two-level context model. Their context ontology is divided into a two-level hierarchy, distinguishing between common and specific context information. The upper level describes global concepts of the ontology and captures general knowledge. The lower level is divided into several pervasive computing sub-domains, each one of which defines specific details and properties for each scenario. Depending on the situation and the available devices, an appropriate sub-domain is selected from the lower level and the corresponding ontology is dynamically plugged into the upper ontology.

In [135], Strang et al. describe a context modelling approach which uses ontologies as a formal foundation and is based on the *Aspect Scale Context* (ASC) model. The *Context Ontology Language* (CoOL) is derived from the model, which is used to enable context awareness and contextual interoperability during service discovery and execution in a distributed architecture. *Aspects* are considered to be dimensions of the situation space and to be sets of one or more scales. Likewise, *scales* aggregate one or more elements of *context information* and correspond to what we call representations. An example for an aspect is *GeographicCoordinateAspect* and two corresponding scales could be *WGS84Scale* or *GaussKruegerScale*. In this respect, the work of Strang et al. stands out from all other analyzed approaches, as it also captures different representations for a type of context information. Besides, transformations between different representations are covered by *IntraOperations* and *InterOperations*. *IntraOperations* just convert between two scales of one aspect, whereas *InterOperations* also involve information corresponding to other aspects.

Major contributions: The ASC model and CoOL provide a context modelling approach based on ontologies and explicitly address heterogeneity with regard to different representations of context information. Transformations between different scales of an aspect are also envisaged. Similar to our work, they propose no concrete ontology but a language or meta-model that can be utilized to construct appropriate ontologies. In this respect, the ASC model and CoOL serve as a sound basis for our work.

Weaknesses: The ASC model prescribes meta-data attributes for context information like *timestamp*, *minError*, *meanError* and other quality attributes, but no hint is given how these attributes are handled when performing conversions between different scales. In general, no concrete description is provided how conversions are performed in a corresponding framework. Besides, with the ASC model it is not possible to express uncertainty among several hypotheses of a concrete context information, e.g. the location of a user. Thus, it is only of limited use to cover the real characteristics of sensor data.

6.1.2 Context Modeling Language and PACE

Henricksen and Indulska [56, 57, 59] introduced the *Context Modeling Language* (CML) in order to support designers with the task of exploring and specifying the context requirements of context-aware applications. CML is a graphical modelling approach utilizing *Object-Role Modelling* (ORM) as a formal foundation and provides concepts to define types of context (i.e. fact types) and a classification of these context types (static, sensed, derived, or profiled). In addition, CML allows to associate meta-data attributes and to define dependencies between different context types. With the situation abstraction, support for reasoning on context information is provided. The proposed context modelling approach can also deal with special characteristics of context information, such as its temporal nature, incompleteness, ambiguity, etc., and it even incorporates ontologies to address particular aspects like privacy.

A big advantage of enhancing ORM is the availability of a mapping to a relational model, which in turn enables a straight-forward representation in relational databases. Thus, it is obvious to realize the context management system on top of a corresponding database system, which opens up to use SQL [64] as a flexible way to access context information.

The authors of [57] not only proposed the CML but also the *Pervasive Autonomic Context-aware Environments* (PACE) middleware [58], which is intended to help managing the complexity arising from a heterogeneous distributed environment. The middleware system allows to seamlessly bind together the different involved sensors, actuators, application components and context processing components. It provides aggregation and storage of context information, performs query evaluation and contains a distributed set of context repositories, each of which manages a collection of context models. To facilitate interaction between application components and the context management system, a programming toolkit is defined and implemented in Java using *Remote Method Invocation* (RMI) interfaces for communication.

The approach is complemented by a model-driven development approach and an associated set of tools assisting with the generation of components and with the development and deployment of context-aware systems, starting from context models specified in the context management system.

Major contributions: The main objective of the CML approach is ease of development. In this respect, the CML was a pioneering approach. Henricksen and Indulska have also identified the need for an easy but flexible way to access context information and integrated support in the PACE middleware similar to what we understand of querying context information. Furthermore, a lot of different aspects have been investigated as part of the CML/PACE approach, as e.g. the imperfect nature of context information, context reasoning and privacy aspects.

Weaknesses: Although the PACE middleware supports a distributed environment, distribution is mainly considered from a static perspective. The dynamic nature of a ubiquitous computing environment with appearing and disappearing information providers and consumers is only addressed to a limited extent. The same is true for the independent development of the involved services and heterogeneous data representations. Although the imperfect nature of context information was taken into account in their work, no real solution was provided for considering the impreciseness of low-level context data in reasoning approaches applied to derive high-level context information.

6.1.3 Context Toolkit

Today still widely referenced in the literature and at the time of its introduction a pioneering approach is the *Context Toolkit* [29, 30]. Its main objective is to simplify the development of context-aware applications by allowing the reuse of specialized components, which include widgets, aggregators, interpreters, services and discoverers. Widgets are components mainly responsible for gathering context information directly from the sensors. Aggregators are a kind of meta-widgets showing all characteristics of a widget but additionally providing the ability to aggregate context information of real-world entities. They can also act as gateways between applications and widgets. Interpreters derive high-level information from low-level data, services are utilized by applications to invoke actions and actuators, and discoverers are used to locate the available widgets, aggregators, interpreters and services. As the Context Toolkit utilizes the HTTP protocol for communication and an XML-based content language, a wide range of components can be incorporated via the use of available Web standards.

Several extensions have been developed for the Context Toolkit. For instance, Newberger and Dey [93] proposed an extension for providing user control of context-aware systems. Wu et al. [159] have recognized the benefits of Dempster-Shafer Theory of Evidence (DST) to fuse context information from different sources and developed a corresponding extension for the Context Toolkit. As part of their work, they evaluated a number of variants of Dempster's Rule of Combination and showed the viability of their approach in a scenario where the focus of attention of meeting participants has to be detected.

Major contributions: The Context Toolkit facilitates the discovery of widgets, aggregators, interpreters and services and enables easy integration via the HTTP protocol and Web standards. An extension of the Context Toolkit incorporates Dempster-Shafer Theory as a means to fuse sensor information and to resolve conflicts among low-level context information.

Weaknesses: Although facilitating the discovery of components, heterogeneity issues like different representations of context information due to independent development are not considered. The Dempster-Shafer Theory is used to combine evidences from different sources and to resolve conflict but only for individual fusion processes. Incorporating Dempster-Shafer Theory in more elaborate reasoning approaches as envisaged in this thesis has not been addressed so far.

6.1.4 Context Fusion Networks

The *Context Fusion Networks* (CFN) approach proposed by Chen et al. [20, 21] explicitly targets the challenges arising from ubiquitous computing environments. Chen argues in [21] that it is not acceptable that individual applications are required to maintain connections to sensors and to process the raw data from scratch. This would increase the programmer's burden and the application would probably work poorly on a resource-constrained mobile device. Likewise, it is not feasible to deploy one common context service that could meet the needs of all possible applications. Thus, the CFN infrastructure model is proposed allowing context-aware applications to select distributed data sources and to compose them with customized data fusion operators into a directed acyclic information fusion graph. This graph describes how an application computes high-level understandings of the execution context from low-level sensory data.

The CFN infrastructure model has been realized in a prototype system called *SOLAR* [19]. This prototype was utilized to investigate and to propose new approaches for a number of issues like buffer overflow, packet loss, dependency management, etc., in an infrastructure for ubiquitous computing. SOLAR also provides a naming service supporting persistent queries and context-sensitive resource discovery.

Major contributions: Chen et al. [21, 19] identify a number of important challenges arising from the paradigm of ubiquitous computing and propose an infrastructure model along with a prototype system, which explicitly address these challenges. With its concept for context-sensitive resource discovery, the approach is well suited to handle environment dynamics.

Weaknesses: Although named Context Fusion Networks, the approach of Chen et al. is mainly concerned with sensor data fusion at the architectural level. Concrete approaches able to deal with the imperfect nature of sensors are not in the focus of their work. Heterogeneity is also claimed to be addressed in their work, but this mainly comprises heterogeneity with respect to the used network and communication facilities. However, independent development and heterogeneity of data representations are not addressed at all.

6.1.5 ECORA

ECORA (*Extensible Context Oriented Reasoning Architecture*) [104] is a prototype framework for building context-aware applications which are designed with a focus on reasoning about context under uncertainty and addressing issues of heterogeneity, scalability, communication and usability. The framework provides an agent-oriented hybrid approach combining centralized reasoning services with context-aware, reasoning-capable mobile software agents. The ECORA framework adopts the *Context Spaces* model [102, 103] to describe the context and to apply reasoning over modelled information under uncertainty. The underlying concepts use insights from geometrical spaces and the state-space model [97] hypothesizing that geometrical metaphors such as states within spaces are useful to guide reasoning about context.

In [104], the authors argue that ontology or logic-based models provide a way to uniformly represent context, but the typically applied logic-based reasoning mechanisms are suitable only for dealing with precise context information. On the other hand, most probability-based approaches lack an underlying general-purpose context model. The context model is tailored either to the specific inference algorithm or to a particular application domain. We share their opinion and a major contribution of our work consists of providing a general-purpose information model on top of which probability and Dempster-Shafer Theory-based reasoning schemes can be applied.

Major contributions: With the Context Spaces model [102, 103] a general purpose context modelling approach was proposed that is intended to consider the imperfect nature of context information. A number of reasoning algorithms have been developed [101] that are able to perform complementary data fusion and to infer context situations. By defining situations containing states that can vary over time, even the temporal nature of context information is considered.

Weaknesses: The authors of [104] showed that the proposed model and the corresponding reasoning algorithms are suited for complementary data fusion. However, competitive data fusion required to resolve conflicts in the available context information is not addressed and can hardly be realized within the Context Spaces model without incorporating probability- or Dempster-Shafer-based approaches. Besides, independent development and heterogeneity issues are mostly neglected.

6.1.6 Gaia

Ranganathan et al. [115] developed a prototype pervasive computing infrastructure (Gaia) to support context awareness for automated agents, which can be applications, services and/or devices. The middleware aids the development of context-aware agents by supporting context sensing and reasoning on context information, relieving the developers from many implementation issues. A main objective of Gaia is also to support applications and services to reason about uncertain context. For this purpose, a number of well-established approaches have been utilized: probabilistic logic, fuzzy logic, and Bayesian networks.

From a formal perspective, any piece of information whose truth value is potentially uncertain is represented as a predicate. The structure of predicates and their semantics are specified in an ontology defining various context types as well as the arguments that the predicates must have. Each context type corresponds to a class in the ontology. The ontology is written in DAML+OIL [62].

Apart from *Context Consumers*, *Context Providers* and *Context Synthesizers*, Gaia also prescribes a *Context History Service* allowing to query for past context information and a *Context Provider Lookup Service* enabling *Context Providers* to advertise what kind of information they offer.

Major contributions: Gaia addresses many of the challenges we have identified for our work. A general-purpose context model is defined, which is based on a common vocabulary defined in an ontology. The imperfect nature of context data is considered in the context representation and also in reasoning schemes for deriving high-level context information. Furthermore, Gaia provides facilities to advertise and discover information offers and thus also deals with the dynamic nature of a ubiquitous computing environment.

Weaknesses: Similar to ECORA (see Section 6.1.5), Gaia provides good support for complementary data fusion. However, competitive data fusion is addressed only to a limited extent. Although a common vocabulary is defined via an ontology, independent development of context components involving heterogeneous data representations is mostly neglected.

6.1.7 Context Integration and Abstraction Approach of University College Dublin

The research group around Dobson from University College Dublin has recently published a series of papers on resolving uncertainty in context integration and abstraction [164, 83, 82]. In [164], a method is proposed that is able to consider the imperfect nature of context information and also its temporal nature: the older context information is the more unreliable it is. The approach involves concepts of traditional probability theory and shows similarities to Dempster-Shafer Theory but has to be considered rather an ad-hoc solution. However, the authors recognized that Dempster-Shafer Theory is a good candidate to avoid the shortcomings of their previous approach and that it builds a sound formal basis.

Consequently, in [82] McKeever et al. have revisited their previous work and presented a more formal approach based on Dempster-Shafer Theory, the viability of which is demonstrated by an activity recognition system. Concretely, they represent different layers of abstraction through a directed acyclic graph (DAG) and apply concepts like evidential mapping and compatibility relations [78]. In our view, the DAGs used in their approach correspond more or less to a hierarchy of Naïve Bayes Classifiers but involving DST instead of traditional Bayesian inference. In [82], it is argued that using DST for context integration and abstraction is also useful for scenarios where it is difficult to collect a huge amount of training data. Instead, domain knowledge can easily be integrated to compensate missing training data.

Major contributions: It is recognized that Dempster-Shafer Theory builds a sound formal basis for resolving uncertainty in context integration and abstraction. The approach also demonstrates how the temporal nature of context information can be handled. Furthermore, a number of alternatives for Dempster's Rule of Combination [126, 87] are evaluated which are useful for situations where much conflict is included in the evidences to be combined.

Weaknesses: The approach for situation inference based on DAGs seems to be quite limited, as it mainly corresponds to the combination of evidences similar to what is done by a Naïve Bayes Classifier. It can be expected that it is difficult to apply this approach to recognize situations that involve a time series of states. Furthermore, to the best of our knowledge, this approach has not been incorporated in a more general framework for context management and reasoning. Thus, it is not shown how these concepts can be incorporated with approaches for dynamic discovery and integration of context providers and consumers, with ontologies or with heterogeneous data representations.

6.1.8 Summary

	ASC/ CoOl[1,3]	CML/ PACE	Context Toolkit	CFN	Gaia	ECORA	UC Dublin[2]
Dynamic Environment	✓	(✗)	✓	✓	(✓)	(✗)	✗
Common Vocabulary	✓	(✗)	(✗)	✗	✓	✗	✗
Interpretation of Data Structures	✓	✗	✗	✗	✗	✗	✗
Conversion of Data Structures	(✓)	✗	✗	✗	✗	✗	✗
Expression of Offers and Needs	(✗)	(✗)	✗	(✓)	✓	(✗)	✗
Competitive Fusion	✗	(✗)	✓	(✗)	✓	✓	✓
Reasoning Schemes	✗	(✓)	(✓)	(✗)	✓	✓	✓

✗ = no support (✗) = very limited support (✓) = partial support ✓ = full support

1 = focus on ontology model, not part of a comprehensive framework

2 = focus on fusion and reasoning, not part of a comprehensive framework

3 = selected as representative for the other ontology-based approaches

Table 6.1: Properties of Context Management and Reasoning Systems

In Section 6.1.1 to Section 6.1.7 we have reviewed a number of context management and reasoning approaches and have highlighted their main contributions and their weaknesses. The results of our evaluation with respect to the requirements identified above are presented in Table 6.1.

Several projects (ASC/CoOL, ContextToolkit, CFN, and Gaia) do a fairly good job in addressing the challenges resulting from a dynamic pervasive computing environment with appearing and disappearing context providers and consumers. There are also a number of approaches (ASC/CoOL, SOCAM, CoBrA, Gaia) that facilitate independent development of context components by establishing a common vocabulary based on ontologies. However, the ASC/CoOL model stands out as it is, to the best of our knowledge, the only approach in the literature that explicitly addresses the problem of heterogeneous representations of context information and conversions between them. All other approaches neglect the issue of heterogeneous representations and thus fail to meet important requirements resulting from the independent development of context components. Consequently, we use many concepts of ASC/CoOL as baseline for our information model and enhance them to allow a more elaborate specification of Inter-Representation Operations and a representation of uncertainty, impreciseness and unreliability of context information in terms of Dempster-Shafer belief functions.

The ContextToolkit, Gaia, ECORA, and the approach of UC Dublin are tailored to support competitive context fusion and provide reasoning approaches, which are partially able to consider the imperfect nature of context information. However, they are either based on approaches like Bayesian inference, fuzzy/probabilistic logic (Gaia) or ContextSpaces (ECORA), which lack the ability to represent partial or complete ignorance, or apply DST (Context Toolkit, UC Dublin) but fail to incorporate it with well-established reasoning schemes to derive high-level context information.

In this respect, the work presented in this dissertation has to be considered as a new comprehensive approach for context modelling, fusion and reasoning, which is tailored to meet the requirements resulting from independent development of context components and to explicitly take into account the imperfect nature of context information.

6.2 Cooperative Teams of Autonomous Mobile Robots

6.2.1 FIPA Specifications

The *Foundation for Intelligent Physical Agents* (FIPA) [36] is a standards organization fostering agent-based technology and the interoperability of its standards with other technologies. FIPA originates from a Swiss-based organization established in 1996 with the aim to produce software standards specifications for heterogeneous interacting agents and agent-based systems. The Swiss organization was dissolved in 2005 and an IEEE standards committee was set up to replace it.

In 2002, FIPA completed a process of standardizing a subset of all its specifications, which are a collection of standards intended to promote the interoperation of heterogeneous agents and the services they can represent. In total, 25 specifications have made it to the standardization stage, which cover a number of different aspects of agent interaction. For example, an *Agent Communication Language* [38] was standardized along with specifications for *Message Transport* [42, 41], *Message Representation* schemes [39, 40] and a *Content Language* [43] inspired by first-order and modal logics. Just as in our approach, it is assumed that the contents of messages can be interpreted with the help of an ontology, which defines the used symbols together with their relationships. All these concepts are complemented by architectural support in terms of an *Abstract Architecture Specification* [37]. Here, abstract means that no concrete architecture is proposed. Instead, architectural elements and their relationships are identified and described in such a way that they can serve as a guideline for the implementation of concrete systems.

Several agent platforms, as e.g. Jade [66] or JIAC [67], have adopted FIPA standards for communication and the most widely accepted standard is the *Agent Management and Agent Communication Language* (FIPA-ACL) specification.

Major contributions: The FIPA specifications provide a comprehensive set of standards covering a huge number of different aspects related to agent communication and interoperability. Standard approaches for message structuring and message representation schemes are provided and a common vocabulary in terms of an ontology to allow interpretation of the message contents is also envisaged. The FIPA also identified the need for architectural support and has provided the Abstract Architecture Specification. In conclusion, the FIPA standards have inspired many concepts of our work.

Weaknesses: There are two reasons why we do not rely more on FIPA standards in our work. First, the standards are tailored to agent communication whereas our objective is to provide an approach generally applicable to information exchange and fusion in dynamic and heterogeneous distributed environments. Second, the FIPA standards consider coordination, communication, and message exchange among heterogeneous agents mainly at a high symbolic level. Thus, the exchange of sensor data with measures for impreciseness like covariance matrices, value ranges, etc., is hard to realize on top of the existing FIPA support. Thus, at least a new content language would have been required. Another consequence of viewing communication only at a high symbolic level is that important aspects like sensor data fusion are not considered.

6.2.2 Probabilistic State Estimation of the AGILO RoboCuppers

In his dissertation, Schmitt proposes a probabilistic state estimation approach for cooperating autonomous robots [125], which has successfully been applied to the RoboCup [121] scenario as part of the software of the *AGILO RoboCuppers*.[1] As a major result of his work, algorithms for cooperative iterative incremental localization of the robots on the field and for cooperative object detection and tracking have been developed. Both algorithms utilize Bayesian filtering as underlying approach.

For this purpose, a number of different alternatives to realize the Bayesian filtering has been evaluated. The Kalman Filter [71] is selected as the most promising approach and is used to fuse the likelihoods derived from different observations. Opponent detection and tracking is realized by an enhancement of the Multi-Hypotheses Kalman Filter [118] approach.

One highlight of Schmitt's work is the rigorous and very accurate propagation of Gaussian distributions from the image processing level to the object fusion level. In order to propagate the Gaussian distributions through a series of non-linear functions, the *Unscented Transformation* [70] is used. In this respect, the concepts of Schmitt have been very inspiring for our work. However, we apply the *Unscented Transformation* to handle heterogeneity with respect to different data representations caused by independent development of information providers and consumers.

Major contributions: Bayesian filtering approaches have been evaluated and extended to be applicable for cooperative localization and object tracking. Thus, the approach has a sound theoretical basis. The involved Gaussian distributions have not been estimated by ad-hoc approaches at the object detection level but have rigorously been propagated through a series of non-linear transformations starting at the image processing level.

Weaknesses: In his work, Schmitt does not deal with aspects like sensor reliability or freshness of sensor data and it could be difficult to consider domain knowledge in the sensor

[1] AGILO RoboCuppers is the name of the *RoboCup Middle Size Team* of the TU Munich, Germany.

fusion approach, as we also intend to do (see Chapter 12). Besides, the proposed approaches are tailored to robot localization as well as object detection and tracking. They are not available as part of a sensor fusion framework, which is generically applicable to a number of different problems. However, it has never been the intention of Schmitt to provide such a framework.

6.2.3 Multi-Robot Cooperative Object Localization Approach of ISocRob

At the RoboCup Symposium 2009, Santos and Lima presented a new multi-robot/sensor cooperative object detection and tracking method [124]. It is based on a decentralized Bayesian approach and uses Particle Filters to avoid simplifying assumptions about the object motion and the sensors' observation models. However, in order to save communication bandwidth, a reduced dimension representation of the sample belief is used to exchange information about the object's location among the teammates. For this purpose, the sensors' particles are approximated as *Gaussian Mixture Model* (GMM) with the help of the *Expectation Maximization* algorithm [28].

The approach distinguishes between a local filter and a team filter. The local filter receives the GMMs from the robot's teammates and mixes the particles representing its own belief with the particles sampling the received GMMs. All particles are weighted by the local observation model and the best ones are re-sampled for the next local iteration. The team filter receives GMM representations of the object in the world frame and fuses them by performing Covariance Intersection among the GMM components. The estimate of the local filter is used when the sensor detects the object in order to improve its estimate from the teammates' observations. The team estimate is only used when the sensor does not detect the object alone. In order to prevent fusion of incorrect estimates, the disagreement between different beliefs is calculated applying the approach of Beigi et al. [9].

Together with the work of Schmitt [125] reviewed in Section 6.2.2, the approach of Santos and Lima constitutes the state of the art of cooperative object localization in the RoboCup MSL.

Major contributions: Whereas many RoboCup MSL teams use ad-hoc approaches for establishing a common world view, Santos and Lima propose a theoretically sound approach for cooperative object localization. The approach improves the estimates of the local sensor by incorporating the information of the teammates. Here, also errors in the robots' self-localization are taken into account.

Weaknesses: A major problem with using information of remote sensors to improve the localization estimate of a highly dynamic object (the ball) is network latency, which causes the received information to be outdated. This problem, however, is not addressed at all. Besides, the approach has been designed for a homogeneous team of robots. Additional challenges arising from heterogeneous data representations and coordinate systems have not been discussed.

6.2.4 Skills, Tactics and Plays (STP) and TraderBots

In [68], Jones et al. introduce the *Pickup Team Challenge*. It consists of the dynamic formation of heterogeneous robot teams, which execute coordinated tasks where a priori little information is known about the tasks, the robots, and the environments and where the interaction between individual robot developers is minimal. The authors explicitly state that much of the existing work on multi-robot coordination assumes that the robot team is built by a group of people working closely together over an extended period of time and that they are unaware of any work focused on the principles of building such highly dynamic teams with minimal interaction between developers.

Furthermore, it is argued in [68] that there are several reasons for building heterogeneous robot teams in a dynamic fashion. First, it is impractical to develop large teams or teams of expensive robots at the same site and at the same time, which currently hinders multi-robot research. Second, robots may be needed for emergency tasks where there may be insufficient time to hand-engineer the coordination mechanisms before task execution. Third, as robots fail, get lost, or otherwise malfunction, it is often necessary to substitute or add new robots.

Their approach to address the *Pickup Team Challenge* combines the *Skills*, *Tactics* and *Plays* (STP) approach proposed by Veloso et al. [12] and *TraderBots* [31]. STP was developed for controlling autonomous robot teams in adversarial environments. Teamwork, individual behavior and low-level control are decomposed into three separate modules. For the *Pickup Team Challenge*, the most important concept are *Plays*. Each play describes a fixed team plan comprising a sequence of actions for each role in the team to achieve the team goals. A boolean evaluation function allows to determine the applicability of a play in the current situation. Team strategy consists of a set of plays, called a *Playbook*, of which the team can execute only one play at any instant of time.

TraderBots, developed by Dias and Stentz [31], is a coordination mechanism which was designed to inherit the efficacy and flexibility of a market economy and to exploit these benefits to enable robust and efficient multi-robot coordination in dynamic environments. The team aims to complete tasks successfully while minimizing overall costs, and each robot aims to maximize its individual profit. To solve the task allocation problem, the robots run a kind of task auctions and bid on tasks in other robots' task auctions.

Three kinds of components are used to enable teamwork in the pickup team: *RoboTrader*, *PlayManager* and *RobotServer*. The *RobotServer* provides an interface between the *PlayManager* and the components on the robot responsible for controlling the robot. Thus, the *RobotServer* can be considered as a kind of proxy between the teamwork infrastructure and the robot control systems in a similar way as it is proposed in STEAM/Teamcore [112].

Major contributions: The *Pickup Team Challenge* has been defined and several reasons have been provided to justify why it is beneficial to dynamically form heterogeneous teams of robots. A combination of the STP approach and TraderBots has been introduced as first attempt to address this challenge.

Weaknesses: Although a teamwork model and infrastructure is provided that can be deployed on top of existing robot platforms, the problem of data exchange among the heterogeneous robots is almost neglected. Thus, teamwork is mainly viewed in the perspective of coordinated tasks that can be executed without establishing a common view on the world based on the sensor information of the different robots involved. However, we consider this as very important to solve more complex tasks in environments that require high quality perception.

6.2.5 The Sensor Fusion Effects (SFX) Architecture

Sensor Fusion Effects (SFX), proposed by Murphy [89], is a reusable generic sensor fusion approach for autonomous mobile robots, which is suitable for a wide variety of sensors and environments. In particular, it is targeted at autonomous mobile robots operating in only partially known or unknown environments. Within SFX, sensor fusion consists of three distinct activities: planning, execution, and exception handling. In the planning activity, the task goals of the robot are used to generate expectations of percepts and to predict what features will be observable to which sensors. The execution activity is concerned with collecting observations and computing the total belief in the percept. Depending on the belief, the robot proceeds with a behavior to accomplish a goal, explores further or adds more sensor resources, or reconsiders its goals. In addition, the execution activity tries to detect sensing anomalies such as sensor malfunction. If an anomaly is detected, an appropriate exception handling activity is invoked.

Sensor fusion in the SFX architecture is based on DST, where the accrual of evidence follows a three level hierarchy. At each level, evidence is represented as a Dempster-Shafer belief function. At the lowest level, evidence is collected for features, which may contribute to a percept and can directly be collected from the processing of sensor data. At the second level, the evidences for features are transferred to evidences for descriptions of a percept. At the highest level, these evidences are converted to evidences for the percept and combined with Dempster's rule. Propagation of the evidences between the different levels is accomplished with the help of enlargement functions that encode evidential mappings between different frames of discernment. In SFX, the evidential mappings are implemented as weight vectors, which are selected according to some predefined rules.

In [89], Murphy discusses several sensor fusion schemes for their applicability for autonomous mobile robots. In particular, he compares the suitability of Bayesian methods with Dempster-Shafer Theory. He argues in favor of DST because it facilitates modelling of full or partial ignorance, does not require a priori probabilities and allows to include domain-specific knowledge and dependencies which are not of probabilistic nature.

Major contributions: The SFX architecture is not tailored to a concrete sensor fusion task as part of a particular application, but provides a reusable generic sensor fusion architecture for autonomous mobile robots. The benefits of DST compared to other sensor fusion schemes are highlighted. It is also shown how Dempster-Shafer belief functions can be propagated between different levels of abstraction.

Weaknesses: Sensor fusion is mainly viewed in the perspective of a single robot. Additional challenges arising from the inherent distribution of sensors in a team of cooperating robots are not discussed. Besides, the sensory system is assumed to be designed in a monolithic fashion. Consequently, heterogeneity issues arising from independent development of the different sensor components are neglected as well.

6.2.6 Ecology of Physical Embedded Intelligent Systems

The concept of an *Ecology of networked Physically Embedded Intelligent Systems* (PEIS) [110] aims at putting together insights from the fields of ambient intelligence and autonomous robotics in order to generate a new approach for building assistive, personal and service robots. In contrast to many current approaches which aim at building a single robot device empowered with extraordinary capabilities for perception, action, and cognition, the PEIS-Ecology approach generalizes the notion of a robot to encompass the entire environment. In the PEIS-Ecology vision, the robot disappears in the environment in a similar way as computers are assumed to disappear in the vision of ubiquitous computing.

For this purpose, all robots in the environment are abstracted by the notion of a physical device including a number of functional components. Here, the term robot is used in its most general notation and interpreted as as computerized system interacting with the environment through sensors and/or actuators. All PEIS elements are linked through a cooperation model based on functional components. Each participating PEIS element can offer functionalities to or use functionalities from other PEIS elements in the environment in order to improve the capabilities of the PEIS-Ecology.

In this respect, the PEIS-Ecology approach is close to our vision where information providers (sensors) and information consumers (actuators) are available in the environment and are dynamically linked together in order to use the offered information (offered functionalities) for satisfying information needs (required functionalities).

As part of the project, a middleware [16] is developed which provides support for dynamic configuration of functionalities, for knowledge representation and for realizing the communication among the involved entities. A functionality is specified through its inputs, outputs, preconditions, postconditions, a transfer function and its costs. For dynamic configuration of functionalities, a hierarchical planning approach from artificial intelligence is used [92]. This is close to our approach, where we aim at bridging heterogeneity with respect to different representations by providing a configuration of Inter-Representation Operations.

In [76], an approach for cooperative anchoring in heterogeneous multi-robot systems is proposed. Cooperative anchoring means the problem of determining which items of information in a distributed system refer to the same objects and combining these items accordingly. In this approach, information is represented using geometric spaces, which are inspired by Gärdenfor's conceptual spaces [45], and concepts for data association, data fusion and prediction are provided. Futhermore, an implementation of the concepts based on fuzzy sets is discussed.

Major contributions: The PEIS-Ecology approach is able to dynamically combine functional components available in the environment in order to realize the overall functionality of a robotic system. It also deals with cooperative anchoring in multi-robot systems including the problem of data fusion. Considering the functionalities as information provisioning and consumption, it shows a number of similarities to our approach and fulfills many of the requirements identified above.

Weaknesses: The PEIS-Ecology approach is claimed to be tailored to incorporate heterogeneous functional components. However, independent development of these components is only considered to a limited extent. Functionalities are specified involving a logic-based formalism and it is allowed to specify representations for inputs and outputs. This can be assumed to serve as a kind of common vocabulary for developers, but this has not been explicitly discussed. Data fusion based on fuzzy sets has been proposed, but it is assumed that all robots use the same data representation avoiding the need to transform the measures for impreciseness.

6.2.7 Summary

	FIPA	Schmitt AGILO[1]	Santos ISocRob[1]	SFX[1]	STP/ TraderBots	PEIS
Dynamic Environment	✓	✗	✗	✗	✓	✓
Common Vocabulary	✓	✗	✗	✗	✗	✓
Interpretation of Data Structures	✓	✗	✗	✗	✗	✗
Conversion of Data Structures	✗	✗	✗	✗	✗	✓
Expression of Offers and Needs	✓	✗	✗	✗	✓	✓
Competitive Fusion	✗	✓	✓	✓	✗	✓
Reasoning Schemes	✗	✓	✓	✓	✗	✗

✗ = no support ✗ = very limited support ✓ = partial support ✓ = full support

1 = focus on data fusion and reasoning

Table 6.2: Properties of Approaches and Frameworks for Autonomous Mobile Robots

Table 6.2 presents the detailed results of our evaluation of the approaches discussed in Section 6.2.1 to Section 6.2.6 with respect to the requirements identified above.

The approaches of Schmitt (*AGILO RoboCuppers*) and of Santos and Lima (*ISocRob*) represent the state of the art in cooperative object localization and tracking in the RoboCup Middle Size League. Both approaches are based on probability theory and utilize Bayesian filtering techniques as underlying sensor fusion approach, and thus suffer from the drawbacks of traditional probability theory. These are, for example, the difficulty to represent partial or complete ignorance or to incorporate non-probabilistic domain specific knowledge. Besides, the approaches are tailored to the specific task of cooperative object localization in the RoboCup domain and do not provide a generic sensor fusion approach. In contrast, the SFX architecture provides generic and reusable sensor fusion and reasoning schemes and is based on DST exploiting all its benefits. However, all three approaches assume a statically composed system and do not consider the additional challenges arising from a dynamic ubiquitous computing environment and from independent development of information providers and consumers.

The FIPA specifications and STP/TraderBots aim at supporting interoperability and cooperation of independently developed and heterogeneous autonomous mobile systems. Here, STP/TraderBots even more emphasizes the requirement of achieving interoperability, when the interaction of the different development teams is minimal. Both approaches are able to incorporate possibly heterogeneous systems which dynamically join the team. However, interoperability and cooperation is mainly considered from a high-level perspective focusing on the exchange of information at the symbolic level and on the communication about goals, tasks and plans. Exchange of sensor data with its measures of uncertainty and impreciseness as well as appropriate sensor fusion and reasoning schemes for deriving high-level information are discussed only to a very limited extent, if at all.

The PEIS approach is closest to our vision of autonomous mobile systems, which are dynamically composed and configured at runtime. PEIS composes functionalities already available in the environment to realize the overall functionality of the system. Our approach aims at providing requested information from the information available through sensors in the environment and bridges heterogeneity issues by applying (possibly a series of) Inter-Representation Operations. The PEIS approach meets many requirements arising from a dynamic ubiquitous computing environment and presents approaches for data fusion. However, a common vocabulary serving as a baseline for independent development is provided only implicitly through a logic-based definition of functionalities. Although the developers of PEIS envisage functionalities that can perform coordinate transformations and specify the type of data representations, semantically interpretable data structures and the transformation of measures of uncertainty and impreciseness are not supported. Instead, with their data fusion approach based on fuzzy logic they explicitly assume a common data representation.

In conclusion, the work described in this dissertation represents a further ingredient to realize teams of heterogeneous autonomous robots, which are composed dynamically at runtime and where interaction among the different development teams is minimal. Our approach focuses on the exchange of sensor data and on bridging heterogeneity with regard to the utilized data representations, which has been neglected by the other approaches. Furthermore, we present a generic and reusable information model and provide a generic data fusion and reasoning approach based on the DST.

Part II

Solution

7 Information Model

In this chapter, we describe the Information Model which forms the baseline of our solution approach. Generally, we assume a message-based communication scheme between the different nodes of the computing environment, where a piece of information can be encapsulated in a single message. Communication between the different nodes is enabled by using a standard communication protocol or an appropriate communication middleware. Our description of the Information Model is centered around the second option and shows how the proposed Information Model directly fits within the concepts of SPICA [6, 3], a model-driven development framework for communication infrastructures involving heterogeneous nodes (see also Section 2.2.3).

7.1 Three Layers of Abstraction

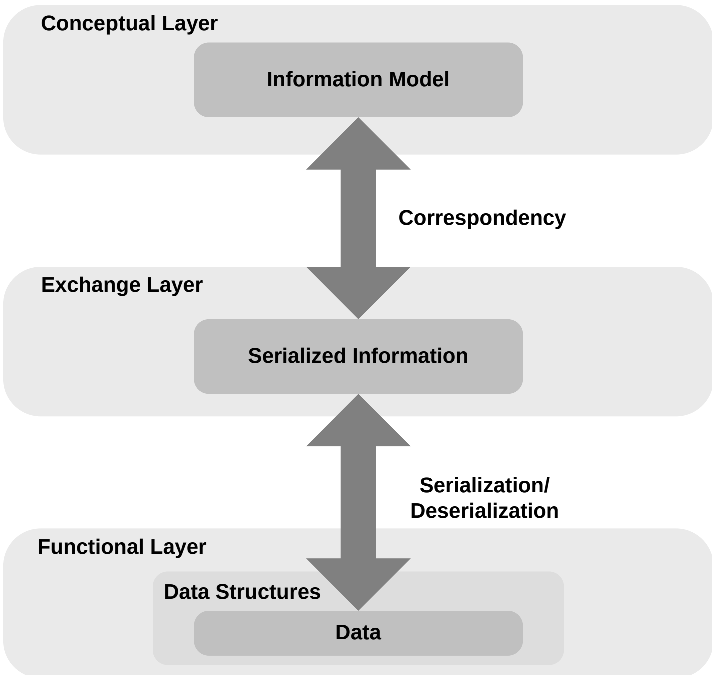

Figure 7.1: Three Layers of Abstraction

We identify three basic layers of abstraction for the realization of information exchange in heterogeneous distributed environments: the conceptual layer, the exchange layer, and the functional layer as shown in Figure 7.1. The conceptual layer is formed by the Information Model, which is an ontology specified in OWL 2 and provides a common vocabulary defining the application domain as well as the structure and semantics of the exchanged information.

At design time, this vocabulary is leveraged by the developers for the specification and semantic annotation of the exchanged messages. At runtime, it is used to perform a number of reasoning tasks, e.g. checking the availability of and determining potentially required *Inter-Representation Operations*. The exchange layer aims to be utilized for interoperability between different nodes. At this layer, the information is expressed/serialized in any adequate exchange format such as XML, JSON (JavaScript Object Notation) [69] or the SPICA binary format [3], to mention only a few. Finally, the functional layer refers to the actual implementation of the Information Model and the internal mechanisms used in the different nodes. It provides the required serialization and deserialization mechanisms to bridge between the functional and the exchange layer. The main objective of this layer is efficiency, both in terms of processing speed and resource consumption.

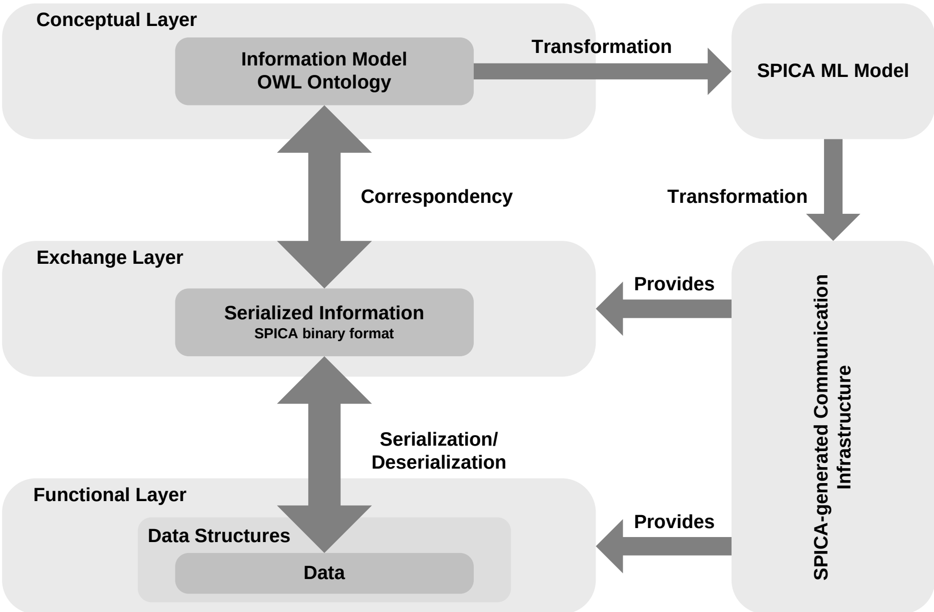

Figure 7.2: Three Layers of Abstraction and SPICA-based Realization

As shown in Figure 7.2, one possibility for the realization of the exchange layer and the functional layer is a communication infrastructure that is generated by the SPICA development framework (see also Section 2.2.3). SPICA provides a specification language called SPICA Modelling Language (SPICA ML) which is similar to IDL approaches and ASN.1 [65]. From a SPICA ML model, the data structures for the exchanged messages along with the serialization and deserialization methods as well as module stubs acting as communication endpoints can be generated in different programming languages. This and its support for easy interfacing of already existing modules makes SPICA particularly well suited for providing a communication infrastructure in heterogeneous environments. As the structure of the information to be exchanged is entirely defined in the ontology, the definitions of the ontology can be automatically transformed into the message part of a SPICA model acting as starting point for the code generation. However, SPICA does not provide the capabilities required for the further processing of the received data as envisaged in this work.

In the following sections, we focus on the description of the ontology concepts and provide further hints how the definitions can be transformed into the message part of a SPICA ML model.

7.2 Ontology Concepts

The main purpose of the ontology concepts presented in this section is to provide a common scheme for the ontology-based definition of application domains and for the specification of the data structures used for information exchange. In particular, this means that we do not propose a single concrete ontology. Instead, we only propose top-level concepts that form the baseline for concrete ontology specifications and allow easy integration with existing ontologies.

As the information exchanged between heterogeneous and independently developed information sources and sinks is intended to be processed and to serve as input for calculations and reasoning tasks, it has to be ensured that the exchanged information can be fully semantically interpreted. The corresponding ontology concepts are inspired by the *Aspect Scale Context (ASC)* model and the *Context Ontology Language (CoOL)* proposed by Strang et al. [135]. ASC/CoOL envisage the specification of different representations (scales) for a certain information element (aspect) and defines the internal structuring of the representations in the ontology. As this fits very well our requirements, we adopt a number of concepts from the ASC model and incorporate them into a more comprehensive ontological framework. More detailed information about the concepts adopted from ASC is provided in the following subsections.

7.2.1 Structuring of Exchanged Information

In our approach, we assume a simple structuring of information as depicted in Figure 7.3. A piece of information contains an arbitrary number (but at least one) *Information Elements*. In turn, an *Information Element* is composed of further *Information Elements* and of an arbitrary number of information *Values*. *Values* correspond to atomic values of a certain data type. In order to enable semantic interpretation of the information, the ontology provides a common vocabulary and modelling concepts used to define the different types of *Information Elements* and their internal structuring with regard to the contained *Information Elements* and *Values*.

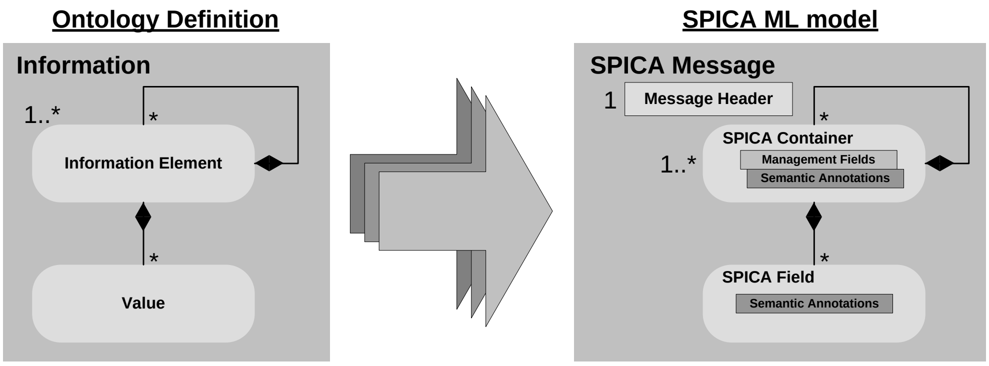

Figure 7.3: Structuring of Exchanged Information

Figure 7.3 also illustrates that the structuring of information assumed in our approach is in line with the structuring of information applied in SPICA. In SPICA, an *Information Element* corresponds or is represented by a *SPICA Container*, which is a collection of further *SPICA Containers* and *SPICA Fields*. Just as in our Information Model, *SPICA Fields* correspond to atomic *Values* of a simple data type such as integer, float, uint8, etc. SPICA also envisages the possibility of annotating the different elements of a message. This means that references to concepts (individuals and classes) of our ontology can be included in order to support semantic interpretation of the SPICA data structures. A *SPICA Message* also comprises a *Message Header* as well as some *Management Fields* for each container. These provide identifiers for the message or container, depict the used serialization/deserialization scheme, etc.

7.2.2 Entity Types, Scopes and Representations

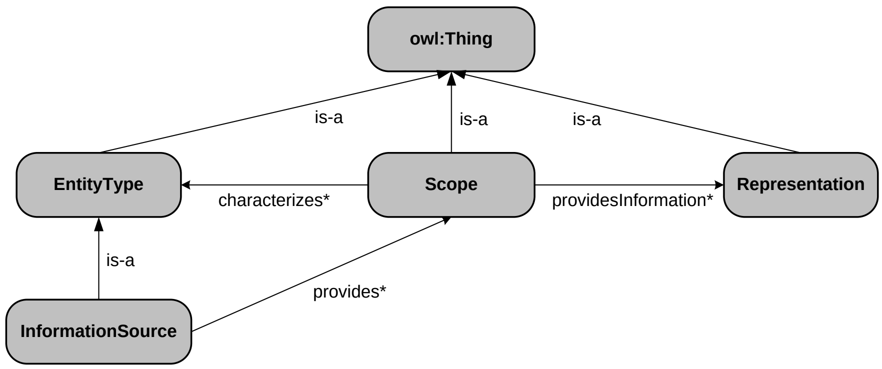

Figure 7.4: Ontology Concepts: Entity Types, Scopes and Representations

Support for representing heterogeneous sensor information is one of the main requirements for the Information Model presented in this section. Sensor information usually provides a certain kind of information about a real or logical object of the world in a certain representation. Consequently, at the top level the ontology defines the classes *Entity Type*, *Scope* and *Representation* (see Figure 7.4). *Entity Types* depict the types of real or logical entities in the application domain. In the domain of context-aware ubiquitous applications, examples for *Entity Types* are *User*, *Device*, *Building* and *Application*. These types abstract concrete *Entities* like '*User Christoph*', '*Device with IP 141.51.122.44*', '*Building of University of Kassel*' or '*Application named ActivityAwarePhone*'. *Entities* are characterized by *Scopes*, which indicate the type of information elements. Examples for *Scopes* are *LocationInfo*, *BatteryInfo*, *DateTimeInfo*. *Scopes* provide the information in a defined *Representation*, which describe the internal structuring of the information. Here, special attention has to be paid to the multiplicity of the property *providesInformation*: a *Scope* can provide an arbitrary number of information elements of the corresponding type, each of which has a certain representation. In this respect, a *Scope* can also be interpreted as a collection of information elements of a certain type that can be encapsulated in a message. As described in Section 7.2.5, this is very important, since it allows to represent multiple information elements corresponding to multiple hypotheses of the same phenomenon in a single message. This is required for information fusion according to DST which relies on the association of basic belief masses to different hypothesis of a phenomenon. A *Scope*, i.e. an information element or a collection of information elements of a certain type, is provided in a certain representation by an *Information Source*. This class is a specialization of *Entity Type* and mainly represents raw sensors (including persons) or devices that perform calculations on raw sensor data. As also logical *Entities* are allowed, an *Information Source* can also be a certain reasoning component deployed on a node of the computing environment. It is also noteworthy here that an *Information Source* can also be a characterized *Entity* of a *Scope*. For example, a robot determines its position by a self-localization approach, i.e. the robot provides information on its own position.

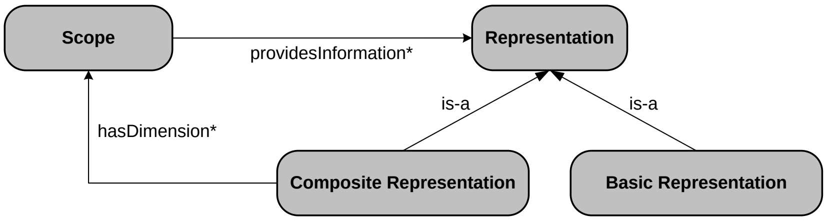

Figure 7.5: Ontology Concepts: Basic and Composite Realizations

Whereas *Scopes* correspond to semantic concepts for information types, *Representations* provide semantic definitions of the data structures, or containers in SPICA terminology, used to represent the information of a certain type. We distinguish between *Basic Representations* and *Composite Representations* as shown in Figure 7.5. *Composite Representations* comprise an arbitrary number of dimensions corresponding to different types of information elements, i.e. *Scopes*. This allows to specify that an information element of a certain type in a certain representation includes further information elements, which in turn correspond to scopes and are provided in a certain representation. The recursion stops once a dimension (*Scope*) is provided in a *BasicRepresentation*, which does not comprise further dimensions. Usually, a *Basic Representation* corresponds to a basic information value, or field in SPICA terminology, that can be represented using an OWL datatype property.

This definition of the internal structuring of information is adopted from the ASC model [135]. The ASC model defines *Aspects* (corresponding to our *Scopes*), which are represented in a certain *Scale* (corresponding to our *Representation*). *Scales* in the ASC model refer to *Aspects* again, which leads to a recursion of the same kind as in our model.

A *Composite Representation* represents an information element which is of a certain *Scope* and characterizes a certain *Entity* by referring to other Scopes. These *Scopes* in turn characterize *Entities*. For example, a *SharedWorldInfo* scope in the RoboCup domain provides information about the current world view of a particular robot. A possible representation of *SharedWorldInfo* may comprise a dimension of scope *PositionInfo* depicting the position of the ball on the field. This means that a corresponding element of scope *SharedWorldInfo* characterizes the robot providing the information and the ball. In general, an information element characterizes all of the entities that are described by an included information element. In OWL 2, this can be represented by defining the following property chain as sub-property of *characterizes*:

$$\boxed{\textit{providesInformation} \circ \textit{hasDimension} \circ \textit{characterizes} \rightarrow \textit{characterizes}}$$

Just in the same way as the ASC model, we allow different *Representations* for a certain *Scope* (see Figure 7.6). For example, the scope *LocationInfo* depicting the current location of a person can be represented in GPS coordinates (*LocationWGS84*) or as the address of a building (*LocationAddress*). In the same way, a *DateTimeInfo* may be represented as a tuple of day, month and year, or as a simple string. Allowing multiple possible representations for

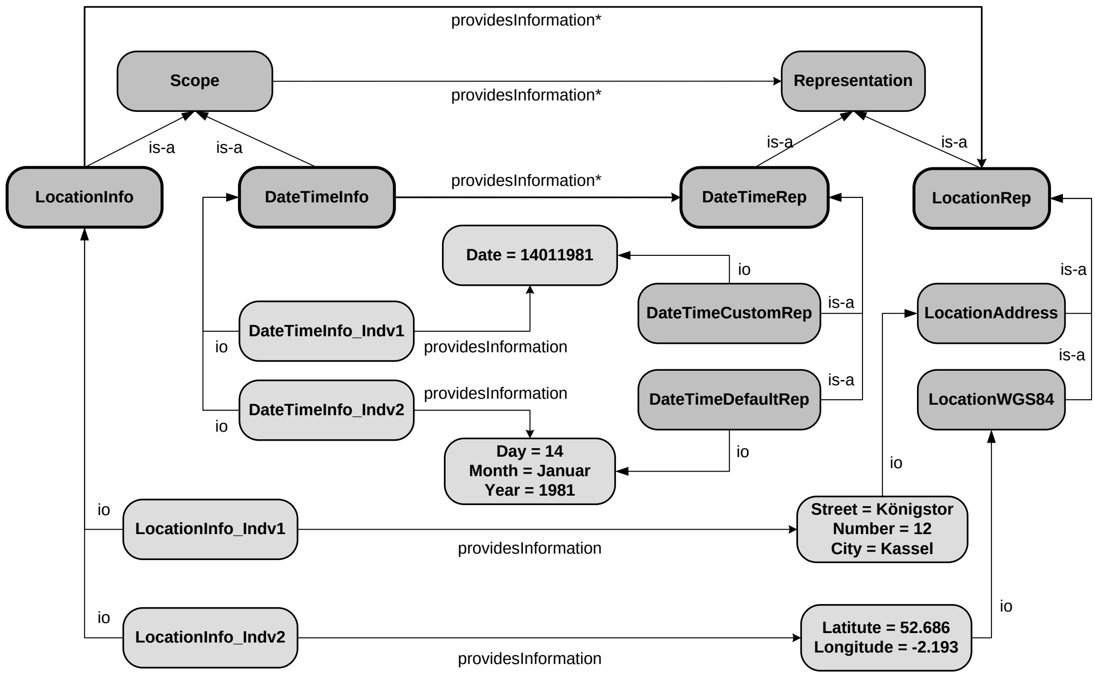

Figure 7.6: Ontology Concepts: Multiple Representations

a certain type of information is one of the most important requirements for our Information Model, as we cannot assume that independent developers always use the same representation for a certain type of information. In this respect, support for multiple representations is a prerequisite to boost interoperability for dynamically discoverable and heterogeneous information consumers and providers.

In this work, the ASC model is extended by introducing taxonomies for *Scopes* and *Representations* and restricting the allowed *Representations* for a certain *Scope*. Restricting the allowed *Representations* for a certain *Scope* to a certain class of *Representations* does not prevent the possibility to allow heterogeneous representations for *Scopes*. As illustrated in Figure 7.6 possible heterogeneous representations are covered as subclasses of the *Representation* class the *Scope* is restricted to. In the example of Figure 7.6, a restriction for the class *LocationInfo* can be defined in OWL 2 as:

providesInformation **onlyValuesFrom** *LocationRep*

It is noteworthy here that through restrictions of this kind, the taxonomies for *Scopes* and possible *Representations* are close-coupled.

In order to further illustrate the ontology-based definition of representations, a concrete example is provided in Figure 7.7. In this example, the *CNPositionInfoRep* is intended to provide a data structure for position information of objects in a 3D Cartesian coordinate system. Consequently, the *CNPositionInfoRep* refers to the *Scopes XCoordinate*, *YCoordinate*, and *ZCoordinate*, which are defined as subclasses of *CoordinateSystemScope*. Actually, a *Representation* refers to *Scopes* by the *hasDimension* property (see Figure 7.5). However, in order to provide a semantically more meaningful property and to ease definition of

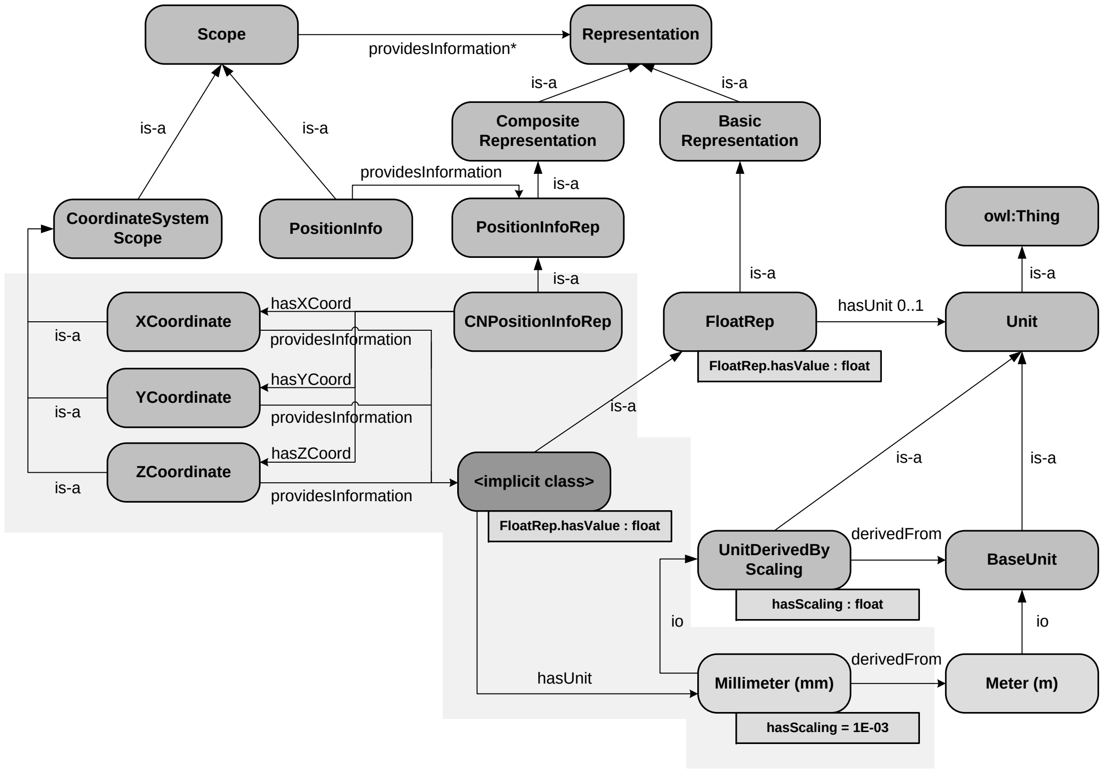

Figure 7.7: Example for the Detailed Specification of Representations

cardinalities[1], we define three sub-properties of *hasDimension*: *hasXCoord*, *hasYCoord*, and *hasZCoord*. Without further restrictions, the *XCoordinate*, *YCoordinate* and *ZCoordinate* scopes can be provided in an arbitrary representation. This would mean that *CNPositionInfoRep* only specifies the involved scopes, but does not provide further information about the concrete representation of the internal scopes. From a conceptual point of view this is possible, but in most practical cases also the need for a concretion of the representations of the internal scopes can be assumed. In our example, all three coordinates should be represented as floats in the unit millimeters. In order to be able to refer to units, we integrate an ontology from the NASA SWEET project [137]. The SWEET Units Ontology defines classes for *BaseUnits* and *DerivedUnits*. *DerivedUnits* refer to one or more *BaseUnits* and incorporate *Prefixes*, as e.g. *Kilo*, *Mega* or *Milli*, and the corresponding scaling factors. This allows conversion between the different units only with the information available in the ontology. Concrete units like *Meter* and *Millimeter* appear as individuals in the SWEET ontology. In order to specify that the *XCoordinate*, *YCoordinate* and *ZCoordinate* scopes involved in the *CNPositionInfoRep* are represented as floats and in millimeters, property restrictions of the following kind have to be defined for *CNPositionInfoRep*:

[1] It has to be ensured by cardinality restrictions that the Representation comprises exactly one XCoordinate, one YCoordinate and one ZCoordinate.

hasXCoord	**onlyValuesFrom**						
	(*XCoordinate*	**and**	(*providesInformation*	**onlyValuesFrom**			
	(*FloatRep*	**and**	(*hasUnit*	**hasValue**	*Millimeter*))))		

The third line of this property restriction corresponds to the definition of an implicit class, which is depicted in Figure 7.7. The definition of *CNPositionInfoRep* from the example (gray background) can easily be translated into a SPICA ML container specification:

```
urnpfx # http://carpenoctem.das-lab.net/ontologies/SimpleExample.owl#

CNPositionInfoRep : DataContainer
      [compare={XCoord, YCoord, ZCoord}, rep=#CNPositionInfoRep] {
            float XCoord [refconcept=#XCoordinate,
                              refrep=#FloatRep{hasUnit=#Millimeter}];
            float YCoord [refconcept=#YCoordinate,
                              refrep=#FloatRep{hasUnit=#Millimeter}];
            float ZCoord [refconcept=#ZCoordinate,
                              refrep=#FloatRep{hasUnit=#Millimeter}];
}
```

The SPICA ML container specification defines a data container named *CNPositionInfoRep* which is annotated with the key words *compare* and *rep*. *compare* lists the variables that are considered in a comparison for equality, *rep* refers to the ontology and depicts the corresponding *Representation* class. The container includes three *float* fields *XCoord*, *YCoord*, and *ZCoord*, the names of which have been derived from the properties *hasXCoord*, *hasYCoord* and *hasZCoord* defined in the ontology. All three fields are annotated with the key words *refconcept* and *refrep*. *refconcept* refers to the ontology classes for the involved *Scopes* and *refrep* depicts the ontological class for the corresponding *Representation*. In the same way as in the ontology, the class for the *Representation* is an implicit class derived from *FloatRep* with property *hasUnit* restricted to the individual *Millimeter*. In this respect, the ontology-based specification of representations, or data structures respectively, directly fits within the SPICA model-driven development approach and forms another abstraction layer for the message part of SPICA ML with focus on semantic interpretability. Another possibility which has to be investigated in future is the transformation of SPICA ML models into ontology specifications as this would allow developers to specify data structures in the way they are used to, and still to exploit the benefits of our Information Model.

It is also noteworthy here that all the ontology concepts presented above are designed in such a way that a concrete piece of information represents an individual of a certain scope of the ontology. This can be exploited to combine our Information Model with approaches applying ontology-based reasoning on individuals to infer new knowledge (see Section 7.2.4).

7.2.3 Entity Types as Value Domains

In many cases, the information given through a *Scope* corresponds to an individual of an *Entity Type*. A very prominent example for this is a scope *LocationInfo* that provides information about the location of a person. The characterized *Entity Type* of the *Scope* is *Person*, and one may want to express that the current location of the person is a particular room, e.g. *'Room 1305'* which represents an individual of the entity type *Room*. In a way, this would require to allow *Entity Types* as value domains for the *providesInformation* property, i.e. to consider *Entity Types* as a kind of *Representation*.

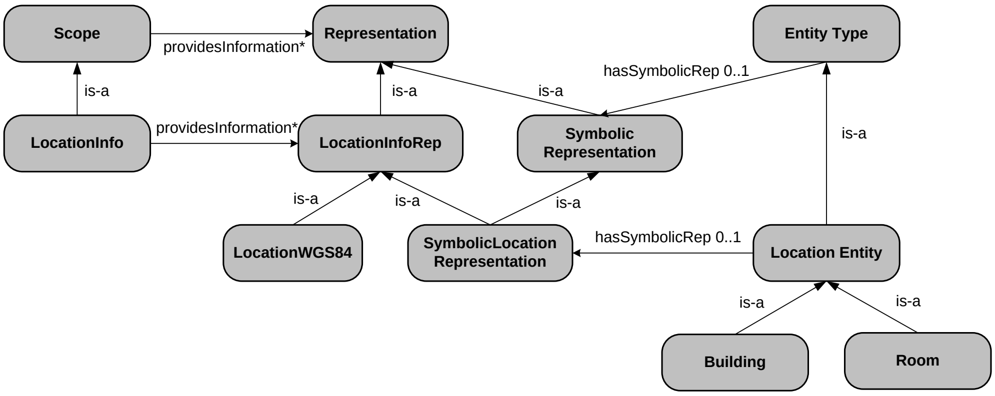

Figure 7.8: Symbolic Representations for Entities

In order to retain a clean scheme of *Entity Types*, *Scopes* and *Representations*, however, we introduce another concept/class *SymbolicRepresentation* in our Information Model as subclass of *Representation* (depicted in Figure 7.8). A *SymbolicRepresentation* provides a kind of symbol, e.g. a simple string[2], that uniquely identifies a certain *Entity*. For this purpose, *Entity Types* are related to *SymbolicRepresentations* through the functional property *hasSymbolicRep*. This ensures that if two *SymbolicRepresentations* are equal, they are associated to the same individual of the corresponding *Entity Type*. This is a first step towards the establishment of correspondencies between symbols and data values that can be perceived by sensors. This is commonly referred to as *perceptual anchoring* [25].

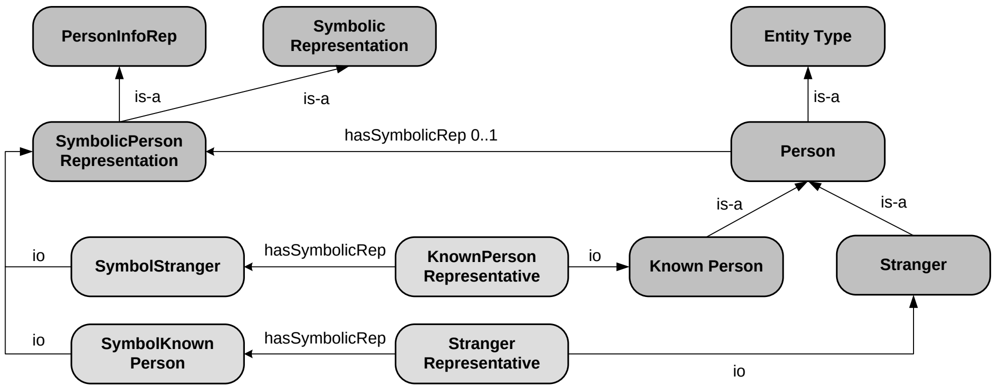

Figure 7.9: Representative Individuals for Classes

Apart from specifying that a person is currently located in a particular room or building, it may also be required just to express that the person is currently located in a class of location entities. For example, for some applications it may be sufficient just to know if a person is currently located in a restaurant or in a cinema, but no more details on the particular restaurant or cinema are required. A similar situation can be observed when a piece of information should express a classification result. For example, a person is classified as

[2]From a conceptual point of view, also *Composite Representations* are feasible.

a *Known Person* or as a *Stranger*. Here too, only a whole class of individuals is relevant. In a way, this requires to reference a whole class in an individual. However, only in the OWL Full sub-language (or in the Full Profile of OWL2 respectively) classes can be used as individuals. But OWL Full discards the guarantee for decidability. In order to avoid this, we introduce *Representative Individuals* (and associated symbols) for the different *Entity Types* (see Figure 7.9). These individuals can be used in the same way as other individuals of the class but represent the whole class of individuals.

7.2.4 Inference through Ontology Reasoning

As already explained in Chapter 5, ontologies are usually specified in a logic-based formalism. This also applies to the ontology concept presented in this thesis, which are described using a decidable fragment of OWL 2 corresponding to the $\mathcal{SROIQ}(D)$ description logic. Such a logic-based formalism can not only be exploited to check consistency of ontologies, but also to infer new knowledge based on the information already available through the classes, their relationships and the corresponding individuals of the ontology. In the area of context awareness, Wang et al. [145] apply this concept for reasoning about the activity of users in a home domain. A part of their ontology, called *COntext ONtology (CONON)*, is depicted in Figure 7.10.

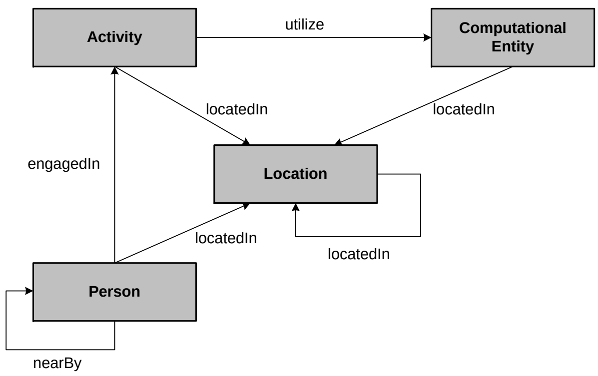

Figure 7.10: Part of the CONON Home Domain Ontology

In this ontology, *Location* is the central concept depicting that a *Person*, the utilized *Computational Device*, and the performed *Activity* all have the same location. A *Person* can be nearby other *Persons* and a *Location* can be contained in another *Location*. If a set of individuals of the different classes matches certain constraints, it can be inferred that some *Persons* are engaged in a certain *Activity*.

The property *locatedIn* plays a central role in this reasoning approach. With regard to our ontology concepts, so far we have only highlighted how *Entity Types*, *Scopes* and *Representations* are related to each other. Figure 7.11 shows how relationships between *Entity Types*, as for example represented through properties like *locatedIn*, fit into our scheme.

The general idea is to resolve the property *locatedIn* by specifying a chain of properties as sub-property of *locatedIn*. For this purpose, we first define the property *describedBy* as inverse property of *characterizes* and the property *identifies* as inverse property of *hasSymbolicRep*. By making *identifies* a functional property, a one-to-one correspondence between individuals

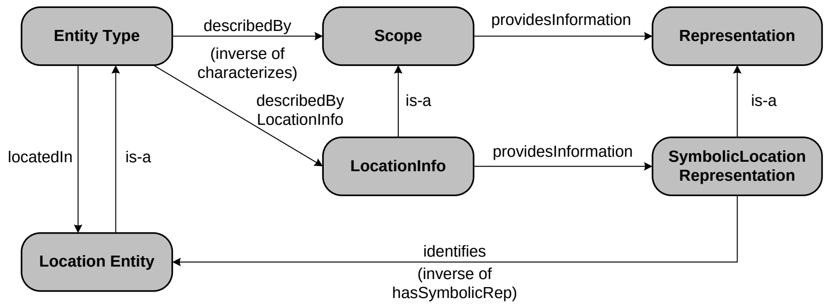

Figure 7.11: Property Chain for locatedIn Property

of an *Entity Type* and individuals of *SymbolicRepresentation* is ensured. Furthermore, the property *describedByLocationInfo* is defined as sub-property of *describedBy* in order to avoid possible ambiguities. The *locatedIn* property can now be represented by the following property chain:

$$\boxed{\textit{describedByLocationInfo} \circ \textit{providesInformation} \circ \textit{identifies} \rightarrow \textit{locatedIn}}$$

By resolving relationships (properties) between *Entity Types* in this way, ontology reasoning approaches as proposed by Wang *et al.* [145] can be directly combined with our scheme of *Entity Types*, *Scopes* and *Representations*. Section 7.2.7 describes how symbols are related to sensor values within this scheme as a general basis for applying reasoning approaches at a symbolic level.

7.2.5 Association of Meta-data

One of the main challenges for the work presented in this thesis is to deal with the special characteristics of sensor information. Thus, we define a number of *Metadata* classes in the ontology, which help in expressing the impreciseness, uncertainty, unreliability as well as the temporal and dynamic nature of sensor information (see Figure 7.12).

We distinguish between *RepMetadata*, which are only associated to or used in *Representations*, *ScopeMetadata*, which are only associated to *Scopes* and *CommonMetadata*, which can be associated to both, *Scopes* and *Representations*. Here it is important to note that when specifying meta-data, *Scopes* are interpreted as a collection of *Information Elements* of a certain type, and the individuals of the corresponding *Representations* are considered as the actual *Information Elements*, i.e. the values for the *Scope*. Consequently, meta-data associated to *Scopes* apply to the whole collection of *Information Elements*. So far, as *ScopeMetadata* we only use *Reliability*, which expresses the reliability of the sensor providing the *Scope*, and *BeliefAssignment*, which allows to form a complex Dempster-Shafer belief assignment from the belief masses of the single hypotheses, i.e. *Information Elements*. As *Metadata* for *Representations* we define *ValueRange*, *UniformValueRange* with a uniform belief mass distribution, *CovarianceMatrix*, *Covariance*, and *Variance*. These classes are used to denote the impreciseness of an *Information Element*. Furthermore, *BeliefMass* and its specialization *Probability* are sub-classes of *RepMetadata* expressing a Dempster-Shafer belief mass assignment or probability for a single *Information Element*. *Probability* is considered

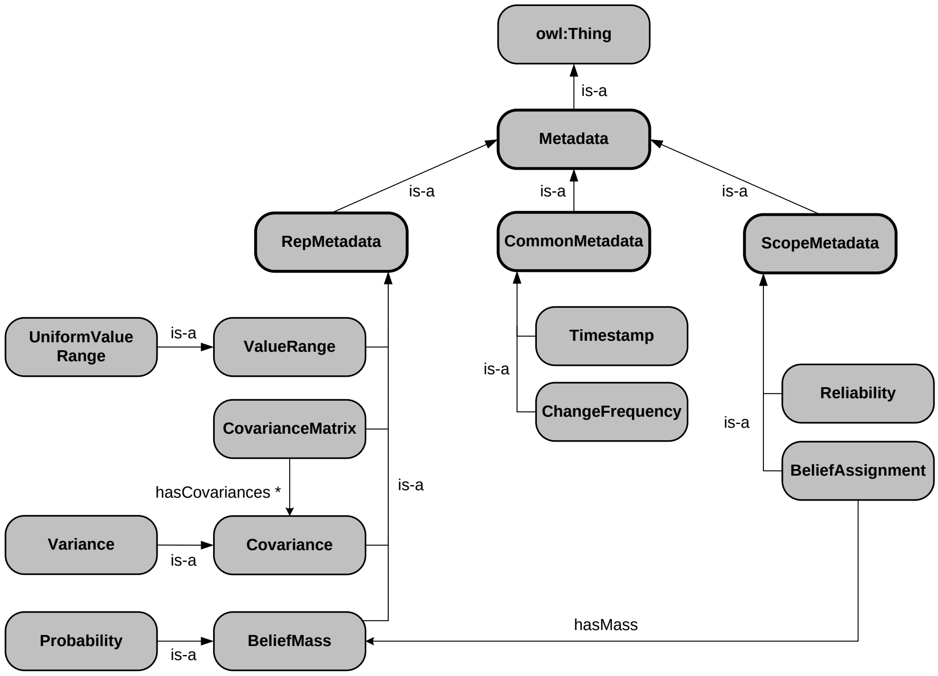

Figure 7.12: Different Types of Meta-data Defined in the Ontology

as a *BeliefMass* that only applies to a singleton of the domain (frame of discernment) of the corresponding *Information Element*. *CommonMetadata* has the subclasses *Timestamp* and *ChangeFrequency* depicting the temporal and dynamic nature of sensor information. Whereas the meaning of *Timestamp* is self-explaining, *ChangeFrequency* allows to express the expected frequency the observations represented in the *Information Elements* change considerably. As shown in Chapter 9, this information can be used in conjunction with a timestamp to model the belief in a piece of information that was sensed at a certain time in the past. *CommonMetadata* associated to a *Scope* apply for the whole collection of *InformationElements*, whereas *CommonMetadata* included in *Representations* are only valid for the particular *Information Element*.

From Figure 7.12 it can be observed that *Metadata* is directly derived from *owl:Thing*, although from a conceptual point of view it would be reasonable to consider *Metadata* as a special kind of information type and thus to put it a sub-class of *Scope*. However, deriving *Metadata* from *Scope* would mean that *Metadata* can have multiple *Representations* and that the internal structure has to be specified in the scheme *Scope* and *Representation* again. As we assume a quite fixed representation for most of the *Metadata* classes, for the sake of simplicity we decided to put *Metadata* as sub-class of *owl:Thing* instead of *Scope*. If it is required to allow multiple representations for a class of *Metadata*, a constraint can be added to the particular *Metadata* class denoting that it also has the superclass *Scope*.

Furthermore, it has to be noted here that we do not claim the set of proposed *Metadata* classes to be exhaustive. However, we have limited the description to these *Metadata* classes that are actually exploited in the further steps of the solution approach. In the following paragraphs, the association of those *Metadata* classes is described in more detail which cannot be expressed as a single value but have an internal structure and refer to *Scopes* of a *Representation* or concrete *Information Elements* of a certain *Scope*.

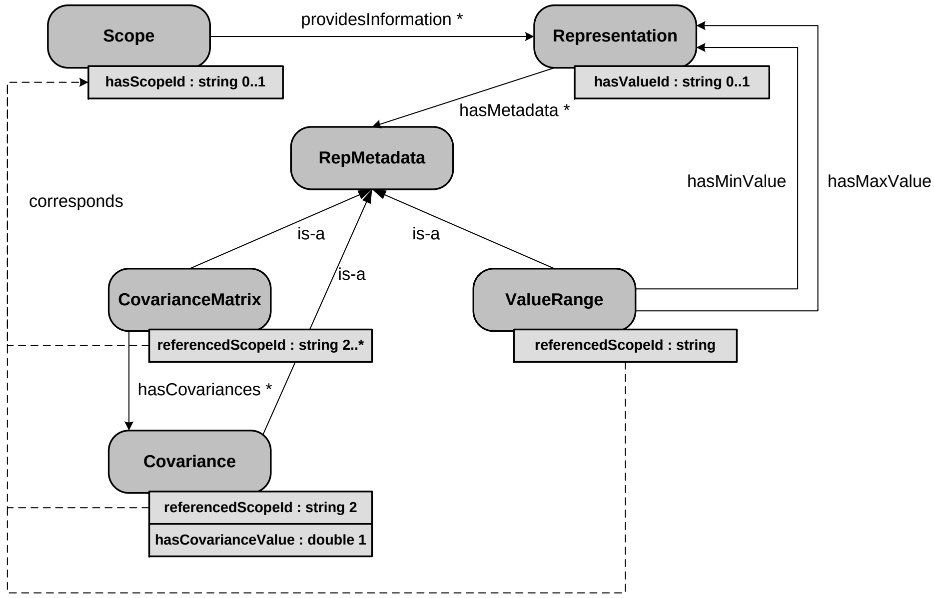

Figure 7.13: Association of Covariance Matrices and Value Ranges to Representations

Figure 7.13 illustrates the association of the measures for impreciseness, *CovarianceMatrices* and *ValueRanges*, to *Representations*. Here some difficulties arise as these meta-data have to refer to particular scopes of the corresponding representation. In order to avoid introducing many additional classes and properties, the links between the referred *Scopes* and the *Metadata* are simply established through two OWL datatype properties of type string: *hasScopeID* and *referencedScopeID*. The required matching of these strings is just assumed and not specified in OWL in full detail, in order to avoid unnecessary complexity in the definition of the ontology. This is reasonable as these definitions are not expected to be exploited by a general-purpose OWL reasoner but only by a dedicated module that performs the data conversion and fusion steps. It is important to note here that when specifying concrete values for the two data properties mentioned above, we are still at the class level, i.e. property restrictions are added as axioms for the class that associate a concrete individual (string) to the properties.

When specifying *ValueRanges* or *Variances*/*Covariances*/*CovarianceMatrices*, special care has to be taken that the corresponding definitions are reasonable. Actually, value ranges are feasible for all domains/sets where a partial or total order can be defined, and generalizations exist for the concepts of variance and covariance for symbolic data. In this thesis, however, we will restrict ourselves to the application of such concepts to discrete or continuous valued numerical *Representations*, such as integers (*IntegerRep*), floats (*FloatRep*), doubles (*DoubleRep*), etc.

The DST forms the central paradigm for competitive and complementary sensor fusion in this thesis. Thus, it is not only required to have elaborate support for the specification of belief assignments to single hypothesis but also to arbitrary sets of hypotheses. In the ontology, this is supported by the concepts shown in Figure 7.14. It is assumed that an Information Element coming as individual of a specific *Representation* corresponds to a single hypothesis. A *Scope* can now be associated with an arbitrary number of *BeliefAssignments*. A *BeliefAssignment* specifies exactly one *BeliefMass* and refers to a *ValueReference*, which

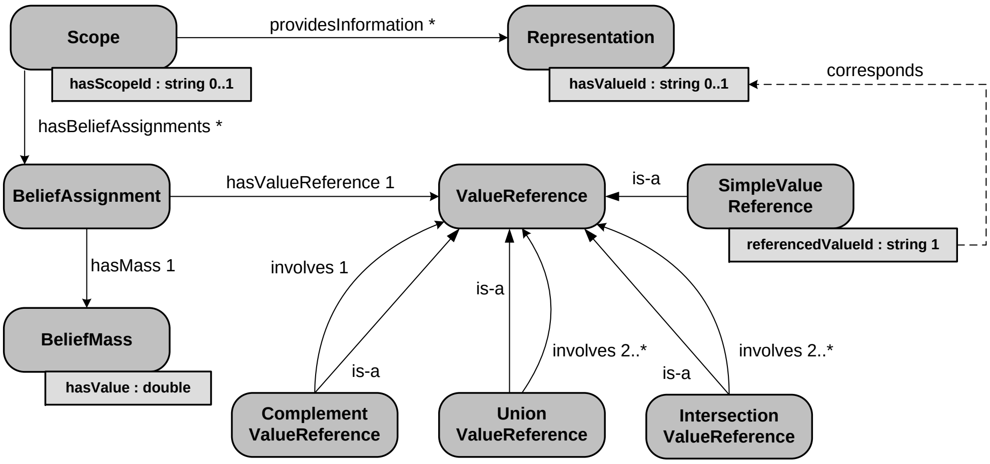

Figure 7.14: Association of Belief Assignments to Scopes

is used to specify the set of hypothesis the belief mass applies to. There exist four subclasses of *ValueReference*: *SimpleReference*, *ComplementReference*, *IntersectionReference*, and *UnionReference*. *SimpleReference* refers to an individual of an *Representation* corresponding to a concrete *Information Element* or hypothesis. This is realized with the help of two simple string datatype properties (*hasValueId*, *referencedValueId*) in a similar way as *Scopes* are referenced in *Metadata* associated to *Representations*. The difference is, however, that here concrete values cannot be specified at the class level, but can only be defined in a concrete individual of a *Scope*. *ComplementReference*, *IntersectionReference*, and *UnionReference* provide the corresponding set operations in order to allow compact specification of arbitrary sets of hypotheses.

7.2.6 Special Properties for Information Fusion with the DST

When it comes to information fusion with DST, the possibility for reasoning about the intersection and inclusion of hypotheses with regard to other hypotheses is a predominant requirement. In the case of a class/sub-class relationship, reasoning with regard to inclusion of the individuals of one class in the set of individuals of another class is inherently supported. However, in the general case, we have to take special care to indicate if a relationship expressed through a property defined in the ontology corresponds to either the intersection or the inclusion of hypotheses. For this purpose, we predefine the three properties ***DSTintersectedWith***, ***DSTincludes*** and its inverse property ***DSTisIncludedIn***. Properties like *partOf*, which allows to express that a room is a part of a floor, and a floor is a part of a building, are then defined as sub-properties of *DSTisIncludedIn*. This allows to infer that a belief mass for being in a room of a building contributes to the belief for being in the building.

7.2.7 Ontology Concepts for Inter-Representation Operations

The ASC model [135] not only envisages to define the internal structuring of information with regard to aspects and scales, it also introduces the concept of *Inter-Representation Operations* (IROs) allowing to convert one scale of a certain aspect to another scale. As this concept is vitally important to boost interoperability in dynamic and heterogeneous distributed environments, it is adopted in the context of this thesis. We align the concept of IROs to the scheme of *Entities* (*Entity Types*), *Scopes* and *Representations* and also discuss the more general case, where an IRO converting a piece of information from one *Representation* to another also depends on additional information.

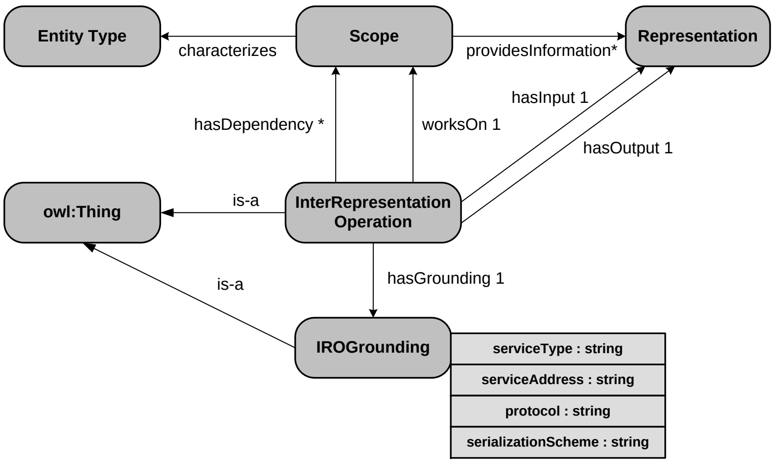

Figure 7.15: Definition of Inter-Representation Operations in the Ontology

The general purpose of IROs is to allow further processing of information, which already corresponds to the required *Scope* and describes the desired *Entity* but differs in the expected/required *Representation*. In order to allow spontaneous integration of heterogeneous information providers, mismatches in *Representations* have to be detected and resolved by applying appropriate IROs automatically and at runtime.

An IRO is defined as an operation that converts an *Information Element* of a specific *Scope* in a certain *Representation* and characterizing a specific *Entity* to the same *Information Element* in another *Representation* applicable for the corresponding *Scope*. In the simplest case, an IRO is just in charge of converting from one measuring unit to another, e.g. from *kByte* to *MByte* or from *Celsius* to *Fahrenheit*. Such simple conversions for measuring units can be derived from the information captured in the SWEET Units ontology and can be executed with the help of an accompanying set of rules by an SWRL engine. Furthermore, we also envisage more complex IROs in our model which also require some additional information apart from the current representation. Assume for example, we would like to convert the location of a user from coordinates relative to a certain building to global coordinates. In order to perform such a conversion, the position of the building in global coordinates is required as well. In general, this means that an IRO can have an arbitrary number of dependencies to additional information.

Figure 7.15 illustrates the ontology concepts for defining IROs. As an IRO converts an *Information Element* of a certain *Scope* from one *Representation* to another *Representation* of the same *Scope*, an IRO is directly associated to the particular *Scope* it works on. Via the *Scope* the IRO is also related to the characterized *Entity*. As an IRO is only in charge of converting the *Representation* of an *Information Element* to another *Representation*, the described *Scope* as well as the characterized *Entity* remain unchanged.

As already mentioned above, an IRO can have an arbitrary number of dependencies to additional information. Thus, the IRO is associated via the *hasDependency* property to an arbitrary number of *Scopes*, describing a certain *Entity* and providing information in a specified *Representation* again. Please note that these *Entities* not necessarily have to be the same as the *Entity* described by the *Scope* the IRO works on. In our previous example, the *Entity* whose location has to be converted is a user whereas the dependency refers to an *Entity* of type *Building*.

Furthermore, each IRO is associated to a grounding, which defines in detail how the IRO can be invoked. Thus, a grounding specifies the type of conversion service (e.g. SWRL, Java JAR, Spica Module, Web service), its address (e.g. IP address and port, URL), the communication protocol (e.g. HTTP, SOAP, RMI, SPICA Udp), and the serialization scheme (XML, JSON, SPICA binary format).

When specifying concrete IROs, i.e. individuals of the class *InterRepresentationOperation*, it has to be considered that an IRO is not applicable only for a single individual of a certain *Scope* and a single individual of the corresponding *Representation* but for all individuals of the *Scope* and all individuals of the *Representation*. Therefore, for a concrete IRO we only define an *object property assertion axiom* with regard to its *IROGrounding* but not with regard to the *Scope* and the *Representations* the IRO works on. The definitions of IRO individuals are exploited at runtime to reason about the availability of IROs that are required to convert between different *Representations* of *Information Elements*.

At this point, we would like to draw the reader's attention to a special kind of IROs: IROs that convert the *Information Elements* provided by a *Scope* to *SymbolicRepresentations*. Such IROs are applied, for example, to convert a *LocationInfo* provided in GPS coordinates to a symbol for a certain *Location Entity*. In this respect, following the approach presented above, correspondencies between symbols and data values are established through the invocation of appropriate IROs. This has turned out to be a highly flexible approach as the location and invocation details for the corresponding IROs are defined in the ontology and thus are also shared among the nodes involved in the computing infrastructure. This means that all the involved nodes not only share a common vocabulary of symbols but also employ the same operations to relate symbols to sensor data values. Here, another important aspect is the fact that IROs are able to create new symbols corresponding to new *Entities* (individuals of a certain *Entity Type*). This is the case, if an IRO is invoked that converts to a *SymbolicRepresentation*, but the corresponding symbol is not available in the ontology yet.

7.2.8 Structural Organization of the Ontology

Processing and reasoning with ontologies is known to be a resource consuming tasks, in particular if the ontologies become large and comprise a huge number of classes, individuals and properties. New lightweight representation schemes (e.g. the Manchester OWL syntax [61]) have been developed to speed up retrieving, parsing and preprocessing of ontologies. However, it is still vitally important to keep the ontology, or at least the active parts of the ontology, as small as possible. Thus, it has been established as common practice (see e.g. [145, 160]), to organize the ontology in a hierarchical manner incorporating at least two levels of hierarchy. Figure 7.16 shows how this approach is incorporated in and aligned to our Information Model.

In our approach, ontologies are organized in a three-level hierarchy. The top-level ontology comprises the general scheme of *Entity Types*, *Scopes* and *Representations* and includes all

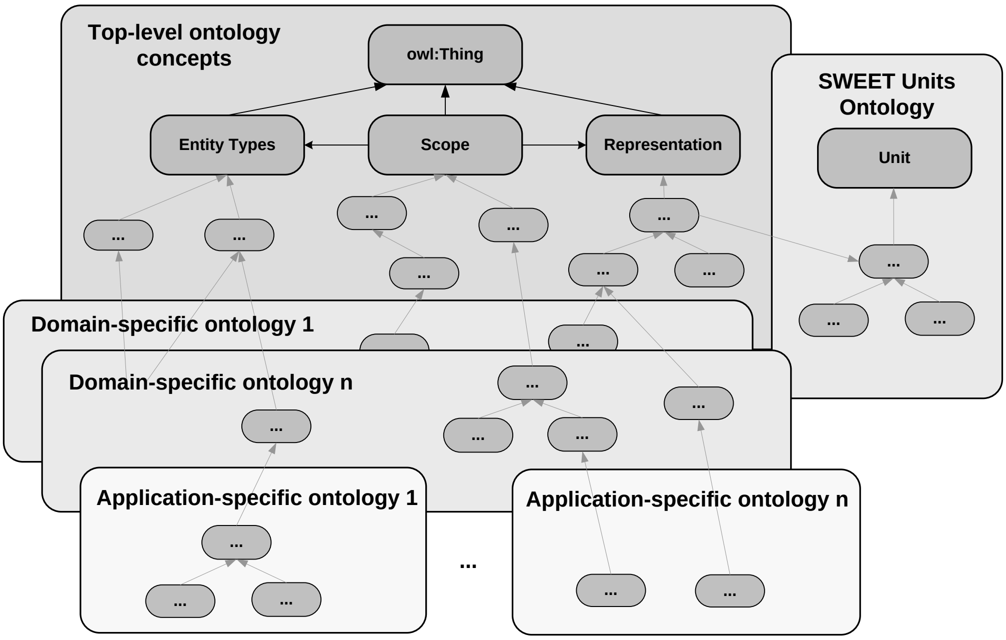

Figure 7.16: Proposed Organization of the Ontology

the basic ontology concepts described in the previous sections. Thus, the top-level ontology forms the baseline for other ontology definitions and ensures that the overall ontology adheres to the basic concepts presented in this thesis.

At the next level, the top-level ontology is extended by a number of domain-specific ontologies. These ontologies are intended to capture basic concepts and general knowledge applicable to almost all scenarios in a certain application domain. Here, with application domain we refer to whole areas like *Context-aware Applications in Ubiquitous Computing* or *Cooperating Teams of Autonomous Robots*. For almost all scenarios of context-aware applications, the notions of location, users, device resources, etc. are required. Consequently, these concepts are captured in a domain-specific ontology. Every domain-specific ontology is then extended at the lowest level by an application-specific ontology, which introduces the concepts that are likely to be important only for a limited number of scenarios in the corresponding domain. In the domain of context-aware computing, for example, one scenario could involve an application that supports the user while working in different office environments. Another scenario may comprise an application assisting the user in her traveling activities. This organizational structure of the ontology is complemented by the SWEET Units ontology. As definitions for units are expected to be important for a great number of applications, the SWEET Units ontology is considered to be a fundamental building block for ontologies as envisaged in this thesis.

In order to reduce computational effort, the structural organization of the ontology is exploited to keep only these ontologies active, which are really required. For example, in context-aware computing, only the top-level ontology, the SWEET Units ontology and the corresponding domain-specific ontology are necessarily required. Depending on the concrete application the corresponding application-specific ontology is plugged-in.

The ontology concepts described above serve a lot of different purposes. They are not expected to be handled by a single general-purpose ontology reasoner, as e.g. Fact++ [140] or Pellet [128]. Instead, it is proposed to also employ a number of dedicated reasoners which are tailored to different purposes of the ontology. In the following list, the usage of ontology reasoners as envisaged in this work is described:

- **General purpose ontology reasoners** (Fact++, Pellet) are used at design time and possibly at start-up of the system to **check consistency** of the set of ontology definitions. At runtime, the usage of such reasoners is restricted to **infer new knowledge through ontology reasoning** as described in Section 7.2.4. For this purpose, only the definitions for *Entity Types/Entities* are required and have to be activated.
- A **dedicated reasoner** is employed at design time (but maybe also at runtime) to transform the definitions of the ontology to SPICA ML models, for example, in order to support the **model-driven generation of data structures** and serialization/deserialization methods as required at the functional and exchange layer.
- At runtime, a **dedicated reasoning module** supports semantic **service discovery and matching**. It reasons about the required **mediation tasks** to extract specific information elements contained in other information elements. This module is also responsible for **reasoning about the necessity and availability of IROs**.
- Another **dedicated reasoning module** is in charge of processing the meta-data definitions and making the **meta-data associated to information elements** available for the information fusion steps.

From the list presented above it can be observed that apart from general consistency checking, general purpose reasoners are only envisaged to use the *Entity Type/Entities* definitions.[3] Thus, general purpose reasoners are used only for tasks they have already proven to show a reasonable performance in other projects. For all other aspects, we propose the usage of dedicated reasoners which are tailored for their specific task and only exploit a subset of defined classes, individuals and properties again.

7.3 Information Offers and Requests

One of the key challenges to deal with in today's and future distributed computing environment is the dynamic appearance of devices and services, which act as information providers and consumers. Here, the devices and services constituting the computing environment cannot be foreseen at design time. This implies the need for runtime service discovery and matching approaches in order to facilitate the establishment of communication links between information consumers and providers in a dynamic fashion. Apart from appropriate service discovery protocols, specification means are required to express information offers and requests. The corresponding language has to facilitate the filtering of inappropriate information providers and to establish only communication links that provide the information actually needed. In the following section, we present the *Information Offer and Request Language*, which is designed to meet these requirements. The proposed language can easily be integrated and be used on top of already existing service discovery protocols like UPnP/SSDP [51], SLP [54] or Geminga [7]. Afterwards, we present the corresponding

[3] This also implies a further splitting of the ontologies at the different levels.

matching approach, which is required to check the appropriateness of information offers with respect to information requests.

7.3.1 Information Offer and Request Language (IORL)

The *Information Offer and Request Language* (IORL) is based on the *Context Query Language* (CQL) [117] developed in the European project MUSIC [90]. Although CQL only focuses on the specification of queries for context information and does not deal with specification of information offers at all, it still forms a sound basis for our IORL. This is justified by the observation that a context query is generally used to restrict the requested context information to a subset of possibly provided context information by defining a set of constraints. The purpose of the specification of information offers is exactly the same: the set of offered information is restricted to a subset of the possibly offered information elements. Transferred to our ontology-based Information Model, this means the restriction of the set of possible individuals of *Scope* to a certain subset.

As the information offers and requests have to be tailored to the ontology-based Information Model, an obvious candidate for an underlying query language is SPARQL [111]. SPARQL is a W3C standard proposal for a RDF query language whose syntax is inspired by SQL. The approach is interesting as a way of incorporating semantic concepts and ontologies into a SQL-based query language. SPARQL allows the definition of constraints on *Entities*, *Scopes* and *Representations* and even the specification of fine-grained constraints on numerical values and strings, but the queries tend to become quite long and complex as they must be specified on the RDF triples. As the XML-based CQL is already tailored to our scheme of *Entities*, *Scopes* and *Representations* and allows to specify the corresponding queries in a more user-friendly compact manner, we have decided to use CQL as baseline for our IORL. For a more detailed discussion on the advantages of CQL with respect to SPARQL and other query languages we refer the reader to [117].

Just as the underlying CQL, the IORL is an XML-based language. Consequently, information offers and requests are specified in an XML document with root element *IORLSpecification*. Services can act as information providers and consumers at the same time (reasoners) and may provide and require different kinds of information. Thus, an *IORLSpecification* element comprises one *InformationOffers* element, which encapsulates an arbitrary number of *Information Offer*s and *Sub-Offer*s, and one *InformationRequests* element, which encapsulates an arbitrary number of *Information Request*s and *Sub-Request*s. The purpose of *Sub-Offer*s and *Sub-Request*s is to detail the specification with regard to *Scopes* nested in the main *Scope*. For example, if a *Scope UserInfo* provides in a certain *Representation* the *Scopes* (dimensions) *LocationInfo* and *ActivityInfo*, an *Information Offer* or *Request* would refer to the main *Scope UserInfo*, whereas the *Sub-Offers/Requests* can be used to detail the specification for *LocationInfo* and *ActivityInfo*. The general structure of an *IORLSpecification* is visualized in Figure 7.17.

The general structure of *InformationOffer* and *InformationRequest* elements is depicted in Figure 7.18. *Information Offer* elements differ from *Information Request* elements only through some additional attributes, which are only available in *Information Offers* (*Source*, *SourceType*) or in *Characterized Entity* elements (*Negotiable*) of *Information Offers* (marked with gray caption).

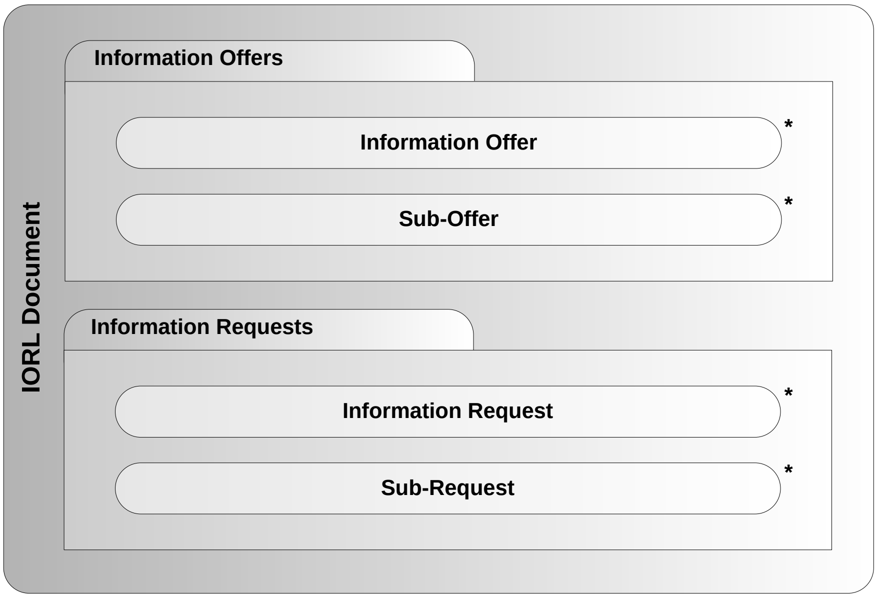

Figure 7.17: General Structure of an IORL Document

The attributes *Scope* and *Representation* of type *xsd:string* refer to the corresponding concepts defined in the ontology and depict the type of the provided information and its *Representation*. *Subscription* is also an attribute of type *xsd:string* and depicts the provided or requested subscription mode. It can take the values '*ONCLOCK*', '*ONCHANGE*', '*ONDEMAND*' and a concatenation of these values separated with '/'. '*ONCLOCK*' defines a subscription offer or request for notification in time intervals corresponding to the reciprocal value of the attribute *Frequency*. '*ONCHANGE*' refers to a notification offer/request for each value change event. '*ONDEMAND*' specifies that the information is exchanged synchronously only as an answer to a corresponding request. The attributes *Source* and *SourceType*, which are only used for *Information Offers*, refer to the ontology again and define the *Entity* and the corresponding *Entity Type* for the information source, i.e. the service.

InformationOffer and *InformationRequest* elements include an arbitrary number of *CharacterizedEntity* elements. These elements are used to define and restrict the set of characterized *Entities* of the provided *Scope*. They have the attribute *Entity*, which is of type *xsd:string* and refers to individuals of a certain *Entity Type*. If all individuals of a certain *Entity Type* should be considered, simply a reference to the *Entity Type* is given. For referring to single individuals there are two different possibilities: A string containing a reference to a concrete individual defined in the ontology, and a string that is a concatenation of a reference to the *Entity Type* defined in the ontology, a separator '|' and the *Symbolic Representation* of the individual. An example for the second alternative is '*...#Person|Peter_Maier*'. The rationale behind the second alternative is that we cannot expect, for example, that all possible persons or users are defined as concrete individuals in the ontology. Thus, it is required to define individuals 'on the fly' by providing the *Entity Type* and the *Symbolic Representation* of the individual.

The boolean attribute *Recursive* defines that the entity characterization applies also to all nested *Scopes* and thus allows to compact the specification. The attribute *Negotiable* is of type *xsd:boolean* and is only available for *Information Offers*. It depicts that the defined

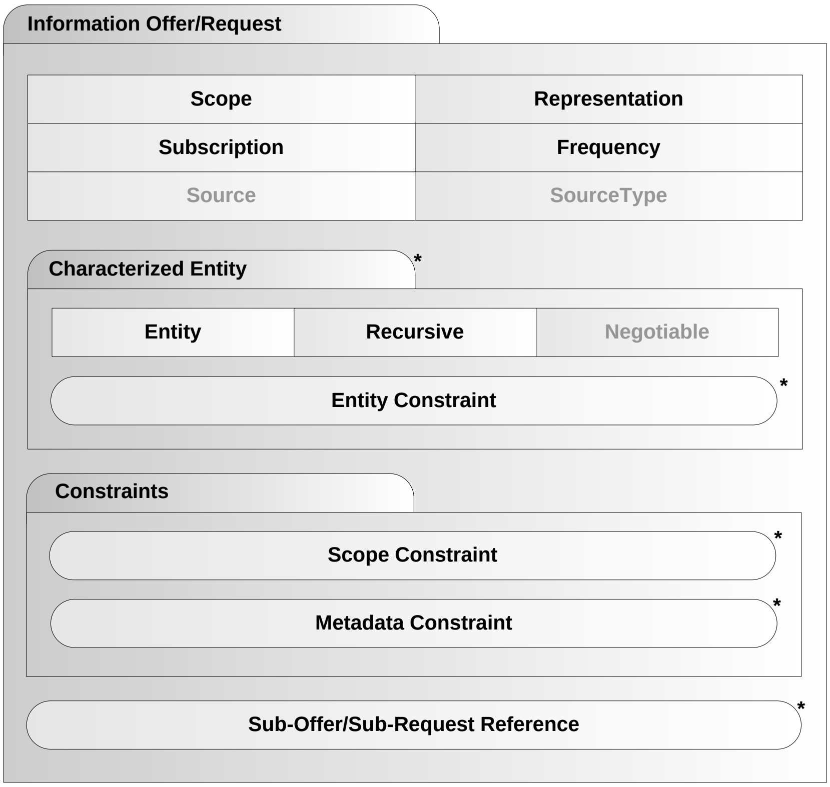

Figure 7.18: Structure of IORL Information Offers and Requests

set of *Entities* is negotiable. For example, for a service that provides *LocationInfo* for a large number of persons it may be difficult to precisely define the set of persons for which information can be provided. In this case, we envisage a kind of negotiation procedure in our future work that is similar to what is practiced for service level negotiation. First, the service announces that it provides *LocationInfo* for individuals of entity type *Person* and marks this characterization as negotiable. If then an information consumer is interested in *LocationInfo* for only a limited set of persons, it matches the information offer but then asks the information provider about the concrete set of persons it is really interested in. Of course, it also has to be investigated in future how this kind of negotiation can be incorporated with the underlying service discovery protocols.

Entity Constraints are used to constrain the set of *Entities* with regard to the relationships they participate in. For example, if the set of characterized persons should be constrained to a set of persons currently located in a building, a simple *Entity Constraint* element is defined through the attributes '*Relation* = '...*#locatedIn*" (defined in the ontology) and '*DomainEntity* = '*University_of_Kassel*" (defined in the ontology).

Scope Constraint elements (see Figure 7.19) are used to constrain the possible values of a *Scope* involved in the main *Scope*. Here, only constraints on *Scopes* in a *BasicRepresentation* or *SymbolicRepresentation* are supported; definition of constraints for *Scopes* in a *CompositeRepresentation* has to be realized in a corresponding *Sub-Offer* or *Sub-Request*. A *Scope Constraint* defines the *Scope* which should be constrained through a *ScopeProperty* or *ScopeID* attribute. The values of these attributes are references to the corresponding property (sub-property of *hasDimension*) or identifier (*scopeID*) of the *Scope* defined with the

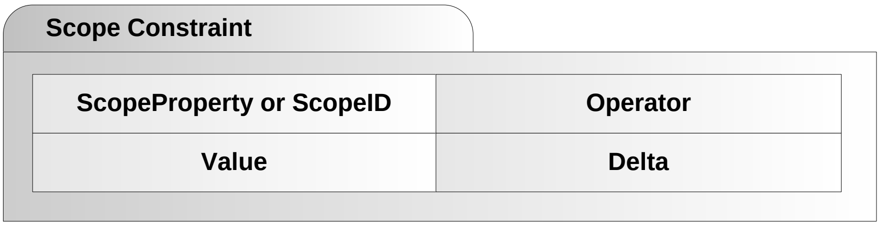

Figure 7.19: Structure of an IORL Scope Constraint

Representation of the main *Scope* in the ontology. The *Operator* attribute of type *xsd:string* defines the operator for the constraint and can take the values '*EQ*', '*NEQ*', '*GT*', '*GE*', '*LT*', '*LE*', '*CONTAINS*', and '*NOTCONTAINS*'. The latter two can be used for constraining string values. The *Delta* attribute allows to define that a comparison to equality or inequality has not to be precise and a tolerance of the value of *Delta* is acceptable.

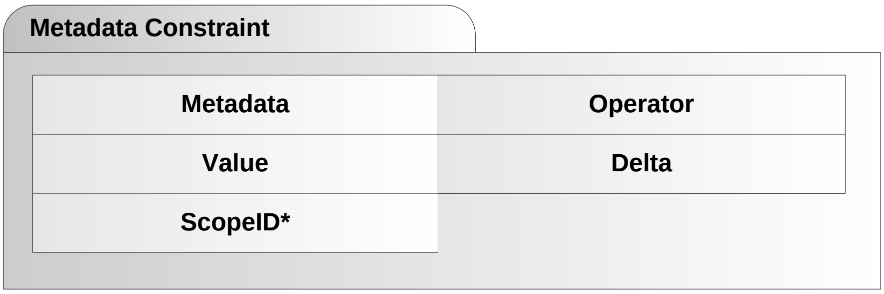

Figure 7.20: Structure of an IORL Metadata Constraint

Metadata Constraint elements (see Figure 7.20) are very similar to *Scope Constraint* elements. However, instead of referring to a property or identifier, the corresponding *Metadata* class defined in the ontology is referenced. Additionally, *Metadata Constraints* may refer to a number of *ScopeIDs*. This is required, for example, if a *Covariance* should be constrained. Here, it is not sufficient just to provide a reference to the class defined in the ontology, but we need also to know the two *Scopes* the *Covariance* relates to. Besides, we also envisage the combination of *Entity Constraints*, *Scope Constraints* and *Metadata Constraints* through the logical operators '*AND*', '*OR*' and '*NOT*' to logical expressions.[4]

The elements used to refer to the *Sub-Offer*s or *Sub-Request*s only have two attributes. They identify the restricted *Scope* (*ScopeProperty* or *ScopeID*) and provide the reference to the corresponding *Sub-Offer*/*Sub-Request* (*RefID*).

Sub-Offer and *Sub-Request* elements have nearly the same structure as *Information Offer/Request* elements. The only difference is that they do provide one additional attribute (*ID*), which is used to establish the references.

It is noteworthy here that if compared to CQL, the structure and some attributes of a request have been changed slightly. However, the way of referring to individuals of *Entity Types*, as well as the way of defining constraints for *Scopes* and *Metadata* and the support for their logical combination have been directly adopted for the IORL.

[4]For the sake of simplicity this is not depicted in Figure 7.18.

7.3.2 Matching of Information Offers and Requests

In general, for an information offer to match an information request with regard to the *Scopes* and *Representations* the following conditions must hold:

1. The requested *Scope* a) matches exactly the provided *Scope*, b) is a generalization of the provided *Scope* or c) corresponds to a nested *Scope* of the provided *Scope*. In the last case, the matching procedure is repeated with the corresponding *Sub-Offer*.
2. The requested *Representation* a) directly matches the provided *Representation* of the *Scope* (or nested *Scope*), b) is a generalization of the provided *Representation* or c) appropriate IROs are available to resolve mismatches.

With regard to the *Subscription* mode and *Frequency*, the following conditions must hold:

1. At least one of the requested *Subscription* modes has to be provided by the offering service.
2. In the case of a notification in regular time intervals ('*ONCLOCK*') the requested *Frequency* has to fall into the interval $\left[(1/\alpha) * f_{provided}, \alpha * f_{provided}\right]$ with e.g. $\alpha = 1.5$.

These conditions can be checked quite easily and can be used to pre-filter inappropriate information offers. However, the matching of *Entity Constraints*, *Scope Constraints* and *Metadata Constraints* is more complex, in particular if it is considered that the constraints can be combined to complex logical expressions. In the previous section, it was explained that both, *Information Offers* and *Requests*, define subsets of possible *Scope* individuals. From this observation it can easily be concluded that an information offer matches a request if the intersection of the two subsets is not empty. In this case, at least part of the information provided by a service corresponds to the requested information. From a theoretical point of view, a non-empty set of individuals for the intersection can be achieved if the conjunction of the constraints defined for the *Information Offer* and the constraints defined for the *Information Request* is satisfiable.

For checking the satisfiability of the conjunction of the constraints on *Information Offers* and *Requests*, for example, an algorithm based on the method of semantic (or analytic) tableaux [10] can be applied. The general idea of such an algorithm is to attempt to break complex formulas into smaller ones. For this purpose, a tree structure is successively created by applying a set of expansion rules until contradictions become directly evident or no further expansion rule can be applied. If all paths from the root of the tree to the leaves include contradictions, i.e. all branches of the tree are closed, the conjunction of the constraints is unsatisfiable. With regard to *Information Offers* and *Requests* this means that we assume matching if no closed tableaux (tree with all branches closed) for the conjunction of the constraints on the offer and the request can be found.

When applying our matching approach, two further aspects are important to consider:

- When searching for contradictions, the constraints on *Entities* have also to be checked against the constraints on *Scopes*. In Section 7.2.4, we have seen how relationships between *Entities* are related to information provided by *Scopes*.
- For the checking of constraints on *Scopes* it may already be necessary to apply the corresponding IROs that are used to resolve the mismatches with regard to the representations.

It is also noteworthy here that the checking of satisfiability of logical expressions is known to be a $\mathcal{NP}$-complete problem. Thus, special care has to be taken that the logical expressions do not become too complex in order to achieve a feasible reasoning time.

8 Inter-Representation Operations

The general purpose of *Inter-Representation Operations* (IROs) is to allow further processing of information which already corresponds to the required *Scope* and describes the desired *Entities* but differs in the expected/required *Representation*. This is vitally important to boost interoperability in dynamic and heterogeneous distributed environments as it has to be expected that independent developers use different *Representations* for the same piece of information.

8.1 Characteristics and Challenges

An IRO is defined as an operation that converts an *Information Element* of a specific *Scope* in a certain *Representation* to the same *Information Element* in another *Representation* applicable for the corresponding *Scope*. The operation is provided through a method in a library (e.g. Java JAR or a C# Dll), a dynamically accessible service available in the computing environment (e.g. a Web service or SPICA module) or through a set of SWRL rules accompanying the ontology.

From a conceptual point of view, a *Representation* is a collection of atomic *Information Values* in primitive data types like integers, floats, strings, etc. *Information Elements* are only used to group the *Information Values* (and maybe other *Information Elements*) into logical units forming *Representations* for more complex *Scopes*. Consequently, from a mathematical point of view an IRO realizes a (partial) mapping:

$$\text{IRO} : \mathcal{D}_1^p \times \mathcal{D}_2^p \times \ldots \times \mathcal{D}_m^p \longmapsto \mathcal{D}_1^r \times \mathcal{D}_2^r \times \ldots \times \mathcal{D}_n^r \tag{8.1}$$

where $m \geq 1$ and $n \geq 1$. $\mathcal{D}_i^p$ denotes the domain of the i-th dimension (*Information Value*) of the provided *Representation*, and $\mathcal{D}_j^r$ denotes the domain of the j-th dimension (*Information Value*) of the requested *Representation*. $\mathcal{D}_i^p$ and $\mathcal{D}_j^r$ can be defined as $\mathbb{R}$, $\mathbb{Q}$, $\mathbb{Z}$, $\mathbb{N}$ (and subsets, intervals, enumerations of these) or as (sub-)set of all possible strings, corresponding to the primitive data type (double, float, integer, short, uint8, bool, string, etc.) of the respective *Information Value* and its possible/allowed values.

In the simplest case, an IRO is just in charge of converting from one measuring unit to another, e.g. from *kByte* to *MByte* or from *Celsius* to *Fahrenheit*. In this case, m and n equal to 1, which means that a data structure comprising only one dimension is transformed into a data structure which also contains only a single dimension. m and n are greater than 1, for example, if polar coordinates are converted to Cartesian coordinates. Furthermore, we also allow for more complex IROs in our model which require some additional information apart from the current *Representation*. Assume, for example, that we would like to convert the location of a user from coordinates relative to a certain building to global coordinates.

In order to perform such a conversion, the position of the building in global coordinates is required as well. In this case, only the dimensions corresponding to the domains $\mathcal{D}^p_1$ to $\mathcal{D}^p_{m1}$ $(1 \leq m_1 < m)$ are provided by the current *Representation* (relative coordinates). The dimensions corresponding to the domains $\mathcal{D}^p_{m1+1}$ to $\mathcal{D}^p_m$ result from the additionally required *Information Element* in the respective *Representation* (global coordinates of the building). Special attention has also to be paid to IROs which establish a mapping of values measurable by sensors to symbols (see also Section 7.2.3 and Section 7.2.7), which means that the IROs convert information provided in a certain *Representation* into a *Symbolic Representation* of a certain *Entity*. In this case, a data structure comprising a number of dimensions is typically transformed to a simple string, i.e. $m > 1$ and $n = 1$.

The concept of IROs as envisaged in this thesis imposes a number of challenges to be addressed. For example, appropriate runtime reasoning mechanisms are required for determining and selecting appropriate IROs that are able to bridge the mismatches in representations. These reasoning mechanisms have also to be aware of possible dependencies of the IROs to additional information. Hence, they have to be integrated well with the discovery and matching approaches for information offers and requests. Furthermore, the IROs have to be located and invoked at runtime. In particular, this means that the availability of the corresponding libraries and services has to be checked, libraries have to be fetched and loaded, services have to be bound and invoked or a SWRL engine has to be employed.

A major challenge, however, results from the general requirement of this thesis to be able to deal with uncertain, imprecise, unreliable and heterogeneous sensor information. This implies that not only the information itself has to be converted but also the measures for uncertainty, impreciseness and unreliability have to be preserved across the transformations. As we employ Dempster-Shafer belief functions to express and reason about uncertainty, impreciseness and unreliability, a corresponding transformation of the belief functions is required as well. In our case, we consider the transformation of basic belief assignments, which are in one-to-one correspondence with belief functions.

Let $\Omega^p = \mathcal{D}^p_1 \times \ldots \times \mathcal{D}^p_m$ and $\Omega^r = \mathcal{D}^r_1 \times \ldots \times \mathcal{D}^r_n$, then the basic belief assignment $m^p : 2^{\Omega^p} \mapsto [0,1]$ of the information in the provided *Representation* has to be converted to a corresponding basic belief assignment $m^r : 2^{\Omega^r} \mapsto [0,1]$ of the information in the requested *Representation*, according to the transformation realized by the IRO.

Here, the belief functions of the additional dependencies of the IROs also have to be considered, which introduces further complexity. Developers should not be bothered with providing support for the transformation of belief functions. They should only be required to specify a 'point-to-point' transformation encapsulated in a simple method, and the underlying framework automatically realizes the corresponding transformation of the belief functions.

From the examples above, it is quite obvious that IROs can result in highly non-linear transformations. Furthermore, if the IROs convert between string representations or establish a mapping between symbols and sensor values, it is even very unlikely that the transformation has mathematical properties like continuity, monotonicity, etc.

Although all challenges described above are very important, in the following sections we focus on concepts for the transformation of the basic belief assignments as it is considered to be most difficult to realize.

8.2 Transforming Measures for Impreciseness and Uncertainty

In our approach, impreciseness and uncertainty are modeled through Dempster-Shafer belief functions, or more precisely, basic belief assignments (BBAs), which are in one-to-one correspondence with belief functions. Before explaining how the BBA of the information in the provided *Representation* is transformed into a BBA of the information in the requested *Representation*, we first have to recall how BBAs are described using the Information Model introduced in Chapter 7.

8.2.1 Basic Belief Assignments and the Information Model

In the previous section, it was stated that a BBA m^p of the information in the provided *Representation* realizes a mapping $m^p : 2^{\Omega^p} \mapsto [0,1]$ with $\Omega^p = \mathcal{D}_1^p \times \ldots \times \mathcal{D}_m^p$, $m^p(\emptyset) = 0$ and $\sum_{A \subseteq \Omega^p} m^p(A) = 1$. As the power set of Ω^p has to be expected to comprise a very large number (or even an infinite number) of elements, in general it is not feasible to define the mapping by enumerating all elements of 2^{Ω^p} and assigning a mass value to them. Instead, usually only the assignments for the *focal elements* of 2^{Ω^p}, i.e. the elements A in 2^{Ω^p} with $m(A) > 0$, are provided, and functions and patterns are used to compact the specification if possible.

With regard to the Information Model we facilitate a compact specification of BBAs by using what we call basic hypotheses and complex hypotheses. Basic hypotheses correspond to a certain Information Element having associated a belief mass and possibly other meta-data. Thus, basic hypotheses are given as individuals of the provided *Representation* that are related to the corresponding *Scope* individual by the *providesInformation* property. Complex hypotheses refer to basic hypotheses and combine them by using the logical operators '*and*' (*IntersectionValueReference*), '*or*' (*UnionValueReference*), and '*not*' (*ComplementValueReference*) as described in Section 7.2.5. Consequently, complex hypotheses and the corresponding belief assignment are specified as part of the *Scope* individual.

The Information Model provides the following modelling elements for the specification of basic hypotheses:

- Masses are assigned to singletons (elements of Ω^p) by simply providing values for all the scope dimensions of the Representation and associating a belief mass to the individual (basic hypothesis).
- Variances and covariance matrices are used for numerical dimensions to indicate that the values follow a normal distribution with the provided variance or covariance matrix, where the mean is given through the values provided for the scope dimensions. Masses are assigned only to singletons of Ω^p and are calculated with the corresponding Gaussian function. If additionally a belief mass is assigned as meta-data, this belief mass acts as scaling factor for the Gaussian function.
- Uniform value ranges are used for numerical dimensions to indicate that the values are uniformly distributed within this value range. Masses are assigned only to singletons within the value range, are constant and sum up to 1 or to the value which is associated through the belief mass meta-data attribute. It is also noteworthy here that multi-dimensional uniform value ranges can be specified by associating uniform value range meta-data to several scope dimensions.

- Value range meta-data specify the assignment of a belief mass to the whole value range, i.e. to the union set of all singletons of Ω^p within the value range. This has to be seen in contrast to the uniform value range where it is additionally assumed that the complete mass is equally distributed among the singletons, which results in a mass assignment for the singletons itself. In the same way as for uniform value ranges, multi-dimensional value ranges can be modeled by associating value range meta-data to several scope dimensions.

In order to fully understand the interpretation of BBA specifications with the help of basic and complex hypotheses, the reader has to become aware of the following details:

- Basic hypotheses are not exclusive, i.e. two different basic hypotheses can assign different belief masses to the same singleton of Ω^p. When calculating the BBA, it is assumed that all the belief masses assigned to a particular singleton are summed up.
- Whereas the meaning of the logical operator '*not*' is obvious for singletons and value ranges, the situation is more difficult for basic hypotheses that are specified with the help of the primitives uniform value range and mean/covariance matrix. Such basic hypotheses specify a kind of mass density for a set of singletons, for which the normal negation of a body of evidence (see Equation 9.13) lacks a meaningful interpretation. Therefore, we introduce the convention that '*not*' assigns the belief mass to the set of all singletons which are outside of the uniform value range or whose Mahalanobis distance from the mean (see also Section 9.2.3.2) is greater than a certain threshold. At this point, it has to be noted that the simplification rule *Rule 6* we will derive in Section 9.2.3.1 is compliant with the normal negation of a body of evidence and also with our modified interpretation.
- A probability assignment for singletons $a \in \Omega^p$ differs from a mass assignment in the usual sense, as it implicitly states that $m(A) = 0$ for all $A \subseteq \Omega^p$, $|A| > 1$. In particular this means that $m(\Omega^p) = 0$. Of course, it has to be ensured that $\sum_{a \in \Omega^p} m(a) = 1$.
- Belief masses can also be assigned to complex hypotheses referring to intersections and unions of basic hypotheses which include the meta-data variance, covariance matrix or uniform value range. As already stated above, such basic hypotheses provide a concrete mass assignment to all singletons or a subset of the singletons. In this case, these concrete mass assignments (without scaling according to the belief mass assigned to the basic hypothesis) form conditional mass assignments $m(A|H)$ with $A \subseteq \Omega^p$ and H denoting the corresponding basic hypothesis. A mass assignment, for example, to the union of two basic hypotheses H_1 and H_2 then represents an assignment for $m(H_1 \cup H_2)$, and the corresponding conditional mass assignment $m(A|H_1 \cup H_2)$ is derived from $m(A|H_1)$ and $m(A|H_2)$ using the *Disjunctive Rule of Combination* (see Equation 4.17).

In summary, a BBA is defined with regard to the Information Model as

$$m(A) = \sum_{H \in \mathcal{H}} m(A|H) \cdot m(H) \quad A \subseteq \Omega^p \tag{8.2}$$

H denotes a basic or complex hypothesis, $\mathcal{H}$ is the set of hypotheses (basic and complex), and $m(A|H)$ describes the conditional mass assignment with regard to the hypothesis H. It is also important to note that we assume an implicit hypothesis H_{Ω^p} such that $m(\Omega^p|H_{\Omega^p}) = 1$

and $m(H_{\Omega^p}) = 1 - \sum_{H \in \mathcal{H}, H \neq H_{\Omega^p}} m(H)$. Thus, the mass assignments to basic and complex hypotheses do not have to sum up to 1 but just to a value less than 1. The remaining mass is assigned to the whole frame of discernment, the belief mass of which represents the amount of total ignorance.

8.2.2 Transformation of Complex Hypotheses

The transformation of the belief assignments specified through complex hypotheses imposes two important challenges. On the one hand, complex hypotheses define mass assignments to possibly very complex sets of singletons. Consider for example, complex hypotheses referring to nested unions and intersections of basic hypotheses whose impreciseness is specified through covariance matrices and uniform value ranges. In particular, if we have multiple dimensions, the resulting set of singletons may form complex geometrical shapes which may be hard to determine, to approximate and to represent. On the other hand, mass assignments resulting from the conditional BBAs defined by the basic hypotheses and the mass assignment to the hypotheses (basic and complex) have to be maintained as well. This requires that the transformed complex hypotheses appropriately relate to the corresponding transformed basic hypotheses.

Considering these aspects, an obvious idea is just to transform the basic hypotheses and to maintain the complex hypotheses and the corresponding belief assignment unchanged. As already mentioned above, however, the transformations cannot be expected to show mathematical properties like continuity or monotonicity. Even more important, they cannot be assumed to realize injective functions. Imagine, for example, a function that converts location information of a user in GPS coordinates into a representation that only provides the name of the city the user is currently visiting. In this case, a huge (infinite) number of GPS coordinates will be mapped to a single city. This causes problems again as only injective functions have the mathematical property $f(A \cap B) = f(A) \cap f(B)$. It is also obvious that for a non-injective function it cannot be guaranteed that $f(\overline{A}) = \overline{f(A)}$.

So far, we have not found a feasible solution for the transformation of arbitrarily complex hypotheses yet. Thus, we restrict our transformation support to complex hypotheses which assign belief masses only to unions of basic hypotheses and apply the approach outlined above (transforming the basic hypotheses and retaining the complex hypotheses and its belief assignments). The belief masses assigned to other complex hypotheses are simply assigned to Ω^r, which means that these beliefs turn into ignorance. Theoretically, we can avoid this situation as all sets of singletons can be represented as unions of sets of singletons. In our future work, we will further investigate the issue and try to find a feasible approach for the transformation of arbitrarily complex hypotheses.

Another difficulty arises as IROs may have a dependency to additionally required information.[1] With regard to the transformation of the measures for impreciseness and uncertainty this means that also the belief function defined for the additional information has to be taken into account. More precisely speaking, we have two BBAs: $m_1 : 2^{\mathcal{D}_1^p \times \ldots \times \mathcal{D}_{m1}^p} \mapsto [0,1]$ for the actual Information Element to be transformed, and $m_2 : 2^{\mathcal{D}_{m1+1}^p \times \ldots \times \mathcal{D}_m^p} \mapsto [0,1]$ for the additional dependency. Both, m_1 and m_2, are specified using basic and complex hypotheses as explained above.

[1]Without loss of generality, we assume here only one additional dependency. The general approach is also applicable for an arbitrary number of additional dependencies.

The general idea of our solution approach is to convert the two BBAs into a single BBA and to apply the approach already described above for IROs without additional dependencies. For this purpose, the following steps have to be performed:

1. So far, we can consider only complex hypotheses which are unions of basic hypotheses. Thus, all mass assignments to complex hypotheses involving intersections and negations are shifted to Ω^1 (frame of discernment of m_1) and to Ω^2 (frame of discernment of m_2) respectively. As we implicitly assume the hypotheses H_{Ω^1} and H_{Ω^2} which represent the whole frames of discernment, this can be realized by simply discarding the corresponding complex hypotheses and their mass assignments.
2. Now we combine each remaining hypothesis H_i^1 (basic or complex) of m_1 with each remaining hypothesis H_j^2 (basic or complex) of m_2 to a hypothesis H_{ij}^p of the common frame of discernment Ω^p and assign the mass $m_1(H_i^1) \cdot m_2(H_j^2)$ to it. This nearly corresponds to the *vacuous extension* [132] of m_1 and m_2 to the common frame of discernment and to the application of *Dempster's Rule of Combination* at the hypotheses level. The only difference is that complex hypotheses H_{ij}^p involving the implicit hypotheses H_{Ω^1} and/or H_{Ω^2} are neglected. However, this is reasonable as such hypotheses where either the actual information or the additionally required information is modelled as totally unknown, cannot be transformed anyway, and thus the result of the transformation is unknown as well. The combination of two hypotheses is given as:

$$
\begin{gathered}
H_i^1 = (h_1^i \cup \ldots \cup h_{n1}^i) \quad H_j^2 = (h_1^j \cup \ldots \cup h_{n2}^j) \quad \rightarrow \\
H_{ij}^p = ((h_1^i, h_1^j) \cup (h_1^i, h_2^j) \cup \ldots \cup (h_{n1}^i, h_{n2-1}^j) \cup (h_{n1}^i, h_{n2}^j))
\end{gathered}
$$

   where the h^i are the basic hypotheses of H_i^1 and the h^j the basic hypotheses of H_j^2. If H_i^1 and/or H_j^2 denote basic hypotheses, then $n1$ and/or $n2$ equal to 1.
3. The tuples (h_a^i, h_b^j) are now considered as basic hypotheses of the extended *Scope* individual.

The number of basic hypotheses to be transformed corresponds to the product of the number of basic hypotheses of the actual *Scope* individual and the numbers of basic hypotheses of the additionally required *Scope* individuals. This can result in quite high numbers of basic hypotheses even if the numbers of basic hypotheses for the single *Scope* individuals are small. Thus, special care has to be taken that the transformation of the basic hypotheses can be achieved in a reasonable time frame.

8.2.3 Transformation of Basic Hypotheses

As basis for the following description we would like to call the reader's attention to the following characteristics and challenges for the transformation of basic hypotheses as envisaged in this thesis:

- In general, a developer should only be required to specify a point-to-point transformation and should be relieved from the burden to also provide the transformation for the BBAs. As a consequence, the transformation of the BBAs, and in particular the transformation of the basic hypotheses, is realized with the help of **sampling techniques**.

- The transformation can realize an arbitrary mapping. Thus, the transformation cannot be expected to have mathematical properties like continuity, monotonicity, differentiability or to be an injective, surjective or bijective function.
- The granularities of Ω^p and Ω^r can differ considerably. With different granularities we refer to the fact that a potentially high (or even infinite) number of elements of Ω^p can correspond to only a small number of elements or even only a single element of Ω^r. It may also be the case that one element in Ω^p corresponds to a potentially high (or even infinite) number of elements in Ω^r. Consider, for example, a transformation of location information of a user from the address of a building to GPS coordinates.
- It was already stated above that we allow value ranges, variances and covariance matrices only for numerical scope dimensions. However, different types of these meta-data can be used in a single basic hypothesis. This is particularly the case if the IROs have dependencies to additionally required information.

In this thesis, we focus on the transformation of BBAs involving (uniform) value ranges, variances and covariance matrices for IROs that have the mathematical properties of continuity and infinite differentiability for all involved scope dimensions. The corresponding approaches are described in the following paragraphs. Afterwards, we also discuss some possibilities to realize the transformation of the BBAs for other cases.

8.2.3.1 Transformation of (Uniform) Value Ranges

Although we assume that the transformation is a mathematical function that is continuous and infinitely differentiable, we still have to cope with the problem that the function may realize a multi-dimensional mapping, i.e. (without loss of generality) $f : \mathbb{R}^m \mapsto \mathbb{R}^n$, and has to be expected to be highly non-linear.

A common approach, for example, to propagate normal distributions through non-linear functions is to use the first- or second-order Taylor approximation of the transformation function [125]. As first-order approximations have proved to show quite poor performance in a number of applications [125], we discuss the usage of second-order Taylor approximations of the transformation function in our approach. For the i-th dimension of the function f, $f_i : \mathbb{R}^m \mapsto \mathbb{R}$, it is given as:

$$f_i(\vec{x}) \approx f_i(\vec{x_0}) + (\vec{x} - \vec{x_0})^T \nabla f_i(\vec{x_0}) + \frac{1}{2}(\vec{x} - \vec{x_0})^T H_{f_i}(\vec{x_0})(\vec{x} - \vec{x_0}) \tag{8.3}$$

where $\nabla f_i(\vec{x_0})$ denotes the gradient of f_i at $\vec{x_0}$ and $H_{f_i}(\vec{x_0})$ denotes the Hessian matrix[2] of f_i at $\vec{x_0}$. Now the problem arises that at runtime the transformation function is not available in an appropriate representation to derive the gradient and the Hessian matrix analytically. Instead, we have only the possibility to perform point-to-point transformations. In order to approximate the gradient and the Hessian matrix, approaches are applied which are known from the area of image processing for estimating the first-order and second-order partial derivatives of the discretized signal functions.

[2]The Hessian matrix is the matrix of the second partial derivatives of a function.

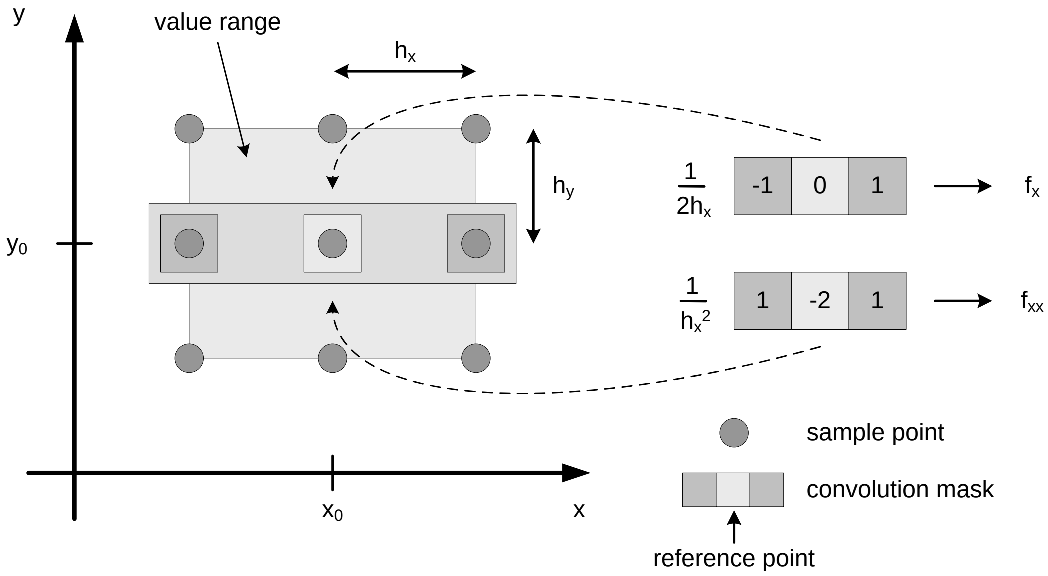

Figure 8.1: Estimation of Partial Derivatives of Discretized Functions

In image processing it is common to estimate the first-order and second-order partial derivatives of the image function with the help of simple convolution masks.[3] The general approach along with the corresponding convolution masks and their application to a discretized function in two variables is shown in Figure 8.1. Using this approach, we can estimate the first-order partial derivative f_x and the second-order partial derivative f_{xx} for a function $f : \mathbb{R}^2 \mapsto \mathbb{R}$ as:

$$f_x(x_0, y_0) \approx \frac{1}{2h_x}\left(f(x_0 + h_x, y_0) - f(x_0 - h_x, y_0)\right) \tag{8.4}$$

$$f_{xx}(x_0, y_0) \approx \frac{1}{h_x^2}\left(f(x_0 + h_x, y_0) - 2 \cdot f(x_0, y_0) + f(x_0 - h_x, y_0)\right) \tag{8.5}$$

Analogously, we estimate the first-order and second-order partial derivatives of our transformation function within a certain value range by transforming a number of sample points and then applying the convolution masks. Second-order derivatives like $f_{xy}(x, y) = \frac{\partial}{\partial y}\frac{\partial}{\partial x} f(x, y)$ are estimated by successively applying the convolution masks first in x-direction and afterwards in y-direction. For a function in two variables we need to transform 9 sample points (as depicted in Figure 8.1), and for a function in three variables 19 sample points have to be transformed. Generally speaking, $1 + 2m + 2m(m - 1)$ sample points are required for a mapping from $\mathbb{R}^m$ to $\mathbb{R}^n$. If a first-order approximation suffices, then the number can be reduced to $2m$ sample points.

If basic hypotheses specified by uniform value ranges have to be transformed by a mapping $f : \mathbb{R}^m \mapsto \mathbb{R}^n$, apart from the computational costs it has also to be considered that a uniform distribution is generally not preserved when propagated through a non-linear function.

[3] A discrete convolution calculates the weighted sum of the function values at discrete points in the neighbourhood of a reference point. The weights are defined by the convolution mask.

Even for a linear transformation of uniformly distributed random variables with multiple independent dimensions it is not guaranteed that the uniform distribution is preserved. This becomes obvious when we consider the following example.

We assume that the gradient for a function $f : \mathbb{R}^2 \mapsto \mathbb{R}$ is $(1,1)^T$ and all the elements of the Hessian matrix as well as $f(x_0, y_0)$ are equal to 0. Consequently, the function is approximated by $f(x,y) \approx (x - x_0) + (y - y_0) = x + y - (x_0 + y_0)$. If we further assume that $x_0 = 0$ and $y_0 = 0$, it holds $f(x,y) \approx x + y$. However, it is known that the density of the sum of two independent random variables calculates as the convolution of the densities of the two random variables, i.e.

$$p_{x+y}(\xi) = \int p_x(v) \cdot p_y(\xi - v) dv \tag{8.6}$$

This results for uniformly distributed random variables X and Y, both in the range $[0,1]$, in the situation depicted in Figure 8.2.

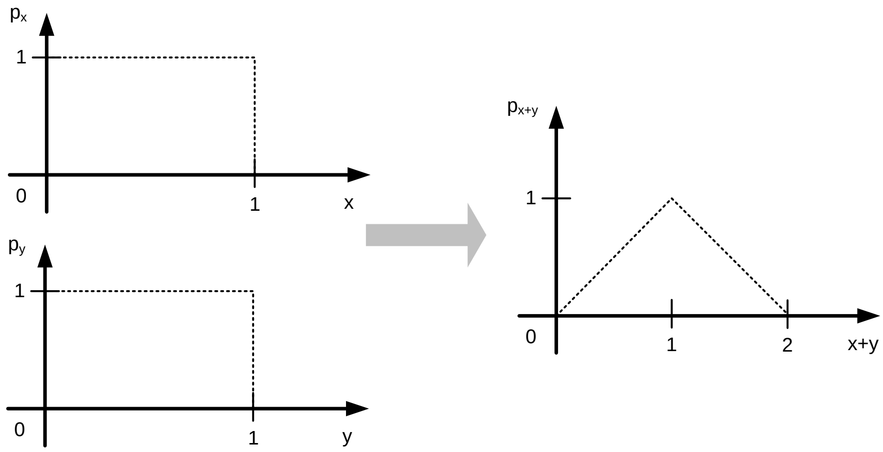

Figure 8.2: Density of the Sum of two Uniformly Distributed Random Variables

In order to check whether the uniform distribution is approximately preserved during the transformation, the following criteria can be applied:

1. The Taylor approximation is dominated in each dimension by a single linear term, i.e. the absolute value of a single element of the gradient is higher than the sum of the absolute values of all other elements of the gradient and the absolute values of all elements of the Hessian matrix by one or several orders of magnitude. However, here it is vitally important to mind that this criterion only holds if the value ranges are normalized, i.e. we assume the interval $[0,1]$ for all value range dimensions. The normalization can be considered as part of the transformation function, if we assume that the transformation function first realizes a linear transformation to map the interval $[0,1]$ to the actual value range and then performs the actual transformation. Of course, the Taylor approximation can also be adjusted after the transformation of the sample points by means of substitution.
2. Each dimension of the transformation function is dominated by a linear term in a different variable. Otherwise, the dimensions are not independent and a uniform value range over several dimensions cannot be specified anymore.

If the criteria described above are fulfilled, the calculation of the transformed uniform value range is straight-forward: We just have to apply the linear transformation. For most cases, however, it has to be expected that these criteria do not hold. Here we propose to approximate the uniform value range by a multivariate normal distribution retaining the mean and variances and to apply the same approach as for the transformation of normal distributions as described in Section 8.2.3.2. This way, at least the moments of the transformed distribution are approximated with reasonable precision.

The above criteria do not have to hold for the transformation of value ranges which do not make statements about the BBAs within the range as no assumptions of the transformed distribution can be made anyway. Hence, it suffices to compute the maximum extent of the transformed value range, i.e. the minimum and maximum value for each of the dimensions. This leads to a **bound constrained optimization problem**. More precisely speaking, if we utilize the second-order Taylor approximation of the transformation function, we have to solve a **quadratic programming problem with inequality constraints**. For the maximum bound of dimension i of the value range, the quadratic programming problem can be written as:

$$\begin{aligned} &\max_{\vec{x}} \left(f_i(\vec{x_0}) + (\vec{x} - \vec{x_0})^T \nabla f_i(\vec{x_0}) + \tfrac{1}{2}(\vec{x} - \vec{x_0})^T H_{f_i}(\vec{x_0})(\vec{x} - \vec{x_0}) \right) \\ &\qquad \vec{x} \geq \vec{b}_{min} \quad \text{and} \quad \vec{x} \leq \vec{b}_{max} \end{aligned} \tag{8.7}$$

where b_{min} describes the vector formed by the lower bounds of the original value range for the different dimensions, and $\vec{b}_{max}$ denotes the vector of the upper bounds. $\vec{x} \geq \vec{b}_{min}$ and $\vec{x} \leq \vec{b}_{max}$ mean that each element of the vector $\vec{x}$ has to be greater or equal than the corresponding element of $\vec{b}_{min}$ and less or equal than the corresponding element of $\vec{b}_{max}$.

Numerous approaches have been developed to (approximately) solve quadratic programming problems, as e.g. the method of the *conjugate gradient* [60]. In this dissertation, however, we apply a simple approach which is easier to implement, relies on standard gradient ascent and uses Resilient Propagation (RProp) [119] for determining the step size. Critical to all gradient ascent methods is the selection of the starting point $\vec{x}^{(0)}$, for which we propose the following heuristic. We inspect the signs of the elements of the gradient vector $\nabla f_i(\vec{x_0})$. If the sign is negative or 0, the lower bound of the value range for the corresponding dimension is used as element for the vector $\vec{x}^{(0)}$. If the sign is positive, the upper bound is used. From this starting point, $n = 10$ gradient ascent steps are performed:[4]

$$\vec{x'}^{(k+1)} = \vec{x}^{(k)} + \eta \left(\nabla f_i(\vec{x_0}) + H_{f_i}(\vec{x_0})(\vec{x}^{(k)} - \vec{x_0}) \right) \tag{8.8}$$

where the step size η is adaptively adjusted using RProp. In order to ensure that the inequality constraints are met, the elements of $\vec{x'}^{(k+1)}$ are normalized as follows:

$$x_j^{(k+1)} = \begin{cases} b_{min_j} & \text{, if } x'^{(k+1)}_j < b_{min_j} \\ b_{max_j} & \text{, if } x'^{(k+1)}_j > b_{max_j} \\ x'^{(k+1)}_j & \text{else} \end{cases} \tag{8.9}$$

[4]According to our tests, the number of 10 gradient ascent steps offers a good tradeoff between the quality of the result and the computational costs.

After $n = 10$ gradient ascent steps have been performed as described above, the maximum bound of the value range for dimension i can be calculated by transforming $\vec{x}^{(10)}$. The minimum bound is calculated analogously, but we have to perform a gradient descent instead. This procedure has to be applied for all dimensions of the resulting value range.

From the description above it becomes obvious that this approach requires quite high computational effort. Due to this fact, this approach should only be used if an accuracy better than provided by a first-order approach is essential. Otherwise, it is recommended to only consider the first-order Taylor approximation where the problem can efficiently be solved (only the starting vectors $\vec{x}^{(0)}$ for the minimization and maximization have to be transformed). We can also work on the first-order Taylor approximation if all elements of H_{f_i} are very small and the approximation is dominated by the linear terms.

8.2.3.2 Transformation of Normally Distributed Basic Hypotheses

Basic hypotheses that are specified with variances/covariance matrices define a basic mass assignment which corresponds to the probability density function (pdf) of a normally distributed random variable. Thus, the corresponding transformation reduces to the problem of propagating Gaussian pdfs through non-linear functions. Here too, it is possible to utilize the first-order or second-order Taylor approximation of the non-linear function. However, the *Unscented Transformation* represents a very efficient approach for the propagation of Gaussian pdfs. As it is based on sampling techniques and achieves at least second-order accuracy, it is particularly well-suited for the purposes of this thesis.

The Unscented Transformation was developed by Julier and Uhlmann [70] following the intuition that *it should be easier to approximate a given distribution with a fixed number of parameters than it is to approximate an arbitrary nonlinear mapping or transformation*. Based on this intuition, they have developed a parameterization that captures the mean and the covariance matrix of a Gaussian pdf, while being directly propagated through an arbitrary set of non-linear functions.

This is achieved by generating a discrete distribution having the same first and second (and possibly higher) moments where each point can directly be transformed. Afterwards, the mean and the covariance matrix of the transformed set of points can be calculated as an approximation of the transformed original distribution.

Intuitively, the Unscented Transformation samples the 1σ contour of a Gaussian random variable at $2m$ predefined points. Each of these sample points is propagated through the nonlinear transformation function, and finally the transformed set of sample points is used to determine the mean and the covariance matrix of the transformed pdf.

Formally, the Unscented Transformation allows to determine the n-dimensional Gaussian random variable $N(y, \Sigma_y)$[5] resulting from a transformation of the m-dimensional random variable $N(x, \Sigma_x)$ with the transformation function $f : \mathbb{R}^m \mapsto \mathbb{R}^n$ and $y = f(x)$ with the help of the following four steps:

1. Compute the set Z of $2m$ points from the rows or columns of the matrices $\pm\sqrt{m\Sigma_x}$, which is zero mean and has the covariance matrix Σ_x. The matrix square root can efficiently be calculated with the help of the Cholesky decomposition.

[5]Actually, the resulting pdf is only approximated with mean y and covariance matrix Σ_y. It can not be assumed that the result of the transformation constitutes a Gaussian distribution.

2. Compute the set of points X with the same covariance matrix Σ_x, but with mean x by translating each point in Z by $x_i = z_i + x$.
3. Compute the set of points Y by transforming all points of X with function f, i.e. calculate $y_i = f(x_i)$.
4. Compute y and Σ_y by calculating the mean and the covariance matrix of the $2m$ points of the set Y.

The Unscented Transformation achieves second-order (or better) accuracy in determining the mean, whereas only first-order accuracy is achieved by linearization. Although a linearization approach transforms the covariance matrix correctly up to the second-order as well, the absolute errors in the forth and higher order terms of the Unscented Transformation are smaller. It is also important to note here that for the Unscented Transformation it is sufficient to transform only $2m$ sample points. An estimation of gradients or Hessian matrices is not required.

8.2.3.3 Transformation of (Uniform) Value Ranges and Covariance Matrices in One Basic Hypothesis

In particular, if we consider dependencies of IROs to additionally required information, it may be necessary to transform basic hypotheses that involve value ranges, uniform value ranges and covariance matrices for different subsets of dimensions at the same time. For this case, we propose the following approach:

1. If the basic hypothesis involves no value ranges, but covariance matrices, all possibly involved uniform value ranges are approximated through a corresponding mean and covariance matrix.

   The rationale behind this is that uniform value ranges are likely to be not preserved during the transformation, in particular if additional information characterized by a covariance matrix is required as well. However, in this way the moments of the distribution can be preserved by applying the Unscented Transformation. The result of the transformation is a normally distributed basic hypothesis.

2. If the basic hypothesis involves value ranges, all possibly involved covariance matrices and uniform value ranges are assumed to define value ranges.

   A value range abstracts from concrete mass assignments to singletons and only specifies a belief assignment to the whole set of singletons within the value range. Therefore, the concrete mass assignments determined by the uniform value ranges and the covariance matrices are abstracted by a mass assignment for the whole subset of singletons as well. The result of the transformation is a basic hypothesis with a value range again. Another possibility would be to successively split up the value range in reasonably small sub-value ranges which can be abstracted by its means. Then we can consider the basic hypothesis as a complex hypothesis with a union of basic hypotheses corresponding to the sub-value ranges. However, this can be expected to require high computational costs. Furthermore, it is quite difficult to define '*reasonably small*' without knowing the actual properties of the transformation. The result of the transformation would be a set of basic hypotheses specifying (uniform) value ranges or covariance matrices.

3. Finally, the transformation of the basic hypothesis which now involves only value ranges, only (uniform) value ranges or only covariance matrices is performed with the help of the approaches presented in the previous sections.

As already mentioned above, for the application of the Unscented Transformation to basic hypotheses involving only covariance matrices and uniform value ranges, the uniform value ranges have to be approximated by a corresponding mean and a covariance matrix first. Afterwards, the different covariance matrices have to be combined to a single covariance matrix. In this regard, it is important to note that we assume the different subsets of dimensions covered by uniform value ranges or covariance matrices to be conditionally independent of the other dimensions. This is reasonable as the corresponding impreciseness or uncertainty of the dimensions is expressed independently, and the information may reflect independent observations. In particular, the latter holds for the additional dependencies of IROs.

For transforming a multivariate normal distribution into a value range, the covariance matrix is analyzed using the principal axis transformation first. This results in a diagonal matrix and a rotation matrix. The diagonal matrix represents the variances $\sigma_1^2, \ldots, \sigma_n^2$ along the principal axes. The corresponding value range is then bounded by $b_{min} = \alpha \cdot (-\sigma_1, \ldots, -\sigma_n)^T$ and $b_{max} = \alpha \cdot (+\sigma_1, \ldots, +\sigma_n)^T$. The required rotation and translation of the value range can be considered to be part of the transformation in the same way as we have realized the normalization of value ranges. The parameter α can be used to adjust the mass included in the value range. If we use $\alpha = 2$, for example, 95% of the mass is included in the value range. In this case, the mass assigned to the whole basic hypothesis is discounted by multiplication with 0.95.

Sometimes it may not be satisfactory to discard all concrete belief assignments and to transform uniform value ranges and covariance matrices to value ranges. For this case, we consider the following enhancement of the approach to be investigated in future work. First, the value range is abstracted only by its mean and the basic hypothesis (based on the primitive mean/covariance matrix or uniform value range) is transformed. Afterwards, the value range is kept and the other dimensions are abstracted by its mean. The resulting basic hypothesis is also transformed. If the value range resulting from the second transformation is '*reasonably small*' compared to the size of the uniform value range or to the size of the 1σ contour of the covariance matrix resulting from the first transformation, the result of the first transformation is used for the overall transformation. Otherwise, the result of the approach described above is used. However, it is difficult to evaluate if a value range is '*reasonably small*'. A possible criterion might be the ratio of the volume covered by the value range and the volume defined by the 1σ contour of the covariance matrix or the volume of the uniform value range.

8.2.3.4 Discrete Domains and Ranges in Case of Non-continuous IROs

As already stated above, this thesis focuses on the transformation of the measures for impreciseness and uncertainty for continuous domains and ranges of IROs and continuous and differentiable IROs. In this section, we would like to present some preliminary thoughts on the general case. However, a detailed investigation of the issue is out of the scope of this thesis and planned for future work.

First, we still consider IROs that have the mathematical property of continuity but work, for example, on integers, i.e. with discrete domains and/or ranges.

- **Discrete domain, continuous range**. In this case, there is a good chance that the approaches presented above also provide reasonable results as the approaches rely on the definition of appropriate sample points, and hence abstract the domain by a limited set of points anyway. Most difficulties have to be expected with the application of the Unscented Transformation as here the sample points do not form a grid structure. The chance that the approaches presented above provide reasonable results is expected to increase with the dimension of the (uniform) value ranges and the 1σ contour defined by the covariance matrix, i.e. we expect reasonable results when a great number of discrete domain values is involved.
- **Continuous domain, discrete range**. Here too, we expect that the chance that the approaches presented above provide reasonable results increases with the number of the involved singletons of the IRO range. With only a limited number of involved values, the gradient and the Hessian are difficult to estimate. It is also difficult to make assumptions on the distribution if only a very limited number of singletons of the IRO range is involved as a high discretization error has to be expected.
- **Discrete domain, discrete range**. This case can be considered in a way as the intersection of the cases discussed above. Consequently, good results can be expected only if a reasonable number of domain and range values are involved.

So far, we have mainly discussed the applicability of the approaches presented above for continuous domains and ranges and continuous IROs. However, it is always possible to split up the set of involved domain singletons in reasonable small subsets and to consider them as more fine-grained basic hypotheses which are transformed separately. The subsets should be selected so small that it is reasonable to abstract them by their means. Currently, we consider this approach as the only feasible one for the most general case. This method, however, can be expected to require high computational effort, and here too it is quite difficult to define '*reasonable small*'. At this point, it has to be noted that generally if two basic hypotheses are mapped to the same singleton of the IRO range, the corresponding mass assignments are summed up. This concept can be extended to the equality of (uniform) value ranges and mean/covariance matrices which result form the transformation of basic hypotheses. In particular for mean/covariance matrices, however, equality is expected to be hardly achieved. In Section 9.2.3.2, an approach to check whether multivariate normal distributions can be considered to be equal is presented. We also discuss the abstraction of several multivariate normal distributions by a single one.

Besides, it becomes obvious from the discussion above that the granularities of the basic hypotheses and their transformed basic hypotheses often cause problems. Here, with granularity we refer to the number of involved elements of the IRO domain or the IRO range respectively. Consider, for example, a transformation from a street name to GPS coordinates where a single street name is only reasonably represented by a set of GPS coordinates, e.g. a value range. If we assume here only the availability of a point-to-point transformation, one possibility would be to utilize the reverse transformation (GPS coordinates to street name) and to apply an appropriate search strategy in order to determine the bounds of the value range. However, in such cases it seems to be more reasonable that the developer takes care of providing the value range as part of the actual IRO, which should be indicated in the definition of an IRO by means of a corresponding property.

8.3 Transformation of Nested Scope Individuals

In general, a *Scope* individual provides information in form of individuals of a certain *Representation*. In turn, a *Complex Representation* refers to *Scope* individuals again. For converting from a *Complex Representation* of a *Scope* to another *Complex Representation* of the same *Scope*, it is not necessarily required that an IRO is explicitly defined. It may be sufficient that the corresponding IROs for the nested scopes are available. This is the case if the following conditions are met:

- The nested scopes, i.e. the dimensions of the representation can be converted independently. In particular, this means that no covariances are specified involving different dimensions.
- The nested scopes do not have a particular semantic as part of a certain Representation. Consider, for example, the conversion between two different Cartesian coordinate systems which differ in their units and in the direction of the axes. If there is no explicit IRO defined, the units of the coordinates are adjusted correctly, but a possible rearrangement of the axes would not be taken into account if the scopes *XCoordinate* and *YCoordinate* are transformed separately without having the particular context in mind.
- The set of nested scopes of the provided representation corresponds to (or is a subset of) the set of nested scopes of the requested representation, and the scopes of the provided and required representation can be related unambiguously.

Here too, a property is required for the representations to indicate if the conditions listed above are met.

If there are mass assignments for a hypothesis corresponding to an individual of a certain representation which includes nested scopes defining mass assignments as well, the mass assignments of the nested scopes are considered to be conditional mass assignments given the hypothesis is true. If there are several hypotheses, basic or complex, for different nested scopes, they are combined to a set of basic hypotheses in the same way as it is practiced when resolving dependencies of IROs to additionally required information.

8.4 Summary and Discussion

In the previous sections, numerous concepts have been described for the realization of IROs in general, but we have particularly focused on the transformation of measures for impreciseness and uncertainty. As the general case is very difficult to realize if only point-to-point transformations are available, a number of assumptions and simplifications have been made. Despite these assumptions and simplifications, the problem still remains very challenging and our approach shows some limitations. Therefore, a critical assessment of the proposed methods is presented in the following paragraphs. In particular, we discuss the issues of scalability and performance. But first, we provide a summary of the proposed concepts along with the different assumptions/simplifications we have acted on.

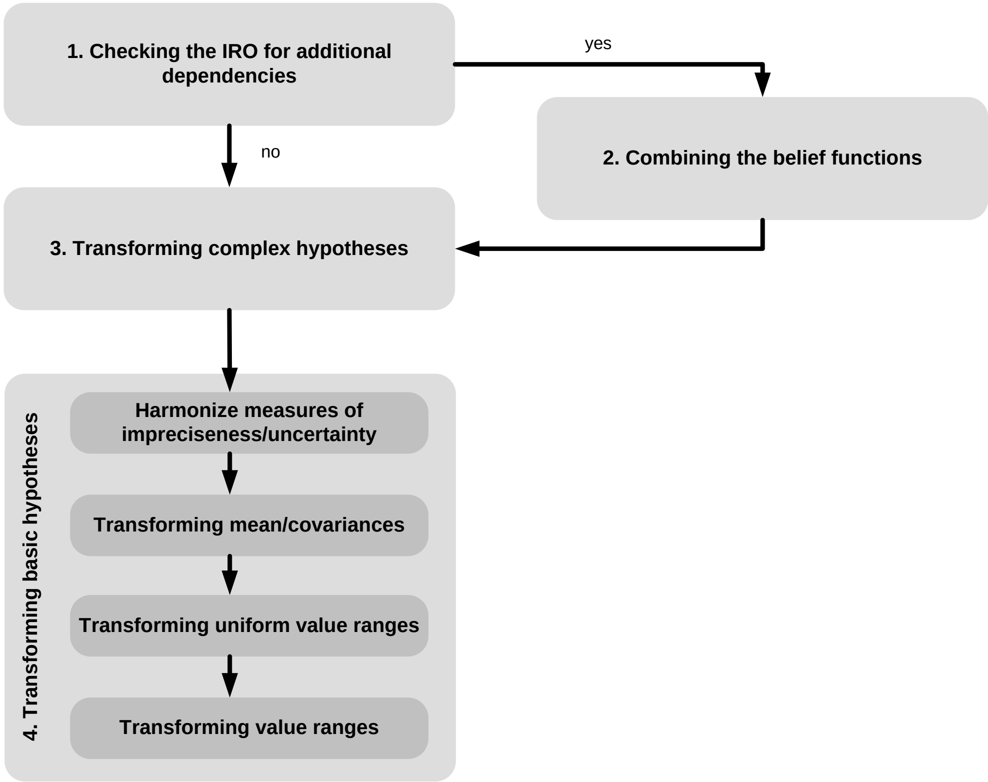

Figure 8.3: Overview of the IRO Approach for Transforming Measures of Impreciseness and Uncertainty

The presented approach mainly consists of the following four steps, which are also illustrated in Figure 8.3:

1. *Checking of the IRO for dependencies to additionally required information.*

2. *Resolving the additional dependencies and combining the belief functions.*

   Complex hypotheses are expressed as tuples of basic hypotheses of the involved *Scope* individuals. The tuples then correspond to the basic hypotheses of the new belief function. In total, a number of $b_{total} = b \cdot \prod_{i=1}^{d} b^{(i)}$ basic hypotheses has to be expected, where b depicts the number of basic hypotheses of the actual *Scope* individual, $b^{(i)}$ corresponds to the number of basic hypotheses of the i-th dependency, and d is the number of additional dependencies.

3. *Transforming the complex hypotheses.*

   Simplification: Complex hypotheses are transformed by processing the involved basic hypotheses and retaining the corresponding expressions of set-operations on the basic hypotheses. However, the correctness of this approach cannot be guaranteed for complex hypothesis involving the intersection and/or the complement of basic hypotheses. Hence, belief masses assigned to such complex hypotheses are assigned to Ω^r.

4. *Transforming the basic hypothesis*.

   Simplification: We assume that the IRO realizes a mapping $f : \mathbb{R}^m \mapsto \mathbb{R}^n$ which has the mathematical properties of continuity and infinite differentiability.

   a) *Harmonize measures of impreciseness and uncertainty*

      Simplification: Basic hypotheses involving different measures for impreciseness/uncertainty are transformed to involve only one kind of measure for impreciseness/uncertainty.

   b) *Transformation of basic hypotheses involving covariance matrices using the Unscented Transformation*. Here, the transformation of $2m$ sample points is required.

   c) *Transformation of basic hypotheses involving uniform value ranges*. For this purpose, first the properties of f within the value range are checked by transforming $1 + 2m + 2m(m-1)$ sample points and estimating the second-order Taylor approximation.

      Simplification: Uniform value ranges are assumed to be preserved across the transformation only if the Taylor approximation is dominated by a single linear term in each range dimension and the dominating terms across the range domains do not involve the same domain dimensions. Otherwise, the uniform value range is approximated by a mean and covariance matrix and transformed with the help of the Unscented Transformation.

   d) *Transformation of basic hypotheses involving value ranges*. For this purpose, first the properties of f within the value range are checked by transforming $1 + 2m + 2m(m-1)$ sample points and estimating the second-order Taylor approximation. If the Taylor approximation is dominated by the linear terms, then the transformed value range can directly be calculated. Otherwise a gradient ascent/descent is performed for each of the dimensions of the transformed value range in order to calculate its minimum and maximum bound.

8.4.1 Scalability and Performance

In the approach presented above, a number of different calculations/operations have to be performed, even within the single steps. For example, expressions of set-operations have to be transformed, Cholesky decompositions and principal axis transformations of covariance matrices have to be realized, and gradients and Hessians have to be estimated. All these operations cause considerable computational costs.

When we consider the scalability and performance of our approach, however, we particularly focus on the number of sample points to be transformed as we expect it to be the dominating factor with regard to performance and scalability. This is due to the following reasons:

- A great number of transformations may be necessary ($\mathcal{O}(b_{total} \cdot m^2)$)
- The transformation itself may be heavyweight and computational costly, e.g. if complex calculations have to be performed or a SWRL rule engine is invoked.
- The transformation may require the invocation of an external service with a potentially high communication overhead.

As b_{total} and m not only determine the number of transformations to be performed but along with the dimensionality n of the IRO range also the computational costs of most of the additionally required operations, it is obvious that these three factors have to be kept as small as possible. Whereas m and n have to be expected to be more or less fixed, b_{total} can be decreased by applying some heuristics. One possibility is to only use the k hypotheses with the highest belief mass and to assume the remaining belief mass to be assigned to Ω^p. Another possibility for significantly reducing the computational costs is to use just the first-order Taylor approximation of f instead of the second-order Taylor approximation. Here, only ($\mathcal{O}(b_{total} \cdot m)$) transformations have to be performed. Besides, it is also obvious that an IRO should be realized as a function of a local and dynamically loadable library whenever possible.

With regard to the number of basic hypotheses b_{total}, the problem of different granularities arises again if we consider the general case. Assume, for example, information about the location of the user in GPS coordinates with a covariance matrix defining a very large 1σ contour which covers half of a city. If this information is transformed to the address of buildings, this results in a huge number of basic hypotheses. In this case, a transformation to location information that covers whole areas of a city and still allows to reason about the association of buildings to the area would be much more reasonable. However, this imposes several additional challenges and is out of the scope of this thesis.

A detailed investigation of the performance of the proposed approach with concrete statements about the time required for performing different IROs is presented as part of the evaluation (see Part III).

8.4.2 Limitations

A considerable effort has been spent for the transformation of basic hypotheses in order to achieve an accuracy better than provided by approaches based on the first-order Taylor approximation. As already mentioned above, for time-critical applications the option of restricting the transformation to only utilize a first-order Taylor approximation has to be taken into account if we can expect that b_{total}, m, or n becomes too big. However, this has to be traded off against potentially poor approximation results.

Another big limitation is that currently we have only available a comprehensive approach for transforming measures of impreciseness and uncertainty if the corresponding IROs define a mapping $\mathbb{R}^m \mapsto \mathbb{R}^n$ which is continuous and infinitely differentiable. For the general case, the only currently available solution is to split up the set of involved domain singletons in small subsets and to consider them as more fine-grained basic hypotheses which are transformed separately. However, this can be expected to result in huge computational costs. For some IROs, e.g. the conversion of the name of a building to GPS coordinates, we even have to rely that the developer provides the corresponding value range by herself.

Another issue to be considered is that an IRO may not be able to perform the transformation for all input values. We assume again, for example, the conversion of GPS coordinates to the address of a building. In this case, it is obvious that not all GPS coordinates can reasonably be transformed, e.g. if the user is currently hiking in a big forest. In this case, the only possibility we currently see is the assignment of the corresponding belief mass of the basic hypotheses to Ω^p.

9 Information Fusion

Computing environments as envisaged in this thesis are characterized by the dynamic appearance and disappearance of information providers. This does not only mean that actually required information may not be available all the time, but also that several services may provide the requested information at the same time. In this chapter, we show how competitive information fusion based on the Dempster-Shafer Theory of Evidence can be applied in order to resolve contradictory statements and to combine a number of unreliable, uncertain and imprecise observations to obtain more reliable state estimates.

9.1 Requirements

Before presenting the concrete approach for information fusion based on the Dempster-Shafer Theory, first we summarize the requirements we have identified for the work of this thesis.

- Information providers, in particular sensors for real-world entities which are subject to a number of influences from the environment, are inherently unreliable. The varying reliability of different sensors has also to be taken into account in the information fusion approach.
- Sensor information is not only unreliable due to different sensor characteristics but also due to its age. For example, if information on the location of an entity which possibly changes its position significantly every second is older than a few seconds, it has to be considered extremely unreliable. This also has to be reflected in the information fusion approach.
- In our Information Model, uncertainty is represented through basic mass assignments to basic and complex hypotheses. In general, a basic mass assignment for a *Scope* individual is calculated as:

$$m(A) = \sum_{H \in \mathcal{H}} m(A|H) \cdot m(H) \tag{9.1}$$

  where A is an element of 2^{Ω^r}, H denotes a (basic or complex) hypothesis from the set of hypotheses $\mathcal{H}$, $m(A|H)$ describes the conditional basic mass assignment for A under the condition that H holds, and $m(H)$ denotes the belief mass that H holds. It is important to recall that all conditional mass assignments do not assign a mass to Ω^r. Hence, we introduce an implicit hypothesis H_{Ω^r} such that $m(\Omega^r|H_{\Omega^r}) = 1$ and $m(H_{\Omega^r}) = 1 - \sum_{H \in \mathcal{H}, H \neq H_{\Omega^r}} m(H)$, as already mentioned above. Now, competitive fusion of two *Scope* individuals has to be realized as conjunctive combination of two basic mass assignments m_1 and m_2 which both are represented in the form of Equation 9.1.

- It has to be expected that at least one dimension $\mathcal{D}_i^r$ of Ω^r is continuous, i.e. it allows values from $\mathbb{R}$ or from a subset of $\mathbb{R}$. If uniform value ranges and covariance matrices are specified for basic hypotheses which involve continuous dimensions, the basic mass assignment given by Equation 9.1 has to be considered as a mass density for the singletons of Ω^r. Consequently, appropriate support for dealing with such mass densities is required in our information fusion approach.
- The result of the information fusion should be a basic mass assignment or mass density which can be expressed in the way shown by Equation 9.1 and which only uses the already introduced primitives (mean/covariance matrix, uniform value range or value range). The introduction of new primitives has to be avoided as this would further complicate the approaches for the realization of IROs and the information fusion.

9.2 Fusion Approach

In the following sections, we present our information fusion approach based on the Dempster-Shafer Theory of Evidence in detail. In particular, we show how the requirements listed above are met.

9.2.1 Consideration of the Reliability of Information Providers

A common approach (already proposed by Shafer in [126]) to take into account the reliability of sensors, or of information providers in general, within the DST is the so called method of *discounting*. Intuitively, *discounting* shifts a fixed portion of the masses of all elements of 2^{Ω^r} which do not equal to Ω^r to $m(\Omega^r)$. The portion to be shifted is determined by the reliability of the sensor and is expressed by the discounting factor α. Typically α is selected as $1 - \textit{reliability}$. As $m(\Omega^r)$ can be interpreted as a measure of ignorance included in the basic mass assignment, a *discounting* decreases the belief in concrete observations and increases the ignorance expressed by the belief assignment. Formally, the method of discounting calculates the discounted basic mass assignment m_d from the original basic mass assignment m as follows:

$$\begin{aligned} m_d(A) &= (1-\alpha)\cdot m(A) \qquad \forall A \in 2^{\Omega^r}, A \neq \Omega^r \\ m_d(\Omega^r) &= m(\Omega^r) + \alpha \cdot (1 - m(\Omega^r)) \end{aligned} \tag{9.2}$$

With regard to basic mass assignment of the form shown in Equation 9.1, a discounting with factor α yields to:

$$m_d(A) = (1-\alpha)\cdot \sum_{H\in\mathcal{H}} m(A|H)\cdot m(H) \quad \forall A \in 2^{\Omega^r}, A \neq \Omega^r \tag{9.3}$$

which can be reformulated as:

$$m_d(A) = \sum_{H\in\mathcal{H}} m(A|H)\cdot ((1-\alpha)\cdot m(H)) \quad \forall A \in 2^{\Omega^r}, A \neq \Omega^r \tag{9.4}$$

If $m(\Omega^r|H) = 0$ ($H \in \mathcal{H}$, $H \neq H_{\Omega^r}$) and $m(\Omega^r|H_{\Omega^r}) = 1$, a *discounting* of a basic mass assignment which is expressed in the form of Equation 9.1 simplifies to a *discounting* of the basic mass assignments of the hypotheses. This is formally specified in Equation 9.5.

$$\begin{aligned} m_d(H) &= (1-\alpha)\cdot m(H) \qquad \forall H \in \mathcal{H}, H \neq H_{\Omega^r} \\ m_d\left(H_{\Omega^r}\right) &= m\left(H_{\Omega^r}\right) + \alpha \cdot \left(1 - m\left(H_{\Omega^r}\right)\right) \end{aligned} \tag{9.5}$$

9.2.2 Consideration of the Information Age

As motivated with the above example on position information for an entity of the world, the age of a piece of information has to be considered in the fusion approach as well. In general, it is justified to assume that the reliability of non-constant information decreases with increasing age. Consequently, the age of the information is taken into account by another discounting step which can be performed analogously to the discounting step for the consideration of sensor reliability. Thus, the problem is reduced to the selection of an appropriate discounting factor α.

As described in Section 7.2.5, a meta-data attribute *ChangeFrequency* can be associated to *Scope* individuals which denotes the frequency with which the provided information is expected to change significantly. When performing the information fusion step, this attribute can be used along with the *Timestamp* meta-data associated to the corresponding *Scope* individuals to calculate the discounting factor α. Here we propose the following equation which corresponds to the equation Shafer presented in [126] to address this problem but which is adjusted to the attributes available in our Information Model:

$$\alpha = 1 - e^{-f\cdot(t_n - t_s)} \tag{9.6}$$

where f corresponds to value of the attribute *ChangeFrequency*, t_n to the current time and t_s to the *Timestamp* meta-data of the *Scope* individual. With Equation 9.6, α evaluates to 0 if $t_n = t_s$, and evaluates to 0.95 for $(t_n - t_s) = 3 \cdot \frac{1}{f}$. This means that the reliability of the information is considered to be less than 5% if the difference of the current time and the time when the information was sensed is greater than three times the reciprocal of the value of the attribute *ChangeFrequency*.

If covariance matrices or (uniform) value ranges are specified for basic hypotheses, we could also consider the age of the information by increasing the covariance matrices or the dimensions of the (uniform) value ranges dependent on the age of the information. However, this would require additional information on how the values can be expected to change over time, which may be highly dependent on the *Scope* and the characterized *Entity*. For example, for information on the position of a moving object at least an estimation of the maximal velocity has to be known. As the required information is very specific to the *Scope* and *Entity*, integration of this idea in a generic framework as envisaged in this thesis is difficult. However, we will further investigate the idea as part of our future work.

9.2.3 Fusion of two Independent Basic Mass Assignments

As introduced in Section 3.5, two basic mass assignments m_1 and m_2 resulting from two independent observations are usually fused with *Dempster's Rule of Combination* or with the *Conjunctive Rule of Combination*. In this thesis, we have decided to employ *Dempster's Rule of Combination* as the unnormalized *Conjunctive Rule of Combination* tends to yield very small belief masses if several belief functions have to be combined, even if the conflict is quite small. For the reader's convenience *Dempster's Rule of Combination* is presented again in Equation 9.7.

$$m_{1,2}(\emptyset) = 0 \tag{9.7}$$

$$m_{1,2}(A) = \left(m_1 \oplus m_2\right)(A) = \frac{1}{1-K} \sum_{B \cap C = A \neq \emptyset} m_1(B)\, m_2(C)$$

$$K = \sum_{B \cap C = \emptyset} m_1(B)\, m_2(C)$$

With regard to two basic mass assignments in the form of Equation 9.1 the combination results in:

$$m_{1,2}(A) = \frac{1}{1-K} \sum_{B \cap C = A \neq \emptyset} \sum_{H_1 \in \mathcal{H}_1} m_1(B|H_1) \cdot m_1(H_1) \cdot \sum_{H_2 \in \mathcal{H}_2} m_2(C|H_2) \cdot m_2(H_2) \tag{9.8}$$

where K is analogously calculated as above and also $m_{1,2}(\emptyset) = 0$ is ensured. With some re-ordering of the terms and the operations in Equation 9.8 we get:

$$m_{1,2}(A) = \frac{1}{1-K} \sum_{H_1 \in \mathcal{H}_1} \sum_{H_2 \in \mathcal{H}_2} m_1(H_1) \cdot m_2(H_2) \sum_{B \cap C = A \neq \emptyset} m_1(B|H_1) \cdot m_2(C|H_2) \tag{9.9}$$

This equation can be interpreted in the following way:

- $\sum_{B \cap C = A \neq \emptyset} m_1(B|H_1) \cdot m_2(C|H_2)$ corresponds to the conjunctive combination (*Dempster's Rule of Combination* without normalization) of two conditional mass assignments on two hypotheses from $\mathcal{H}_1$ and $\mathcal{H}_2$ respectively. The result of this combination is written as $m_{1,2}(A|H_1, H_2)$ (in accordance with Smets in [132]).
- $m_1(H_1) \cdot m_2(H_2)$ is considered as weight for $m_{1,2}(A|H_1, H_2)$ and therefore as weight for the combined hypothesis.
- In this respect, $m_{1,2}(A)$ can be considered as weighted sum over all possible combined hypotheses. If the normalization factor $\frac{1}{1-K}$ is incorporated into the weights as well, this interpretation yields to an expression exactly the same as depicted in Equation 9.1.
- If two hypotheses H_1 and H_2 have no intersection, i.e. $m_{1,2}(A|H_1, H_2) = 0$ for all $A \in 2^{\Omega^r}$, then the whole mass $m_1(H_1) \cdot m_2(H_2)$ is shifted to K. Checking for intersection and calculating the product $m_1(H_1) \cdot m_2(H_2)$ can be considered as conjunctive combination of the hypotheses at an abstract level (hypotheses level):

$$m_{1,2}(H_c) = \frac{1}{1-K} \sum_{H_1 \oplus H_2 = H_c, H_c \neq \emptyset} m_1(H_1) \cdot m_2(H_2) \tag{9.10}$$

where H_c corresponds to the combined hypothesis and $\oplus$ denotes the combination of hypotheses.

In summary, the fusion of two basic mass assignments corresponding to Equation 9.1 can be considered as combination over two hierarchical levels. The mass assignments to the hypotheses from $\mathcal{H}_1$ and $\mathcal{H}_2$ are conjunctively combined at an abstract level, and the conditional mass assignments $m_1(B|H_1)$ and $m_2(C|H_2)$ are conjunctively combined to $m_{1,2}(A|H_1, H_2)$ at a concrete level, i.e. at the level of the elements $A \in 2^{\Omega^r}$. If we fuse the different conditional mass assignments independently, we have to take special care that the normalization factor $\frac{1}{1-K}$ is calculated consistently over all combinations as the resulting combined conditional mass assignments are assumed to be normalized to sum up to 1 (with $m_{1,2}(\emptyset|H_1, H_2) = 0$). This problem will be discussed in detail later on when we describe methods to fuse the conditional mass assignments for the different hypotheses and in particular when we present concepts for the fusion of mass densities.

9.2.3.1 Fusion of two Conditional Basic Mass Assignments

When combining two conditional basic mass assignments $m_1(B|H_1)$ and $m_2(C|H_2)$, we have to consider that H_1 as well as H_2 may represent complex hypotheses, which are formed by a logical expression of basic hypotheses. For example, H_1 may be the union of three basic hypotheses h_{11}, h_{12} and h_{13}, i.e. $h_{11} \cup h_{12} \cup h_{13}$. If we assume that H_2 represents a single basic hypothesis h_{21}, the combination of $m_1(A|H_1)$ and $m_2(A|H_2)$ consists of calculating $m_{1,2}\left(A|\left(h_{11} \cup h_{12} \cup h_{13}\right) \cap h_{21}\right)$.

Of course, the combination of two conditional basic mass assignments can always be represented through the intersection of the two logical expressions on basic hypotheses as shown in the example above. However, complex expressions make it difficult to detect conflicting hypotheses and make it hard to process the resulting belief functions in further reasoning steps. Thus, it is desirable to simplify the logical expression to a small number of new combined basic hypotheses if possible. For this purpose, we will review the basic combination rules and establish a number of simplification rules. Special care has to be taken that the resulting basic mass assignments adhere to the form shown in Equation 9.1 and that no additional primitives are required.

Dubois and Prade have listed the following three basic combination rules in [33]:

1. Intersection of two independent bodies of evidence (corresponding to the *Conjunctive Rule of Combination*):

$$m(A|H_1 \cap H_2) = (m_1 \cap m_2)(A) = \sum_{B \cap C = A} m_1(B) \cdot m_2(C) \tag{9.11}$$

2. Union of two independent bodies of evidence (corresponding to the *Disjunctive Rule of Combination*):

$$m(A|H_1 \cup H_2) = (m_1 \cup m_2)(A) = \sum_{B \cup C = A} m_1(B) \cdot m_2(C) \tag{9.12}$$

3. Complement/negation of a body of evidence:

$$\forall A \subseteq \Omega^r, m(A|\overline{H_1}) = \overline{m_1}(A) = m_1(\bar{A}) \tag{9.13}$$

Besides, the *Conjunctive Rule of Combination* and the *Disjunctive Rule of Combination* are known to be associative and commutative. Furthermore, the union and intersection of bodies of evidences satisfy De Morgan's laws [33], which is illustrated in Equation 9.14.

$$\overline{(m_1 \cup m_2)}(A) = (m_1 \cup m_2)(\bar{A}) = \sum_{B \cup C = \bar{A}} m_1(B) \cdot m_2(C) = \sum_{\bar{B} \cap \bar{C} = A} \overline{m_1}(\bar{B}) \cdot \overline{m_2}(\bar{C}) = (\overline{m_1} \cap \overline{m_2})(A) \tag{9.14}$$

Whereas the rules and properties for the combination of conditional basic mass assignments described above suggest that we can apply the commonly known set operations, it is vitally important to note that the union and intersection of bodies of evidences are not idempotent:

$$(m \cap m)(A) = \sum_{B \cap C = A} m(B) \cdot m(C) \neq m(A) \tag{9.15}$$

$$(m \cup m)(A) = \sum_{B \cup C = A} m(B) \cdot m(C) \neq m(A) \tag{9.16}$$

Moreover, also the distributive laws do not hold generally:

$$((m_1 \cup m_2) \cap m_3)(A) \neq ((m_1 \cap m_3) \cup (m_2 \cap m_3))(A) \tag{9.17}$$

$$((m_1 \cap m_2) \cup m_3)(A) \neq ((m_1 \cup m_3) \cap (m_2 \cup m_3))(A) \tag{9.18}$$

Equation 9.18 is trivial as also with usual sets the intersection is not distributive over union. Equation 9.17 gets clear, if we assume that m_3 only assigns masses to singletons. In this case, the result of the left side of Equation 9.17 assigns masses only to singletons again. The right side of Equation 9.17, however, may also assign masses to the union of two singletons.

In Section 8.2.2 it was described that the application of IROs for complex hypotheses either results in a mass shift to Ω^r (if negations or intersections of basic hypotheses are involved) or in complex hypotheses that consist of a union of basic hypotheses. Therefore, support for the fusion of unions of basic hypotheses is vitally important. Here, the fact that Equation 9.17 does not hold causes additional challenges and requires the establishment of new rules for the simplification of an intersection of unions of basic hypotheses.

As a baseline for the derivation of the simplification rules we first have a closer look on how the conditional basic mass assignment $m\left(E|(H_1 \cup H_2) \cap H_3\right)$ denoted as $\left((m_1 \cup m_2) \cap m_3\right)(E)$ is calculated:[1]

$$((m_1 \cup m_2) \cap m_3)(E) = \sum_{C \cap D = E, E \neq \emptyset} m_3(C) \cdot \sum_{A \cup B = D} m_1(A) \cdot m_2(B) \tag{9.19}$$

Dependent on C we now represent $m_1(A)$ as $m_1^{C+}(A) + m_1^{C-}(A)$ such that:

$$\begin{array}{llll} m_1^{C+}(A) = m_1(A) & \text{and} & m_1^{C-}(A) = 0 & \text{if } A \cap C \neq \emptyset \\ m_1^{C+}(A) = 0 & \text{and} & m_1^{C-}(A) = m_1(A) & \text{if } A \cap C = \emptyset \end{array} \tag{9.20}$$

When doing the same for $m_2(B)$, Equation 9.19 can be written as:

$$((m_1 \cup m_2) \cap m_3)(E) = \sum_{C \cap D = E, E \neq \emptyset} m_3(C) \cdot \sum_{A \cup B = D} (m_1^{C+}(A) + m_1^{C-}(A)) \cdot (m_2^{C+}(B) + m_2^{C-}(B)) \tag{9.21}$$

With some re-ordering of terms this yields:

$$\begin{aligned} ((m_1 \cup m_2) \cap m_3)(E) = & \sum_{C \cap D = E, E \neq \emptyset} m_3(C) \cdot \\ & \sum_{A \cup B = D} m_1^{C+}(A) \cdot m_2^{C+}(B) + m_1^{C+}(A) \cdot m_2^{C-}(B) + m_1^{C-}(A) \cdot m_2^{C+}(B) + m_1^{C-}(A) \cdot m_2^{C-}(B) \end{aligned} \tag{9.22}$$

In essence, the right side of Equation 9.22 is a sum of four terms:

1. $\sum_{C \cap D = E, E \neq \emptyset} m_3(C) \cdot \sum_{A \cup B = D} m_1^{C-}(A) \cdot m_2^{C-}(B)$, which is denoted as $m_{1,2,3}^{--}(E)$.
   It holds $m_{1,2,3}^{--}(E) = 0$ for all $E \neq \emptyset$, as $m_1^{C-}(A) \cdot m_2^{C-}(B)$ may only be greater than zero if $A \cap C = \emptyset$ and $B \cap C = \emptyset$. However, this also means that $C \cap (A \cup B) = \emptyset$, which does not meet the assertion $C \cap D = E, E \neq \emptyset$.
2. $\sum_{C \cap D = E, E \neq \emptyset} m_3(C) \cdot \sum_{A \cup B = D} m_1^{C+}(A) \cdot m_2^{C-}(B)$, which is denoted as $m_{1,2,3}^{+-}(E)$.
   $m_{1,2,3}^{+-}(E)$ can also be written as:

$$\begin{aligned} m_{1,2,3}^{+-}(E) &= \sum_{C \cap D = E, E \neq \emptyset} m_3(C) \cdot \sum_{A \cup B = D} m_1^{C+}(A) \cdot m_2^{C-}(B) \\ &= \sum_{C \cap A = E, E \neq \emptyset} m_3(C) \cdot m_1^{C+}(A) \cdot \sum_B m_2^{C-}(B) \\ &= \sum_{C \cap A = E, E \neq \emptyset} m_3(C) \cdot m_1(A) \cdot \sum_B m_2^{C-}(B) \end{aligned} \tag{9.23}$$

   Here, $\sum_B m_2^{C-}(B)$ corresponds in a way to the conflict of C with regard to m_2.

[1] Actually, $(m_1 \cap m_2)(A)$ denotes the combination of two basic mass assignments according to the Conjunctive Rule of Combination. In this thesis, however, we employ Dempster's Rule of Combination. Therefore, we introduce the assertion $E \neq \emptyset$. Nevertheless, we keep $(m_1 \cap m_2)$(A) as shorthand notation.

3. $\sum_{C\cap D=E,E\neq\emptyset} m_3(C)\cdot \sum_{A\cup B=D} m_1^{C-}(A)\cdot m_2^{C+}(B)$, which is denoted as $m_{1,2,3}^{-+}(E)$.

   Can be rewritten analogously to Equation 9.23.

4. $\sum_{C\cap D=E,E\neq\emptyset} m_3(C)\cdot \sum_{A\cup B=D} m_1^{C+}(A)\cdot m_2^{C+}(B)$, which is denoted as $m_{1,2,3}^{++}(E)$.

   If C is a singleton, i.e. $C\in\Omega^r$, then $m_3(C)\cdot\sum_{A\cup B=D} m_1^{C+}(A)\cdot m_2^{C+}(B)$ may only be greater than zero if $A=B$. Otherwise, either $m_1^{C+}(A)$ or $m_2^{C+}(B)$ is zero.

As already mentioned above, special care has to be taken that the normalization factor $\frac{1}{1-K}$ of Equation 9.9 is calculated consistently over all combinations when we fuse the hypotheses of two independent basic mass assignments. This can be realized by representing K as the sum of all conflicts, i.e. mass assignments to the empty set, across all combinations. The same result is achieved if we assign $A^*\cdot m_1(H_1)\cdot m_2(H_2)$ as mass to the combined hypotheses of $H_1\in\mathcal{H}_1$ and $H_2\in\mathcal{H}_2$ and normalize the sum of the mass assignments of the hypotheses (including the hypothesis representing the whole frame of discernment) to 1 after all fusion steps have been performed.[2] Here, the factor $A^*=1-K^*$ denotes the agreement of the two hypotheses if K^* represents the conflict resulting from the fusion of the particular conditional basic mass assignments $m_1(A|H_1)$ and $m_2(A|H_2)$.[3] When deriving the rules for simplification we will also discuss how A^* is calculated for the particular rule.

For the simplification of expressions of complex hypotheses we have derived the following rules most of which are focused on the primitives mean/covariance matrix, uniform value range and value range. Here, m_i denotes the conditional mass assignment corresponding to H_i.

- **Rule 1:** $((m_1\cup m_2)\cap m_3)(E)=((m_1\cap m_3)\cup(m_2\cap m_3))(E)$

  if m_1, m_2 and m_3 are all based on the primitive value range or are unions of value ranges.

  $A^*=1$, if the hypotheses H_1 and H_3 or H_2 and H_3 have an intersection, $A^*=0$ otherwise.

  Proof: Trivial.

- **Rule 2:** $((m_1\cup m_2)\cap m_3)(E)=(m_1\cap m_3)(E)$

  if H_1 and H_3 have an intersection and H_2 and H_3 have no intersection.

  $A^*=1-K^*_{H_1\oplus H_3}$, where $K^*_{H_1\oplus H_3}$ denotes the conflict resulting from the combination of H_1 and H_3.

  Proof: $m_2^{C+}(B)=0,\forall C,\forall B$. This means that $m_{1,2,3}^{++}(E)=0$ and $m_{1,2,3}^{-+}(E)=0$. $\sum_B m_2^{C-}(B)=1,\forall C$. Thus $m_{1,2,3}^{+-}(E)=(m_1\cap m_3)(E)$.

[2] As already mentioned above, we assume that the conditional basic mass assignments of the combined hypotheses are normalized to sum up to 1.

[3] For the remainder of the document A^* and K^* denote the agreement and the conflict when combining particular basic or complex hypotheses, whereas K corresponds to the overall conflict across all involved combinations of hypotheses.

- **Rule 3:** $((m_1 \cup m_2) \cap m_3)(E) = (m_1 \cap m_3 + m_2 \cap m_3)(E)$

  if H_1 and H_2 have no intersection and m_3 only assigns masses to singletons, i.e. is based on the primitive mean/covariance matrix or uniform value range.

  As this rule results in two hypotheses, we have to calculate two factors $A^*_{1,3}$ (for $(m_1 \cap m_3)(E)$) and $A^*_{2,3}$ (for $(m_2 \cap m_3)(E)$):

$$
\begin{aligned}
A^*_{1,3} = A^*_{H_1 \oplus H_3} &= 1 - K^*_{H_1 \oplus H_3} && \text{if } (A^*_{H_1 \oplus H_3} + A^*_{H_2 \oplus H_3}) \leq 1 \\
A^*_{1,3} &= \frac{A^*_{H_1 \oplus H_3}}{A^*_{H_1 \oplus H_3} + A^*_{H_2 \oplus H_3}} && \text{otherwise} \\
A^*_{2,3} = A^*_{H_2 \oplus H_3} &= 1 - K^*_{H_2 \oplus H_3} && \text{if } (A^*_{H_1 \oplus H_3} + A^*_{H_2 \oplus H_3}) \leq 1 \\
A^*_{2,3} &= \frac{A^*_{H_2 \oplus H_3}}{A^*_{H_1 \oplus H_3} + A^*_{H_2 \oplus H_3}} && \text{otherwise}
\end{aligned} \tag{9.24}
$$

The rationale behind the normalization with $A^*_{H_1 \oplus H_3} + A^*_{H_2 \oplus H_3}$ will become clear when we discuss the calculation of the agreement for the combination of hypotheses based on the primitives mean/covariance matrix and uniform value range. Here, it has to be ensured that the agreements do not sum up to a value greater than 1.

Proof: As already stated above, if m_3 only assigns masses to singletons C, i.e. $C \in \Omega^r$, then $m_3(C) \cdot \sum_{A \cup B = D} m_1^{C+}(A) \cdot m_2^{C+}(B)$ may only be greater than zero if $A = B$. Otherwise either $m_1^{C+}(A)$ or $m_2^{C+}(B)$ is zero. As H_1 and H_2 have no intersection, $m^{++}_{1,2,3}(E) = 0$.

If $m_3(C) \cdot m_1(A) > 0$, then $m_2^{C-}(B) = m_2(B), \forall B$, as H_1 and H_2 have no intersection. Thus $m^{+-}_{1,2,3}(E) = (m_1 \cap m_3)(E)$.

Analogously, it holds $m^{-+}_{1,2,3}(E) = (m_2 \cap m_3)(E)$.

All rules can be applied several times or consecutively for the simplification of more complex intersections of unions of hypotheses. For this purpose, the union of two or more hypotheses may be abstracted by a single hypothesis. With the help of this principle we have derived, for example, the following two rules:

- **Rule 4:** $((m_1 \cup m_2) \cap (m_3 \cup m_4))(E) = (m_1 \cap m_3)(E)$

  if H_2 and H_3, H_1 and H_4 as well as H_2 and H_4 have no intersection.

  $A^* = 1 - K^*_{H_1 \oplus H_3}$.

Proof: H_1 and H_2 are abstracted by a single hypothesis and Rule 2 is applied. To the result Rule 2 can be applied again.

- **Rule 5:** $((m_1 \cup m_2) \cap (m_3 \cup m_4)(E) = (m_1 \cap m_3 + m_2 \cap m_3)(E)$

  if H_1 and H_2, H_1 and H_4 as well as H_2 and H_4 have no intersection and m_3 only assigns masses to singletons, i.e. it is based on the primitive mean/covariance matrix or uniform value range.

  A^* is calculated analogously as for the application of Rule 3.

  Proof: H_1 and H_2 are abstracted by a single hypothesis and Rule 2 is applied. To the result Rule 3 can be applied again.

Actually, we have derived some more simplification rules. However, the results of the application of these rules cannot be expressed according to Equation 9.1 anymore, and thus these rules are omitted here. The same problem also arises if rules are derived for the simplification of expressions which involve the negation of hypotheses. So far, we have derived only the following simplification rule for negations of hypotheses.

- **Rule 6:** $(\overline{m_1} \cap m_2)(E) = m_2(E)$

  if H_1 and H_2 have no intersection.

  $A^* = 1$.

  Proof: if $m_2(B) > 0$ and $\overline{m_1}(C) > 0$, it holds $B \cap C = B$ as H_1 and H_2 have no intersection. Thus $(\overline{m_1} \cap m_2)(B) = m_2(B) \cdot \sum_C \overline{m_1}(C) = m_2(B)$.

Rule 1 to Rule 6 along with the commutativity/associativity of the Conjunctive and Disjunctive Rule of Combination and De Morgan's laws provide a good set of concepts for the simplification of complex hypotheses. As each rule has a number of guarding conditions, however, we certainly have to expect expressions that cannot be simplified. It may even be difficult to represent the result of the combination by a single basic hypothesis if a rule can be applied. Consider, for example, the case where the result of a simplification is the combination of a uniform value range and a mean/covariance matrix. In this case, we just keep the two hypotheses and assign the mass to $H_1 \cap H_2$.

For decision making, however, an estimation of the conflict or agreement is necessary also for the expressions which cannot be simplified. This is due to the fact that the overall conflict is used for the normalization of the belief masses, and thus it influences the absolute values of the belief masses of the involved hypotheses. Therefore, we apply the following preliminary heuristic to the fused belief function before decision making. A detailed investigation of the heuristic and the search for alternatives is subject of future work.

1. First, the complex hypothesis which cannot be simplified is transformed in a disjunction of conjunctions, i.e. the logical expression of the complex hypothesis is represented in *disjunctive normal form*.[4] This can be achieved by applying the usual logical operators

[4]It is important to note here that this is only done at the hypotheses level and does not involve the combination of conditional mass assignments.

and the laws holding for propositional logic which also comprise the distributive law and De Morgan's laws.

2. The agreement of the expression is now calculated as:

$$A^* = \max_i(A_i^*) \tag{9.25}$$

where the A_i^* represent the agreement of the single involved conjunctions and evaluate to:

$$A_i^* = \prod_{1 \le j \le n, j < k \le n} A^*_{H_j \oplus H_k} \tag{9.26}$$

where n is the number of hypotheses in the conjunction and $A^*_{H_j \oplus H_k}$ depicts the agreement of H_j and H_k. If H_j and/or H_k represent the negation of a hypothesis, then the agreement is calculated as $A^*_{H_j \oplus H_k} = 1 - A^*_{h_j \oplus h_k}$, where h_j and h_k denote the basic hypotheses constituting H_j or H_k or referred to in H_j and H_k respectively. Of course, this estimation requires that for all combinations of basic hypotheses an agreement can be calculated. We will show in the next section how this can be achieved.

The rationale behind Equation 9.26 is that in a conjunction of hypotheses each involved hypothesis has to be 'intersected' with each of the other hypotheses. Therefore, we multiply the agreements of all $n(n-1)/2$ pairs of hypotheses to derive the overall agreement in the conjunction.

As in a disjunction of conjunctions it is sufficient that only a single conjunction holds, the overall agreement of the complex hypothesis is calculated as the maximum agreement of all involved conjunctions (see Equation 9.25).

From the descriptions above, it becomes obvious that it is vitally important to be able to check two basic hypotheses for intersection and to calculate/estimate their agreement. In the following sections, we discuss these challenges for the combination of the primitives value range, uniform value range and mean/covariance matrix. Later on, we will also discuss how the properties *DSTintersectedWith*, *DSTincludes* and its inverse property *DSTisIncludedIn* defined in the ontology (see Section 7.2.6) are used to check whether basic hypotheses have an intersection.

9.2.3.2 Intersection, Combination and Agreement of two Basic Hypotheses

Combination of Value Ranges

Checking for intersection, combining and calculating the agreement for two basic hypotheses is straight-forward if the hypotheses are based on the primitive value range.

Two value ranges are checked for intersection by inspecting the single dimensions of the value ranges. If for all dimensions the intersection of the value ranges is unequal to the empty set, the two value ranges have an intersection. The new limits consist of the intersected value ranges of the single dimensions. For the agreement it holds $A^* = 1$ if the two value ranges, i.e. the two basic hypotheses, have an intersection, and it holds $A^* = 0$ otherwise.

Mass Densities

Before we discuss the combination of basic hypotheses of the type mean/covariance matrix and uniform value range, it is vitally important to recall their interpretation for continuous dimensions of Ω^r:

Basic hypotheses based on the primitives mean/covariance matrix and uniform value range are considered to only assign masses to singletons of Ω^r. In this respect, the mass assignments correspond to a probability assignment in the usual sense. Consequently, if continuous dimensions are involved in Ω^r, a basic mass assignment m has to be interpreted as probability density function, which we also refer to as mass density. Based on our interpretation that masses are only assigned to singletons, the conjunctive combination corresponds to a pointwise multiplication of the two basic mass assignments:

$$(m_1 \cap m_2)(\vec{x}) = m_1(\vec{x}) \cdot m_2(\vec{x}) \tag{9.27}$$

As we assume that all conditional mass assignments sum up to 1, we have to normalize the mass assignment of Equation 9.27 with the factor $\frac{1}{\gamma}$, where γ evaluates to:

$$\gamma = \int_{\vec{x} \in \Omega^r} m_1(\vec{x}) \cdot m_2(\vec{x}) d\vec{x} \tag{9.28}$$

Actually, γ corresponds to the agreement $A^* = 1 - K^*$ of the hypotheses to be combined. However using γ as agreement turns out to be a problem when dealing with mass densities. Consider, for example, the combination of two one-dimensional uniform value ranges in the interval $[0, 1]$. As result we get the same interval and for the agreement it holds $A^* = \gamma = 1$, which corresponds to our intuition. However, if working on a different scale and combining two one-dimensional uniform value ranges in the interval $[0, 10]$, then the resulting interval still remains the same, i.e. $[0, 10]$, but the agreement evaluates to $A^* = \gamma = 0.1$, which is in a way counter-intuitive. As the agreement of two basic hypotheses based on mass densities is heavily dependent on the scale, we will propose an alternative calculation of the agreement A^* if the primitives mean/covariance matrix and/or uniform value range are involved.

Combination of two Basic Hypotheses based on the Primitive Mean/Covariance Matrix

A basic hypothesis based on the primitive mean/covariance matrix defines a basic mass assignment (mass density):

$$m(\vec{x}) = m_N(\vec{x}, \vec{x}_0, \Sigma_x) = \frac{1}{\sqrt{(2\pi)^n \det(\Sigma_x)}} e^{-\frac{1}{2}(\vec{x}-\vec{x}_0)^T \Sigma_x^{-1} (\vec{x}-\vec{x}_0)} \tag{9.29}$$

where $\vec{x}_0$ denotes the mean, Σ_x corresponds to the covariance matrix and n is the number of dimensions of Ω^r. For checking two basic hypotheses based on the primitive mean/covariance matrix for intersection, we utilize the Mahalanobis distance [80] of the two corresponding Gaussian random variables:

$$d_m(H_1, H_2) = \sqrt{(\vec{x}_0^{H_1} - \vec{x}_0^{H_2})^T (\Sigma_x^{H_1} + \Sigma_x^{H_2})^{-1} (\vec{x}_0^{H_1} - \vec{x}_0^{H_2})} \tag{9.30}$$

where, $\vec{x}_0^{H_1}$ and $\vec{x}_0^{H_2}$ are the means of the hypotheses H_1 and H_2, and $\Sigma_x^{H_1}$ and $\Sigma_x^{H_2}$ are the covariance matrices of H_1 and H_2. The two hypotheses are assumed to have an intersection if the Mahalanobis distance is smaller than a certain threshold α, e.g. $\alpha = 4$. In many AI applications this criterion is used to solve the data association problem [125], i.e. to decide whether an estimate of an object's state corresponds to an observation. Both, the object's state and the observation, are represented as Gaussian random variables. If the test is successfully passed, it is assumed that both variables are related to the same object, and thus the observation can be used to refine the state estimate of the corresponding object.

The combination of two basic hypotheses, which only assign masses to singletons, corresponds to the pointwise multiplication of the mass densities. As the mass densities are given as two Gaussian functions, the result is a scaled Gaussian function with scaling factor B:

$$\begin{aligned}(m_1 \cap m_2)(\vec{x}) &= m_N(\vec{x}, \vec{x}_0^{H_1}, \Sigma_x^{H_1}) \cdot m_N(\vec{x}, \vec{x}_0^{H_2}, \Sigma_x^{H_2}) \\ &= B \cdot m_N(\vec{x}, \vec{x}_0^{H_{12}}, \Sigma_x^{H_{12}}) \end{aligned} \tag{9.31}$$

where $\vec{x}_0^{H_{12}}$ and $\Sigma_x^{H_{12}}$ represent the mean and the covariance matrix of the combined hypotheses. They evaluate as:

$$\begin{aligned}\vec{x}_0^{H_{12}} &= \Sigma_x^{H_2} \left(\Sigma_x^{H_1} + \Sigma_x^{H_2}\right)^{-1} \vec{x}_0^{H_1} + \Sigma_x^{H_1} \left(\Sigma_x^{H_1} + \Sigma_x^{H_2}\right)^{-1} \vec{x}_0^{H_2} \\ \Sigma_x^{H_{12}} &= \Sigma_x^{H_1} \left(\Sigma_x^{H_1} + \Sigma_x^{H_2}\right)^{-1} \Sigma_x^{H_2} \end{aligned} \tag{9.32}$$

As already discussed above, for mass densities the calculation of the agreement according to $A^* = \gamma$ is highly dependent on the scale, and thus is in a way counter-intuitive. Therefore, we propose to determine the agreement of two basic hypotheses based on the primitive mean/covariance matrix as follows:

$$A^* = \max\left(1 - \frac{d_m(H_1, H_2)}{\alpha}, 0\right) \tag{9.33}$$

where α corresponds to the threshold used to check whether the two basic hypotheses have an intersection. This formula can be interpreted as a rough estimation of the portion of one hypothesis covered by the other hypothesis. Here, we assume that two hypotheses do not overlap at all if $d_m(H_1, H_2) > \alpha$.

In order to abstract two independent hypotheses by a single one, i.e. to check the equality of two hypotheses, the Mahalanobis distance is not appropriate as it only checks the distance of the two means but does not take into account the similarity of the covariance matrices. Thus, we propose to use the squared Bhattacharyya distance of two Gaussian random variables instead, which evaluates as:

$$d_b^2(H_1, H_2) = \frac{1}{4} \cdot d_m^2(H_1, H_2) + \frac{1}{2} \ln \frac{\det(\Sigma_x^{H_1} + \Sigma_x^{H_2})}{2\sqrt{\det(\Sigma_x^{H_1}) \det(\Sigma_x^{H_2})}} \tag{9.34}$$

Two hypotheses are considered to be equal if $d_b^2(H_1, H_2) < \beta$, with e.g. $\beta = 0.01$. Two equal hypotheses are merged by summing up the mass assignments $m(H_1)$ and $m(H_2)$, and to use $m(A|H_1)$ as the new conditional mass assignment.

In order to reduce the number of basic hypotheses, it may be required to abstract a number of similar normal distributions (high agreement, but Bhattacharyya distance greater than β) by a single basic hypothesis. In this case, we represent each normal distribution by a minimal number of sample points as done in the Unscented Transformation [70] and weight the sample points according to the belief mass which is assigned to the corresponding basic hypothesis. The mean and the covariance matrix of the new basic hypothesis can then be calculated as the mean and the covariance matrix of the set of weighted sample points of all basic hypotheses to be abstracted.

Combination of two uniform value ranges

If we combine two uniform value ranges, the check for intersection and the calculation of the limits of the combined value range are performed in the same way as for the combination of value ranges. The result of the combination is a uniform value range again as it holds:

$$\begin{aligned} (m_1 \cap m_2)(\vec{x}) &= \frac{1}{V_{H_1} V_{H_2}} \quad \text{if } m_1(\vec{x}) > 0, m_2(\vec{x}) > 0 \\ (m_1 \cap m_2)(\vec{x}) &= 0 \quad \text{otherwise} \end{aligned} \tag{9.35}$$

where V_{H_1} denotes the volume of the uniform value range associated to H_1 and where V_{H_2} corresponds to the volume of the uniform value range associated to H_2. Here too, the mass assignment has to be normalized to sum up to 1 with the help of the factor $\frac{1}{\gamma}$, where γ is calculated according to Equation 9.28. As already discussed above, using γ as agreement would yield counter-intuitive results. Therefore, we propose the following alternative way to calculate the agreement A^* of the two hypotheses:

$$A^* = \max\left(\frac{V_{H_{12}}}{V_{H_1}}, \frac{V_{H_{12}}}{V_{H_2}}\right) \tag{9.36}$$

This means, the agreement is calculated as the maximum overlap of the volumes of the combined hypothesis $V_{H_{12}}$ and of the two original hypotheses (V_{H_2} and V_{H_2}).

Combination of a uniform value range and a value range

The test for intersection and the calculation of the limits are performed in the same way as for the combination of two value ranges. The result of the combination is a uniform value range again as it holds:

$$\begin{aligned} (m_1 \cap m_2)(\vec{x}) &= \frac{1}{V_{H_1}} \quad \text{if } m_1(\vec{x}) > 0, \vec{x} \in B, m_2(B) > 0 \\ (m_1 \cap m_2)(\vec{x}) &= 0 \quad \text{otherwise} \end{aligned} \tag{9.37}$$

where m_1 denotes the basic mass assignment of the uniform value range, and m_2 corresponds to the basic mass assignment of the value range.

In general, it holds that the combination of a basic mass assignment only assigning masses to singletons with another basic mass assignment results in a basic mass assignment which only assigns masses to singletons as well.

Here, the factor γ required to normalize the mass assignment of Equation 9.37 (with $\frac{1}{\gamma}$) corresponds to the ratio, the uniform value range associated to H_1 is covered by the value range associated to H_2. Therefore, we do not see the requirement for an alternative way to calculate the agreement and use $A^* = \gamma$.

In order to check two (uniform) value ranges for equality, we apply the following approach. Two (uniform) value ranges are considered to be equal if it holds:

$$\min\left(\frac{V_{H_{12}}}{V_{H_1}}, \frac{V_{H_{12}}}{V_{H_2}}\right) > \beta \tag{9.38}$$

with $\beta = 0.95$, for example. The limits of the new (uniform) value range are formed by the minimum/maximum limits of the two original (uniform) value ranges, which result from the union of the (uniform) value ranges defined for the single involved dimensions.

Combination of a basic hypothesis based on mean/covariance matrix with a (uniform) value range

The combination of a basic hypothesis based on mean/covariance matrix with a (uniform) value range is most difficult. In both cases, the result is a basic mass assignment that only assigns masses to singletons, but the result cannot be represented by one of the primitives used in this thesis. Thus, in the following paragraphs we will concentrate on the check for intersection and on the estimation of the agreement of the two basic hypotheses.

For checking the intersection and estimating the agreement, the value ranges are transformed into a mean/covariance matrix representation. This allows to calculate a Mahalanobis distance between the mean/covariance matrix and the (uniform) value range, although a value range does not specify a mass density but only assigns a mass to the union of singletons included in the value range. With the help of the Mahalanobis distance we can easily check the basic hypotheses for intersection and estimate their overlap.

The mean $x_0^{(i)}$ for dimension i is calculated as:

$$x_0^{(i)} = 0.5 \cdot (x_{min}^{(i)} + x_{max}^{(i)}) \tag{9.39}$$

and the variance $\sigma^2_{(i)}$ for dimension i evaluates as:

$$\sigma^2_{(i)} = \frac{(x_{max}^{(i)} - x_{min}^{(i)})^2}{12} \tag{9.40}$$

As the single dimensions are considered to be stochastically independent, all covariances are equal to 0. Now we can check the two basic hypotheses for intersection and estimate their agreement in the same way as for two basic hypotheses based on the primitive mean/covariance matrix.

It should be noted here that the alternative ways of calculating the agreement result in values which usually are bigger than the values for the agreement that are calculated in the normal way. This is the reason why the normalization shown in Equation 9.24 becomes necessary.

9.2.3.3 Combination of Basic Hypotheses in Symbolic Representations

In this thesis, the specification of (uniform) value ranges and covariance matrices is limited to numerical continuous dimensions. Thus, in most cases the basic hypotheses have to be expected to consist of a single singleton, which of course can be combined to unions with the help of complex hypotheses. In this case, the whole approach reduces to the simplest form of a Dempster-Shafer combination of two basic mass assignments, where the concepts of hypotheses and conditional mass assignments are not required.

However, basic hypotheses specified in Symbolic Representations form a special case here. Consider, for example, two hypotheses for the location of the user in a Symbolic Representation for the entity *Location*. One basic hypothesis specifies the Symbolic Representation '*University of Kassel WA73*', the other one specifies '*Room 1305 WA73*'. Consequently, even if '*University of Kassel WA73*' corresponds to a singleton, from our conceptual point of view it may implicitly still represent the union of a number of other singletons, as e.g. '*Room 1305 WA73*', '*Room 1403 WA73*', '*Room 1605 WA73*', etc., which are all part of or located in the building '*University of Kassel WA73*'. Thus, in the fusion step we have to consider the relationships between the entities which are defined in the Ontology through the properties (or sub-properties of) *DSTintersectedWith* (meaning that $H_1 \cap H_2 \neq \emptyset$), *DSTincludes* (meaning that $H_1 \supseteq H_2$) and *DSTisIncludedIn* (meaning that $H_1 \subseteq H_2$). In the same way, the relationships among classes can be exploited if *Representative Individuals* (see Section 7.2.3) are involved in the specification of basic hypotheses. The consideration of these relationships/properties is also important when we calculate the measures of plausibility and belief.

9.3 Summary and Discussion

In this chapter, we have explained how the Dempster-Shafer Theory of Evidence can be used to fuse belief functions that are specified according to the Information Model presented in Chapter 7. The specification of the belief functions is based on basic hypotheses which may involve the primitives value range, uniform value range or mean/covariance matrix, on complex hypotheses which represent logical expressions of the basic hypotheses and on the assignment of belief masses to the hypotheses. We have shown that discounting of such belief functions can be performed at the hypotheses level only. The conjunctive combination of belief functions results in a conjunctive combination at the hypotheses level and in a combination of the involved hypotheses. Therefore, the hypotheses form in a way a set of 'high-level focal elements' enabling the application of the Dempster-Shafer Theory of Evidence even if continuous scales are involved. However, the basic hypotheses must not be considered as singletons in this interpretation. Instead, basic hypotheses can have intersections with other basic hypotheses without being equal to them, and their combination can result in further basic hypotheses.

A major problem arises with the simplification of complex hypotheses in the fusion process because the distributive law does not hold for the combination of bodies of evidence. As the primitives used for the definition of the belief functions result in conditional mass assignments that assign masses only to singletons or to only one union of singletons, we have been able to derive new simplification rules similar to the distributive law that still allow a simplification in a number of cases.

We only use a few primitives for the definition of belief functions. This facilitates their compact representation but also limits the expressiveness, which prevented us, for example, to derive new simplification rules if negated bodies of evidence are involved. This is because the result cannot be represented in terms of the introduced primitives anymore. In our future work, we therefore will investigate other possibilities for representing the belief functions. For example, the approach of *linear belief functions* [77] could be a viable alternative here.

In this chapter, we have extensively discussed the combination of basic hypotheses and have proposed methods to calculate the conflict consistently over all involved combination steps. In particular, the problem has been recognized that different scales may result in different conflict values if mass densities are involved, and an alternative method for the calculation of the conflict/agreement has been presented.

Furthermore, with the presented approach we are only able to fuse belief functions that are specified for single concrete entities, i.e. the different hypotheses represent different possible states of the same entity. It is currently not possible to fuse belief functions whose hypotheses may represent state estimates for a set of entities. Here, the data association problem [8] has to be solved before the actual fusion step, which is out of the scope of this thesis.

If the Dempster-Shafer Theory of Evidence is utilized, scalability and performance are always important issues, as it works on the power set of a frame of discernment. Here too, limiting the expressiveness, basing the definition of belief functions on only a few primitives and considering only the focal elements can help to reduce the computational costs. A detailed discussion on scalability and concrete performance measurements are presented in Part III.

In our approach, information fusion is always performed based on the Representation requested by the corresponding information consumer. However, if IROs are involved that convert between Representations of different granularity, it might be useful to perform the fusion step in a different Representation. Furthermore, the information of all matching information providers is fused. We assume that the quality of the information improves with the number of redundant information sources. However, this cannot be expected to be always true, in particular if sensors provide misleading information under certain circumstances although their reliability is assumed to be high. Here, a conflict measure could provide some hints for the selection of an appropriate subset of information providers to be used in the fusion step.

10 Reasoning with Uncertain and Imprecise Information

In the previous chapters it was explained how Dempster-Shafer belief functions are specified according to our information model, propagated across non-linear transformations and competitively fused. This chapter focuses on the question how belief functions can be incorporated in reasoning schemes known from traditional probability theory and in logic-based reasoning.

10.1 Transferable Belief Model and Probabilistic Reasoning Schemes

Probabilistic reasoning schemes like *Naïve Bayes Classifiers*, *Hidden Markov Models* or *Bayesian Networks* have successfully been applied in a number of application domains, and Pearl's belief propagation algorithm describes an inference method for these reasoning schemes based on a message passing protocol. The *Transferable Belief Model* [130] and reasoning in *Directed Evidential Networks* [161] build the corresponding counterparts from the perspective of Dempster-Shafer belief functions. Although this provides a general method for using the well-known reasoning schemes with Dempster-Shafer belief functions, it is often either not recognized or not emphasized in literature. In [114], for example, the authors introduce the so-called credal HMM (CrHMM) , which is an enhancement of an HMM to Dempster-Shafer Theory of Evidence. The Transferable Belief Model and reasoning in Evidential Networks are referenced several times in [114], but HMMs are not really viewed from the perspective of a simple *Dynamic Bayesian Network* and a corresponding message passing scheme. For this reason, we present a direct mapping of Pearl's belief propagation algorithm to its DST-based counterpart for Directed Evidential Networks in this section. Using this theoretical foundation, DST-based reasoning with Naïve Bayes Classifiers, Hidden Markov Models and Polytree Bayesian Networks can easily be performed.

The core of Pearl's belief propagation algorithm is given in Equation 4.5 to Equation 4.9. In the following paragraphs, we will review these equations from the perspective of DST and the Transferable Belief Model.

Equation 4.5 states that the belief at a node X is calculated by multiplication of the belief induced from its parent nodes ($\pi(X)$) and the belief induced by its child nodes ($\lambda(X)$). In probability theory the two independent pieces of evidence are combined by pointwise multiplication. In DST, such a combination of two independent pieces of evidence is performed with Dempster's Rule of Combination or the Conjunctive Rule of Combination if normalization is postponed or not desired. From this consideration, we get the following mapping:

$$P(X|E_{X^+}, E_{X^-}) = \alpha\pi(X)\lambda(X) \quad \longrightarrow \quad m_X = m_{\pi_X} \oplus m_{\lambda_X} \tag{10.1}$$

The belief induced by the child nodes of X, i.e. $\lambda(X)$, results from the multiplication of the λ-messages and thus from the combination of the independent pieces of evidence received from the child nodes. Therefore, we have to replace the multiplications by Dempster's Rule of Combination or by the Conjunctive Rule of Combination also in Equation 4.7. This yields to the mapping:

$$\lambda_X(x) = \prod_{j=1}^{c} \lambda_{Y_j X}(x) \quad \longrightarrow \quad m_{\lambda_X} = \bigoplus_{j=1}^{c} m_{\lambda_{Yj\to X}} \tag{10.2}$$

where $m_{\lambda_{Yj\to X}}$ denotes the λ-message sent from node Y_j to node X.

The belief induced on X by its parent nodes, i.e. $\pi(X)$, is calculated by applying the law of total probability and exploiting the conditional independence with respect to the different parent nodes:

$$\pi_X(x) = \sum_{u_1,\ldots,u_p} P(X = x|u_1,\ldots,u_p) \prod_{j=1}^{p} \pi_{U_j X}(u_j) \tag{10.3}$$

The counterpart of the law of total probability in DST is given by (see also Equation 3.15):

$$m_{1,2}(A) = \sum_{B\subseteq\Omega} m_1(A|B)\, m_2(B) \tag{10.4}$$

Thus, utilizing a conditional belief function instead of a conditional probability function we can derive the mapping for Equation 10.3 as:

$$m_{\pi_X}(x) = \sum_{u_1\subseteq\Omega^{U_1},\ldots,u_p\subseteq\Omega^{U_p}} m_X(X = x|u_1,\ldots,u_p) \prod_{j=1}^{p} m_{\pi_{Uj\to X}}(u_j) \quad \forall x \subseteq \Omega^X \tag{10.5}$$

where $m_{\pi_{Uj\to X}}$ is the π-message sent from the parent node U_j to the node X. In traditional Bayesian networks, the π-message a node X sends to its child node Y_j is calculated as (see also Equation 4.8):

$$\pi_{XY_j}(x) = \alpha\pi_X(x) \prod_{k\neq j} \lambda_{Y_k X}(x) \tag{10.6}$$

which is the multiplicative combination of the belief induced by its parent nodes, i.e. $\pi_X(x)$, and the multiplicative combination of the belief induced by its other child nodes, i.e. $\prod_{k\neq j} \lambda_{Y_k X}(x)$. In the same way as above, here the multiplication has to be replaced by Dempster's Rule of Combination or the Conjunctive Rule of Combination as well. This yields the following mapping for Equation 10.6:

$$m_{\pi_{X\to Yj}} = m_{\pi_X} \oplus \left(\bigoplus_{k\neq j} m_{\lambda_{Yk\to X}} \right) \tag{10.7}$$

Most difficult to provide is the mapping for the equation that allows the calculation of the λ-messages (see also Equation 4.9):

$$\lambda_{XU_i}(u_i) = \beta \sum_x \lambda_X(x) \sum_{u_1,\dots u_{i-1},u_{i+1},\dots,u_p} P(X = x|u_1,\dots,u_p) \prod_{k\neq i} \pi_{U_kX}(u_k) \tag{10.8}$$

Thus, we carefully inspect the right side of the equation first. Here, it can be observed that the term

$$\sum_{u_1,\dots u_{i-1},u_{i+1},\dots,u_p} P(X = x|u_1,\dots,u_p) \prod_{k\neq i} \pi_{U_kX}(u_k) = P^{\pi}(X = x|u_i) \tag{10.9}$$

corresponds to the likelihood of u_i with respect to $X = x$ if all evidences from the other parent nodes of X have been incorporated, which we denote as $P^{\pi}(X = x|u_i)$. In general, belief is propagated from a child node to its parent node using the Bayesian theorem, resulting in the combination of the likelihood with the a priori belief, which in this case is $\pi_X(x)$, and a normalization. Viewing it from the perspective of DST and the TBM, we have to realize the belief propagation by applying the Generalized Bayesian Theorem [132]. Here, we can exploit the fact that it holds:

$$pl_{\Omega}(x|\theta) = pl_{\Theta}(\theta|x) \quad \forall \theta \subseteq \Theta, x \subseteq \Omega \tag{10.10}$$

Applying Equation 3.15, these considerations yield the λ-messages:

$$pl_{\lambda_{X\to Ui}}(u_i) = \sum_{x\subseteq\Omega^X} m_{\lambda_X}(x) \cdot pl^{\pi}_{U_i}(u_i|x) = \sum_{x\subseteq\Omega^X} m_{\lambda_X}(x) \cdot pl^{\pi}_X(x|u_i) \quad \forall x \subseteq \Omega^X, u_i \subseteq \Omega^{U_i} \tag{10.11}$$

where for the conditional plausibilities it holds:

$$pl^{\pi}_{U_i}(u_i|x) = pl^{\pi}_X(x|u_i) = \sum_{1\leq j\leq p, j\neq i, u_j\subseteq\Omega^{U_j}} pl_X(x|u_1,\dots,u_p) \prod_{k\neq i} m_{\pi_{Uk\to X}}(u_k) \tag{10.12}$$

Now we have mappings for all the core equations of Pearl's belief propagation algorithm. The general message passing protocol remains the same, but still the initialization step remains to be revisited. Observations are considered by providing a corresponding belief function m_{e_X} for the observed node.

- observed nodes without parents: m_{π_X} is initialized with m_{e_X}.
- observed nodes without child nodes: m_{λ_X} is initialized with m_{e_X}.
- not observed nodes without parents: m_{π_X} is initialized with the a priori belief of X, m_{X_0}, if available or the vacuous belief function otherwise, i.e. $m_{\pi_X}(\Omega^X) = 1$.

- not observed nodes without child nodes: m_{λ_X} is initialized with the vacuous belief function, i.e. $m_{\lambda_X}(\Omega^X) = 1$.

Special care has to be taken, if direct evidence can also be gathered from observations for nodes which are not root nodes and also not leaf nodes of the polytree. Here, the evidence has to be propagated in the π-messages and the λ-messages, as well. However, initializing m_{π_X} and m_{λ_X} with the belief function reflecting the observation would yield to wrong results as the conjunctive combination of belief functions is not idempotent. Thus, in this case we have to slightly adjust Equation 10.2 and Equation 10.7 for these nodes in order to also consider the evidence m_{e_X} resulting from the direct observation:

$$m_{\lambda_X} = m_{e_X} \oplus \left(\bigoplus_{j=1}^{c} m_{\lambda_{Yj \to X}} \right) \tag{10.13}$$

$$m_{\pi_{X \to Yj}} = m_{e_X} \oplus m_{\pi_X} \oplus \left(\bigoplus_{k \neq j} m_{\lambda_{Yk \to X}} \right) \tag{10.14}$$

It is also noteworthy that Bayesian networks learned using pure probability theory can be extended for the integration of DST-based observations represented as belief functions. For this purpose, only the a priori probability functions or vectors for the root nodes and the conditional probability tables have to be enhanced. The enhancement of the probability vector to a Dempster-Shafer belief function is trivial: just set the probabilities as masses for the singleton elements and set all other masses to zero. In the same way, the conditional belief function $m(x|u)$ with $x \subseteq \Omega^X$ can be derived for all singletons u, i.e. $u \in \Omega^U$, from the conditional probability function/vector $P(x|u)$. Then all other conditional belief functions $m(x|U)$ with $x \subseteq \Omega^X$, $U \subseteq \Omega^U$ and $|U| > 1$ can be constructed using the Disjunctive Rule of Combination.

In the previous paragraphs, we have shown how Pearl's belief propagation algorithm for Polytree Bayesian Networks can be mapped to its DST/TBM-based counterpart. Its application to a DST-based version of a Naïve Bayes Classifier is illustrated in Figure 10.1.

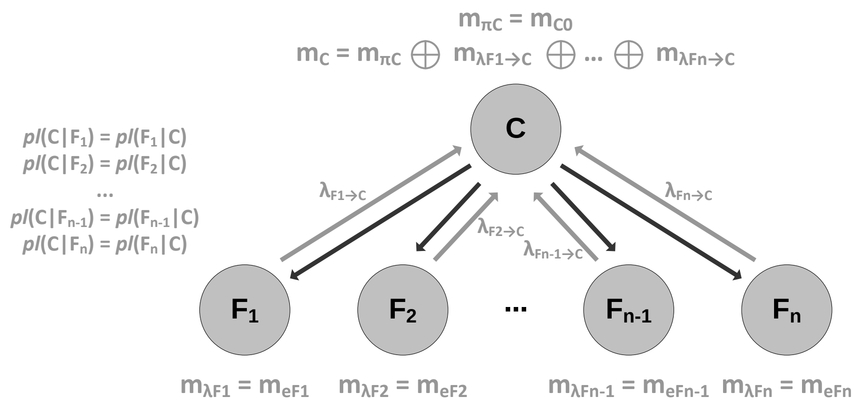

Figure 10.1: Naïve Bayes Classifier from the Perspective of the DST

First, m_{π_C} is initialized with the a priori belief of C, i.e. m_{C_0}, and the observations of the features F_1 to F_n are integrated through the initialization of $m_{\lambda_{Fi}}$ with $m_{e_{Fi}}$. Then, all feature nodes send their λ-messages to C, which are combined with m_{π_C} using *Dempster's Rule of Combination* to calculate the a posteriori belief m_C of node C.

As already mentioned in Section 4.1.3, classification of observation sequences with Hidden Markov Models can be realized by calculating the probability for each of the HMMs under consideration to generate the observation sequence and choosing the HMM with the maximum probability. For this purpose, the α variables, i.e. the α_t, are calculated for each time step and the normalization factors c of the α-variables reflect how well the a priori estimate of the state probability function at a particular time step fit with the corresponding observation at this time step. In HMM theory [113], the α-variables are also called *forward* variables and actually correspond to the π-messages sent from a node to its successor node. Figure 10.2 illustrates, how these π-messages can be calculated applying the mapping of Pearl's belief propagation algorithm to a DST/TMB-based version of a HMM.

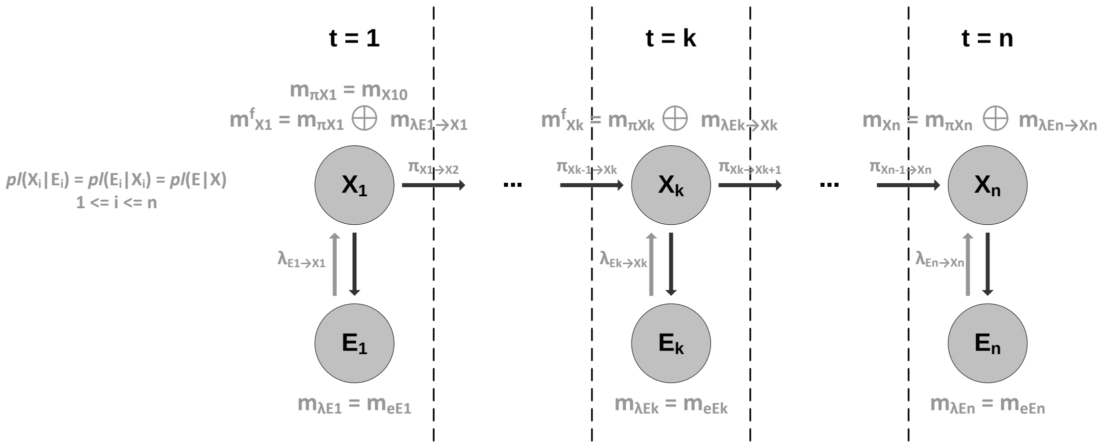

Figure 10.2: Hidden Markov Model from the Perspective of the DST

At time step X_1, $m_{\pi_{X_1}}$ is initialized with the a priori belief for the states of the HMM, i.e. $m_{X_{10}}$. Updating m_{π_X} with the λ-message $m_{\lambda_{E_1 \to X_1}}$ that reflects the induced belief from the observation using Dempster's Rule of Combination yields the π-message (or forward message) $m_{\pi_{X_1 \to X_2}} = m^f_{X_1}$ sent from node X_1 to node X_2. In the same way, the π-messages for all other time steps are calculated by updating $m_{\pi_{X_k}}$ with the λ-message $m_{\lambda_{E_k \to X_k}}$. At time step n, the forward message corresponds to the a posteriori belief m_{X_n} of X_n. For calculating the counterparts for the c-factors in the DST/TMB-based version of the HMM, we have to utilize the Conjunctive Rule of Combination instead of Dempster's Rule of Combination, in order to prevent an implicit normalization by applying the combination rule. The c-factor for a particular time step can then be calculated as:

$$c_t = \sum_{x \subseteq \Omega^X, x \neq \emptyset} m^f_{X_t}(x) \tag{10.15}$$

Just as in the probability-based version of the HMM, the factor c_t is used to normalize $m^f_{X_t}$ and reflects how well the evidence induced by the observation fits the a priori belief of the states of the HMM. It is also noteworthy that if the belief functions for the a priori beliefs and for the observations happen to be probability functions, all DST/TBM-based models reduce to its probability-based counterpart.

10.2 Logic-Based Reasoning with Uncertain Information

Apart from probabilistic reasoning schemes such as Naïve Bayes Classifiers, Hidden Markov Models or Bayesian Networks, reasoning based on formal logics is often applied to infer new high-level information from low-level sensor data. However, such approaches typically rely on precise and complete input data and cannot handle uncertainty and partial or total ignorance of their inputs. Thus, in this section we will briefly present how Dempster-Shafer belief functions can be incorporated in an evaluator for first-order logic formulas. As logic-based reasoning with Dempster-Shafer belief function is not in the focus of this dissertation, only an informal overview of the basic ideas for the evaluator is given here. A more formal description can be found in the Master's thesis of Triller [139] that has been co-supervised as part of the work for this dissertation.

In [139], the evaluator is implemented in Prolog [13] and is able to evaluate first-order logic formulas in clause form. The basic idea is to evaluate predicates not only to either *true* or *false* but also to *unknown* $= \{true, false\}$ and to provide mass assignments that reflect our belief in the truth value of the predicates. Prolog relies on resolution techniques [120] to evaluate clauses. Just as in standard resolution, variables may be bound during predicate evaluation. Given a predicate $P(X_1, \ldots X_m, Y_1, \ldots Y_n)$ with free variables $X_1, \ldots, X_m$ and bound variables $Y_1, \ldots, Y_n$, the evaluation of the predicate yields the conditional mass assignment

$$m_P \left(\begin{array}{c} true \\ false \\ unknown \end{array} \middle| \, X_1, \ldots, X_m, Y_1, \ldots, Y_n \right)$$

that reflects the belief in the truth value of the predicate in the context of the assignments for the free and also the bound variables. In addition, the binding of the free variables in the evaluation process may result in a conditional mass assignment

$$m_P \left(X_1, \ldots, X_m | Y_1, \ldots, Y_n \right)$$

that denotes the belief for the assignment of the variables $X_1, \ldots, X_m$ during the evaluation of predicate P in the context of the already bound variables $Y_1, \ldots, Y_n$. Assume, for example, a predicate $UserLoc(P, X)$ that binds the free variable P with the different hypotheses for the location of the user X (already bound variable). The predicate is then evaluated to *true* for all valid hypotheses for P with mass 1 and the mass assignment $m_{UserLoc}(P = p | X = x)$ denotes the belief mass that the location of user x is p, where $p \subseteq \Omega^P$. Here, $m_{UserLoc}(P = \Omega^P | X = x)$ is the belief mass for the unknown location of user X, which allows to handle total ignorance about the user's location. However, it is important to note here, that not all predicates may return such masses for the assignment of free variables in the context of bound variables. Exceptions are predicates that bind free variables not with different hypotheses for the assignment but instead with different available alternatives for which no mass assignment can be provided. An example for such a predicate is *User(X)*[1] that binds the variable X with the different available users in a ubiquitous computing scenario. A mass assignment $m_{User}(X = x)$ would express the belief that we assume x with $m_{User}(X = x)$ to be the user X, which is not appropriate if we would like to evaluate the predicate for all users independently.

If a conditional mass assignment for the binding of free variables in the context of the already bound variables is available, it can be used for *de-conditioning* the conditional mass assignment for the truth value of the predicate:

$$m_P\left(\begin{array}{c} true \\ false \\ unknown \end{array} \middle| Y_1, \ldots, Y_n \right) = \tag{10.16}$$
$$\sum_{x_1 \subseteq \Omega^{X_1}, \ldots, x_m \subseteq \Omega^{X_m}} m_P\left(\begin{array}{c} true \\ false \\ unknown \end{array} \middle| x_1, \ldots, x_m, Y_1, \ldots, Y_n \right) \cdot m(x_1, \ldots, x_m | Y_1, \ldots, Y_n)$$

If appropriate mass assignments are available for the bound variables $Y_1, \ldots, Y_n$ from previous predicate evaluations, these can be used to perform the de-conditioning of the conditional mass assignment for the truth value of the predicate with regard to the bound values just in the same way.

In order to allow evaluation of first-order logic formulas involving a number of predicates, we also have to clarify how the belief masses for the truth values of conjunctions, disjunctions and negations of predicates can be calculated. For this purpose, we define the $\wedge$-operator for a conjunction of two predicates through the following matrices:

$$mc_{true} = \begin{bmatrix} 1 & 0 & 0 \\ 0 & 0 & 0 \\ 0 & 0 & 0 \end{bmatrix} \quad mc_{false} = \begin{bmatrix} 0 & 1 & 0 \\ 1 & 1 & 1 \\ 0 & 1 & 0 \end{bmatrix} \quad mc_{unknown} = \begin{bmatrix} 0 & 0 & 1 \\ 0 & 0 & 0 \\ 1 & 0 & 1 \end{bmatrix}$$

The mass assignment for a conjunction of two predicates P_1 and P_2 in the context of the variables $X_{1..m}, Y_{1..n}$ can then be calculated as

$$\begin{aligned} m_{P_1 \wedge P_2}(true | X_{1..m}, Y_{1..n}) &= m_{P_1}(.|X_{1..m}, Y_{1..n})^T * mc_{true} * m_{P_2}(.|X_{1..m}, Y_{1..n}) \\ m_{P_1 \wedge P_2}(false | X_{1..m}, Y_{1..n}) &= m_{P_1}(.|X_{1..m}, Y_{1..n})^T * mc_{false} * m_{P_2}(.|X_{1..m}, Y_{1..n}) \\ m_{P_1 \wedge P_2}(unknown | X_{1..m}, Y_{1..n}) &= m_{P_1}(.|X_{1..m}, Y_{1..n})^T * mc_{unknown} * m_{P_2}(.|X_{1..m}, Y_{1..n}) \end{aligned}$$

where $m_{P_1}(.|X_{1..m}, Y_{1..n})$ and $m_{P_2}(.|X_{1..m}, Y_{1..n})$ are considered as vectors of the masses for P_1 and P_2 to be *true*, *false* and *unknown* and $*$ denotes the standard vector/matrix multiplication.

[1]This predicate has no bound variable as input.

The $\vee$-operator for a disjunction of two predicates is defined through the matrices:

$$md_{true} = \begin{bmatrix} 1 & 1 & 1 \\ 1 & 0 & 0 \\ 1 & 0 & 0 \end{bmatrix} \quad md_{false} = \begin{bmatrix} 0 & 0 & 0 \\ 0 & 1 & 0 \\ 0 & 0 & 0 \end{bmatrix} \quad md_{unknown} = \begin{bmatrix} 0 & 0 & 0 \\ 0 & 0 & 1 \\ 0 & 1 & 1 \end{bmatrix}$$

which are applied to the mass vectors of two predicates in the same way as shown above for the conjunction. The mass assignment for a negation of a predicate P is defined as:

$$\begin{aligned} m_{\neg P}(true|X_{1..m}, Y_{1..n}) &= m_P(false|X_{1..m}, Y_{1..n}) \\ m_{\neg P}(false|X_{1..m}, Y_{1..n}) &= m_P(true|X_{1..m}, Y_{1..n}) \\ m_{\neg P}(unknown|X_{1..m}, Y_{1..n}) &= m_P(unknown|X_{1..m}, Y_{1..n}) \end{aligned}$$

If a free variable can only be assigned with a finite number of alternatives, the existential quantor ($\exists$) can be expressed as a disjunction over all alternatives for the corresponding variable and the matrices md_{true}, md_{false} and $md_{unknown}$ can be used to derive the mass assignment for the truth-values of the formula. In the same way, the universal quantor ($\forall$) can be replaced by a conjunction over all alternatives for a variable that has only a finite number of possible assignments. The matrices mc_{true}, mc_{false} and $mc_{unknown}$ can be applied to derive the mass assignment for the truth-value of the formula.

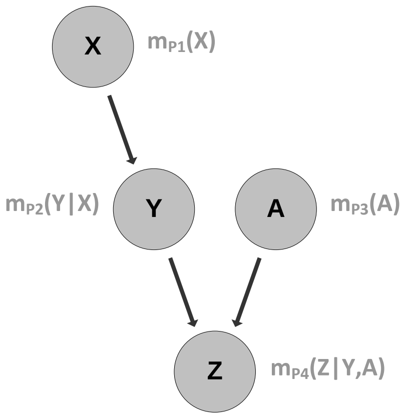

Figure 10.3: Conditional Variable Dependencies Represented as Directed Acyclic Graph

In Prolog, predicates are defined via clauses, i.e. disjunctions of literals typically written as

$$Head \leftarrow Body$$

where *Head* is the predicate and *Body* is a conjunction of literals. This means that the evaluation of a predicate involves the evaluation of all the predicates contained in the *Body* of the clause. Special care has to be taken if variables appear in the *Body* of a clause but not in its *Head*. Assume, for example, the clause

$$Pred(Y) \leftarrow P_1(X) \wedge P_2(X, Y) \wedge P_3(A) \wedge P_4(A, Y, Z)$$

As result of the evaluation we expect $m_{Pred}(.|Y)$ and $m_{Pred}(Y)$, but the *Body* of the clause is evaluated in the context of all variables appearing in it, i.e. X, Y, A and Z, which results in $m_{Pred}(.|X,Y,Z,A)$. Besides, the variable Y is bound during the evaluation of P_2 depending on the variable X, which has already been bound during the evaluation of P_1. Thus, we get a conditional mass assignment $m_{P_2}(Y|X)$ as result of the evaluation of P_2. In order to retrieve the expected mass assignments $m_{Pred}(.|Y)$ and $m_{Pred}(Y)$ we have to perform an appropriate de-conditioning of $m_{Pred}(.|X,Y,Z,A)$ and $m_{P_2}(Y|X)$. For this purpose, we represent the conditional dependencies of the variable bindings as a directed acyclic graph similar to a Bayesian network, which is shown in Figure 10.3 for our example predicate.

Exploiting the conditional independence of variables represented in this directed acyclic graph, we can now process the graph from bottom up until a variable is reached, which should be kept, i.e. in our case variable Y. First, we utilize $m_{P_4}(Z|Y,A)$ in the same way as done in Equation 10.16 in order to derive $m_{Pred}(.|X,Y,A)$ from $m_{Pred}(.|X,Y,Z,A)$. Next, we use $m_{P_3}(A)$ to get $m_{Pred}(.|X,Y)$ from $m_{Pred}(.|X,Y,A)$. Now, just de-conditioning with regard to the variable X has to be performed in order to get the desired result. For this purpose, the graph is further processed bottom up but variable Y is left out. This means that we derive $m_{Pred}(.|Y)$ from $m_{Pred}(.|X,Y)$ by utilizing $m_{P_1}(X)$. In the same way, $m_{P_1}(X)$ can be exploited to get $m_{Pred}(Y)$ from $m_{P_2}(Y|X)$.

10.3 Discussion

As demonstrated with our case study on user activity recognition using Hidden Markov Models (see Chapter 13) and by the Master's thesis of Triller [139], the reasoning approaches presented in the previous section show promising results if the involved (random) variables have a quite small number of possible assignments, or more precisely, if the corresponding frames of discernment for the Dempster-Shafer belief mass assignments are small. In general, the Dempster-Shafer Theory of Evidence works on the power sets of the frame of discernments, the size of which is exponential with the number of elements in the frame of discernments. Thus, problems already arise, for example, if a complete conditional mass assignment of a variable X is needed, under the condition that evidence for a variable Y has been collected and the corresponding frames of discernment Ω^X and Ω^Y have 10 elements each. Then the complete *conditional mass table* has $2^{10} \cdot 2^{10} \approx 10^6$ entries.

It is quite obvious that integration of complex belief functions on continuous variables and involving hypotheses based on the primitives mean/covariance matrix, value range and uniform value range that may result from the methods presented in the previous chapters, might be difficult to handle in the presented reasoning schemes. Here, approaches are essential that reduce the complexity of the belief functions.

One possibility is to abstract hypotheses on continuous variables just by their mean, as it is also done in the Master's thesis of Triller [139]. This work shows that with this abstraction feasible results can be obtained as still the uncertainty among the different hypotheses can be considered. However, all information about the impreciseness of the involved hypotheses is lost, which may lead to inaccurate results. A problem also arises if complex hypotheses have to be processed that involve intersection of basic hypotheses based on the primitives mean/covariance matrix and (uniform) value range, which cannot be combined immediately. Here, the abstraction by the corresponding mean values cannot be expected to yield feasible results.

In order to avoid this situation and also to allow considering the impreciseness of hypotheses to some extent, it is possible to approximate the hypotheses and their intersection by a number of smaller hypotheses, which are then abstracted by their mean values again. However, the quality of the approximation has to be traded against the number of new hypotheses, which determines the size of the new frame of discernment.

For some reasoning tasks, approaches similar to the handling of belief functions within IROs can be applied. Assume, for example, a predicate $NearObject(O, X)$ that evaluates to true if the object O is within a certain distance, e.g. 1000mm, to the user X. In this case, a kind of *'IRO'* can be used to transform the object location to a representation $D = dist(X, O) - 1000$. From this representation, the belief masses m_{NO} for the truth values of the $NearObject(O, X)$ predicate can easily be derived. $m_{NO}(true)$ is assigned with $bel_D(D \leq 0)$, $m_{NO}(false)$ with $bel_D(D > 0)$ and $m_{NO}(unknown)$ with $1 - m_{NO}(true) - m_{NO}(false)$. $bel_D(D \leq 0)$ and $bel_D(D > 0)$ are easy to determine even if the primitive mean/covariance matrix is involved. In this case, computing the beliefs can be reduced to calculating $P_D(D \leq 0)$ and $P_D(D > 0)$ with P_D being the probability density function of the one-dimensional Gaussian distribution that represents the hypothesis.

From the discussion above, it becomes quite obvious that the selection of an appropriate approach for reducing the complexity of the belief functions highly depends on the required accuracy and on the applied reasoning schemes and thus on the concrete application. In this section, we have presented a number of ideas but no general methodology to tackle this problem. However, reduction of complexity is the predominant issue with regard to the integration of Dempster-Shafer belief function in reasoning schemes as presented in this chapter and has to be investigated further in future work, in order to come up with a general method applicable to a wide range of applications.

11 Architectural Implications

The combination of the theoretical concepts proposed in the previous chapters to a comprehensive framework for information exchange and fusion leads to a number of implications at the architectural level. As we provide a generic method applicable to a wide range of applications, however, we will not propose a concrete architecture which is grounded on particular technologies, representations, and programming models, but only identify the main architectural elements and their relationships. Here, we have been inspired by the FIPA Abstract Architecture Specification (FIPA-AAS) [37]. In a similar way as the FIPA-AAS, we only provide an architectural description at an abstract level, which is intended to serve as a guideline for developers to design the architecture for a concrete framework.

Concrete realizations of the architecture may adopt the technologies, representations and programming models that are most suitable for the respective application domain. Furthermore, they may show differences in the distribution of the functionalities over the nodes of the computing environment and functionalities may be realized in a centralized or in a decentralized manner. For example, autonomous mobile robots that have to cooperate in a highly dynamic environment and need to react quickly on changes in the environment may be equipped with their own components for most of the architectural elements in order to avoid communication overhead. For mobile devices with very limited resources, however, this is not feasible. Instead, functionalities have to be deployed on a single or multiple other nodes and to be made available through the respective communication framework.

11.1 Main Architectural Elements and their Relationships

In this section, we define the abstract architecture by identifying the main architectural elements, their functionalities and their relationships to other elements. Here, we have been guided by the experiences from the implementation of the prototype framework that forms the basis for our real-world case study described in Chapter 12 as well as by experiences from our work on the MUSIC context middleware [106, 107, 116]. The latter one utilizes the Information Model presented in Chapter 7 as underlying context modelling approach.

Figure 11.1 shows our abstract architecture with regard to the three layers of abstraction (Conceptual Layer, Exchange Layer and Functional Layer) as introduced in Section 7.1. The architectural elements at the Conceptual Layer are mainly concerned with managing the knowledge captured in the ontology and making it available for the Discovery Service and Matching Service. The Exchange Layer consists of the underlying Communication Infrastructure and the Mediation Engine that establishes the links between Information Providers, Consumers, and Reasoners. The Mediation Engine is also part of the Functional Layer, as it is responsible for conversions between different representations. At the Functional Layer, all the Information Providers, Consumers and Reasoners are located, which are connected to the distributed environment through the Communication Infrastructure. It

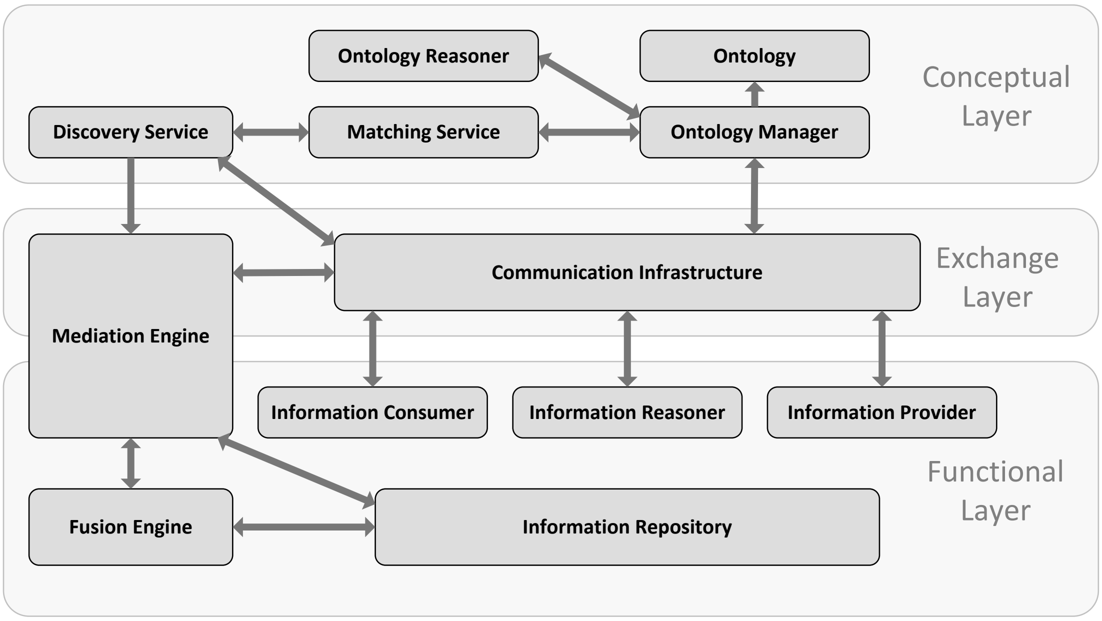

Figure 11.1: Main Elements of the Abstract Architecture

also contains the Fusion Engine and Information Repository. The Fusion Engine provides support for competitive information fusion; the Information Repository facilitates caching of information in order to allow processing of information with matching timestamps. In the following list, the main architectural elements are described in more detail. It starts with the elements at the Conceptual Layer and finishes with the elements at the Functional Layer.

- **Ontology Manager**: The Ontology Manager is responsible for parsing the Ontology and storing all contained information in data structures that facilitate fast processing of queries for the matching of information offers and requests and enable fast reasoning about possibly required mediation tasks. In addition, it may incorporate a general purpose Ontology Reasoner, as e.g. Pellet [128] or Fact++ [140], in order to be able to support general inference tasks, which may not be related to discovery, matching or mediation.

  Relation to other elements:

  - Requires access to the *Ontology*
  - Allows the *Matching Service* to query for information captured in the *Ontology*
  - Establishes a connection to a general purpose *Ontology Reasoner*
  - Provides an interface accessible through the *Communication Infrastructure* to accept queries for general ontology inference

- **Discovery Service**: The Discovery Service enables Information Providers, Information Consumers, and Information Reasoners to register their information offers and requests via broadcast messages or a similar mechanism. As part of the registration messages, which are defined in a platform- and protocol-independent fashion using the Information Offer and Request Language (IORL), the Information Providers, Consumers and Reasoners also provide information about how they can be reached. Later on, this information can be exploited for establishing channels in a peer-to-peer manner. If the registration messages are periodically sent in certain time intervals, they can also serve as a kind of alive message facilitating the detection of disappearing services.

All reasoners in the environment immediately register their information requests after start-up but only advertise their information offers if all their information requests are fulfilled. For reasoning about matching offers and requests, the Discovery Service contacts the Matching Service and grants access to all currently registered information offers and requests. As a result, for each of the information requests that can be fulfilled a Request Result data structure is returned by the Matching Service that contains information about matching information offers and possibly required mediation tasks. The Request Results are then forwarded to the Mediation Engine. If there is a change in the advertised information offers and requests, the Discovery Service contacts the Matching Service in order to update the corresponding Request Result data structures and propagates the changes to the Mediation Engine.

Relation to other elements:

– Allows *Information Providers*, *Consumers* and *Reasoners* to register information offers and requests via the underlying *Communication Infrastructure*

– Contacts the *Matching Service* to find matching information offers and requests and retrieves the corresponding *Request Result* data structures

– Forwards the *Request Result* data structures to the *Mediation Engine*

- **Matching service**: The Matching Service retrieves a set of information offers and a set of information requests from the Discovery Service and is responsible to find possible matchings. For this purpose, it uses the information about the involved entities, scopes and available IROs contained in the Ontology by querying the Ontology Manager. If the constraints on entities and scopes can be fulfilled, the Matching Service also tries to find an appropriate chain of IROs if there are mismatches in the requested and provided representations. As part of this task, it is not only necessary to check if appropriate IROs are defined in the ontology but also if they are available as concrete realization, i.e. as service, library method, etc. Besides, IROs may have additional dependencies to other information. Thus, it must be checked how information requests resulting from these additional dependencies can be fulfilled. As result, a Request Result data structure is returned to the Discovery Service for each of the information requests for which matching offers can be found.

  Relation to other elements:

  – Retrieves the registered information offers and requests from the *Discovery Service* and returns *Request Result* data structures for the found matches to the *Discovery Service*

  – Requires access to the *Ontology Manager* to raise queries about the entities, scopes, representations and IROs defined in the *Ontology*

- **Communication Infrastructure**: The Communication Infrastructure provides the basic facilities required by the involved components, modules and services to communicate in the distributed computing environment. It offers broadcast protocols for advertising information offers and requests but also supports the establishment of peer-to-peer communication channels.

Relation to other elements:

- Utilized by all architectural elements that have to communicate with components, modules and services on remote nodes

- **Mediation Engine**: The Mediation Engine retrieves Request Result data structures from the Discovery Service. It is responsible to perform the possibly required IROs and to establish the links between the Information Consumers, Information Providers and Reasoners. If no mediation steps are required and only one Information Provider fulfills the request, the information is just forwarded to the Consumer/Reasoner. If there is more than one provider available, the information is fused competitively with the help of the Fusion Engine. If additional dependencies of IROs have to be resolved in a mediation step, also the corresponding mediation steps and potential fusion steps are performed for the additional dependencies. The Mediation Engine may utilize an Information Repository in order to ensure a matching of timestamps of the data needed in a mediation task.

  Relation to other elements:

  - Retrieves the *Request Result* data structures from the *Discovery Service*
  - Establishes links between *Information Consumers*, *Providers*, and *Reasoners* utilizing the *Communication Infrastructure*
  - Utilizes the *Fusion Engine* to perform competitive information fusion
  - Accesses the *Information Repository* in order to ensure that information used to resolve additional dependencies of IROs is aligned with regard to its timestamps

- **Fusion Engine**: The Fusion Engine is invoked by the Mediation Engine to perform competitive information fusion and returns the result of the fusion step to the Mediation Engine. In order to ensure a matching of timestamps of the data to be fused, the Information Repository is accessed and data is retrieved from its cache.

  Relation to other elements:

  - Is invoked by the *Mediation Engine* to perform the competitive information fusion tasks and returns the fused result
  - Accesses the *Information Repository*, in order to ensure that information to be fused is aligned with regard to its timestamps

- **Information Repository**: The Information Repository supports caching of information, in order to allow the combination and fusion of data that match with regard to their timestamps.

  Relation to other elements:

  - Provides access to cached information to the *Mediation Engine*
  - Provides access to cached information to the *Fusion Engine*

- **Information Provider**: An Information Provider advertises its offers to the Discovery Service via a broadcast protocol or similar mechanism utilizing the underlying Communication Infrastructure. In the advertising message also more detailed information is provided about the supported protocols, data serialization schemes and how the provided data can be accessed. For example, a port can be provided that accepts connections and publishes the data via these connections. Optionally, an interface could be provided that allows to register a callback method.

  Relation to other elements:

  - Registers its information offers with the *Discovery Service* utilizing the *Communication Infrastructure*
  - Provides information to *Consumers* and *Reasoners* through the *Communication Infrastructure* and the *Mediation Engine*

- **Information Consumer**: An Information Consumer registers its information needs with the Discovery Service. In the registration message also more detailed information is provided about the supported protocols, data serialization schemes and how the requested data can be retrieved. For example, a port can be provided that accepts connections and allows pushing of the data via these connections. Optionally, an interface could be provided that allows to register a callback method.

  Relation to other elements:

  - Registers its information requests with the *Discovery Service* utilizing the *Communication Infrastructure*
  - Retrieves information from *Providers* and *Reasoners* through the *Communication Infrastructure* and the *Mediation Engine*

- **Information Reasoner**: An Information Reasoner acts as Information Provider and Information Consumer at the same time. It registers its information needs with the Discovery Service and also provides information about how the requested information can be retrieved. If all information requests can be fulfilled, it also advertises its information offers to the Discovery Service and specifies how the provided data can be published.

  Relation to other elements:

  - Registers its information offers and requests with the *Discovery Service* utilizing the *Communication Infrastructure*
  - Retrieves information from *Providers* and *Reasoners* through the *Communication Infrastructure* and the *Mediation Engine*
  - Provides information to *Consumers* and *Reasoners* through the *Communication Infrastructure* and the *Mediation Engine*

The architectural elements described in the list above communicate via a number of data structures that are encapsulated in messages, as e.g. Information Offers, Information Requests, or Request Results. Information Offers and Requests are specified using the platform- and protocol-independent IROL introduced in Section 7.3. The Request Result data structure is also very important as it implicitly defines the links between Information

Providers and Consumers and captures the required mediation and fusion tasks. The following paragraph provides a detailed description of the Request Result data structure shown in Figure 11.2.

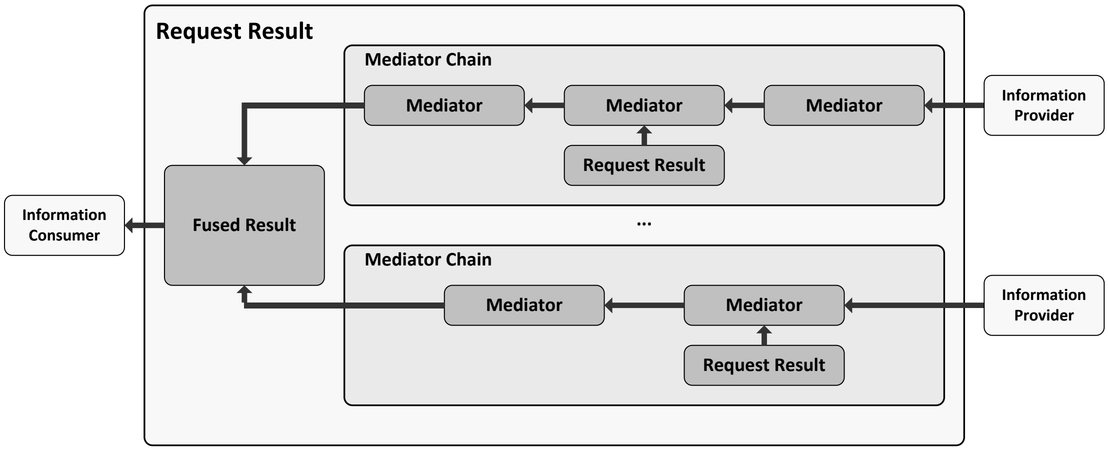

Figure 11.2: The Request Result Data Structure

Request Result: The Request Result is the central data structure that specifies a kind of runtime model of how information requests can be fulfilled by information offers. This possibly involves a number of mediation tasks and fusion steps. A Request Result contains a number of Mediator Chains whose results are competitively fused and then returned to the Information Consumer. Each Mediator Chain specifies a chain of Mediators that have to be invoked in order to cope with mismatches in the requested and provided representations or to extract a requested scope as part of a more comprehensive message. As a consequence, all IROs to be performed are encapsulated in Mediators. As each IRO may have additional dependencies to further information, a Mediator holds a Request Result data structure for each of its additional dependencies.

11.2 Discussion

In the architecture described above, Information Providers and Reasoners are assumed to periodically provide data to Information Consumers/Reasoners if there is a matching of the offered and requested information. Information offers and requests are specified using the IORL and information requests can also be interpreted as queries to a data management system. In this respect, our abstract architecture shows a number of similarities to what is commonly understood by *Data Stream Management Systems* [2].

We also assume in our approach that Information Reasoners always register their information requests with the Discovery Service regardless if the information they provide is actually requested by another Information Reasoner or Consumer. If all requests of an Information Reasoner are fulfilled, it also registers its information offer with the Discovery Service. This assumption simplifies checking whether information requests can be fulfilled, as only information offers are registered whose dependencies are resolved. Thus, reasoning can be performed on the set of currently registered information offers and requests without checking the dependencies of reasoners and involving a kind of recursive search. This approach has been inspired by the OSGi bundle lifecycle [99], where a bundle has to be in the state resolved, i.e. all its dependencies are fulfilled, before being started.

Furthermore, there is no mechanism envisaged in our architecture to select a subset of information providers from the set of available providers matching a request. Instead, the information of the whole set of providers is fused with the assumption, that the quality of the information improves with the number of redundant information sources. However, this may not always be true, in particular if sensors provide misleading information under certain circumstances although their reliability is expected to be high. Here, a conflict measure in the fusion of the different information sources could provide some hints for the selection of an appropriate subset. Selection could also be useful if resource consumption or monetary costs related to the use of information providers/reasoners have to be considered.

Part III

Evaluation

12 Case Study I: Heterogeneous Team of Autonomous Mobile Soccer Robots

12.1 Overall Description

In this chapter, we present a real-world case study centered around a dynamically composed team of autonomous soccer robots in the RoboCup environment. RoboCup [121] is an international joint project attempting to foster research in robotics, artificial intelligence, and related fields. Soccer is a well-understood, highly dynamic game where a wide range of technologies can be integrated and examined. The robots are completely autonomous, i.e. they have all the necessary sensors and control devices on board and must navigate autonomously without external control.

Currently, a RoboCup soccer team consists of 5 robots. For the long time goal, however, the number of players will approach 11. It may be difficult for research groups to keep up with the enlargement of team sizes. For newcomers it even constitutes a virtually infeasible financial effort. This is why so-called mixed teams gain a lot of popularity. Here, two or more research groups pool their resources together to provide a joint, more powerful team [91]. Although this means more effort and difficulties for the affected teams, it is assumed that this situation better reflects the reality also for future large-scale search and rescue scenarios, where it is expected that robots with different capabilities and independently developed by different companies have to dynamically compose to and cooperate in a team.

To realize team play, robots must be able to exchange information with their teammates, interpret the exchanged data, and fuse the information to a consistent world view. This is the basis for coming to an agreement about the current situation and coordinating the cooperative behavior of the team. Nowadays, almost every team in the RoboCup middle-size league implements some kind of team play, which in most cases is tailored to the capabilities and the needs of the underlying robotic software framework. However, the creation of mixed teams has proved to be a difficult and time-consuming task. This is due to the lack of standard software and the variety of involved software frameworks that are tailored to the specific research focus of the corresponding institutes. Although development frameworks like SPICA [6, 4, 5, 7] are able to establish a communication infrastructure incorporating a number of different platforms with minimal effort, there is still the problem of the heterogeneity of the data that have to be communicated, interpreted and fused. The different teams use their own representations, coordinate systems, measuring units, etc., most suitable for their world modelling and behavior control approach. Approaches to establish standard representations [143] have not gained popularity and cannot be expected to be adopted in near future.

In this case study, we show how team play of autonomous soccer robots can be realized incorporating imprecise, uncertain and unreliable information from heterogeneous sources. Here, we concentrate on coming to an agreement on the ball position on the field, which probably is the most important criterion for the organization of the team play. The communicated estimates for the ball position and the robots' positions are annotated with confidence values and with covariance matrices, in order to model the imperfect nature of sensor information. Furthermore, the robots can provide several hypotheses for the ball position.

For the case study, we utilize the robots of the Carpe Noctem RoboCup team of the University of Kassel [18] (see Figure 12.1). The corresponding communication infrastructure for team coordination is already based on the SPICA development framework, which is tailored to allow communication among heterogeneous software platforms. Therefore, the usage of effectively only one software platform does not affect the viability of our case study with regard to heterogeneity of involved software platforms. However, with regard to the communicated data the robots are artificially 'heterogenized'.

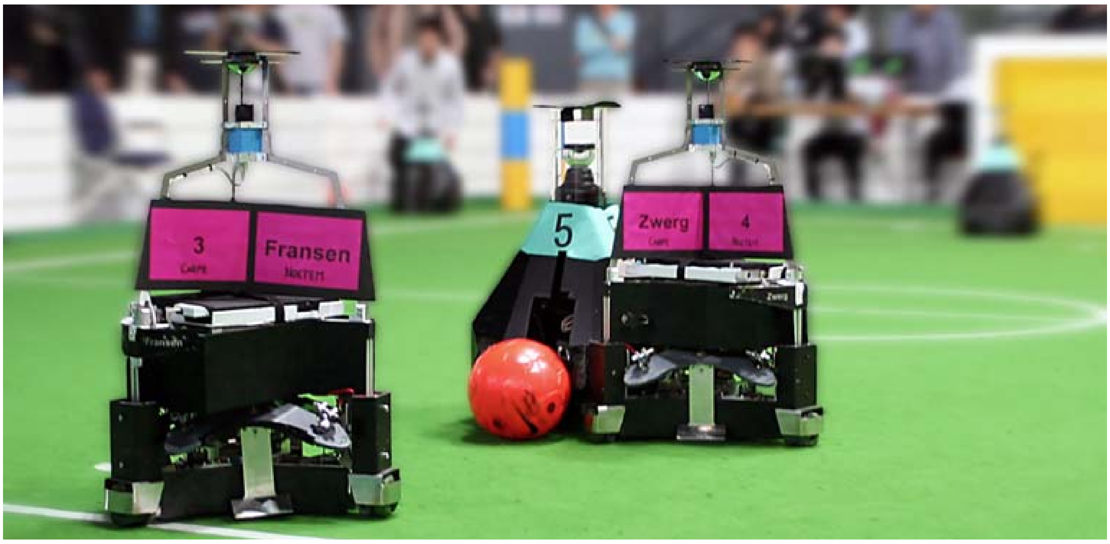

Figure 12.1: Two Carpe Noctem Robots Tackling an Opponent

In RoboCup, it is a common practice of almost all teams to establish at least two different coordinate systems for observed objects and estimated positions: a global coordinate system on the field and a coordinate system relative to the robot. Whereas the global coordinate system is used for organizing the team play, relative coordinates are a convenient representation for the realization of low-level behaviors of a robot, as e.g. approaching the ball. Two possible global coordinate systems, called *CNAllocentric* (a) and *CNAllocentricArt* (b), are shown in Figure 12.2. They differ in the direction of the axes and in the units of measurement.

In the same way as for the global coordinates, there exists an infinite number of possibilities for the relative coordinate systems. Two examples are shown in Figure 12.3. *CNEgocentric* (a) is a Cartesian coordinate system with origin in the center of the robot and x-axis along *Wheel Axis 1* of the robot. *CNEgocentricArt* (b) is a polar coordinate system with the pole in the center of the robot and the polar axis along *Wheel Axis 1* of the robot. *CNAllocentric* and *CNEgocentric* correspond to the actual coordinate systems used in the Carpe Noctem RoboCup team. *CNAllocentricArt* and *CNEgocentricArt* are introduced just for the purpose of this case study. However, the use of a polar coordinate system as *CNEgocentricArt* is plausible

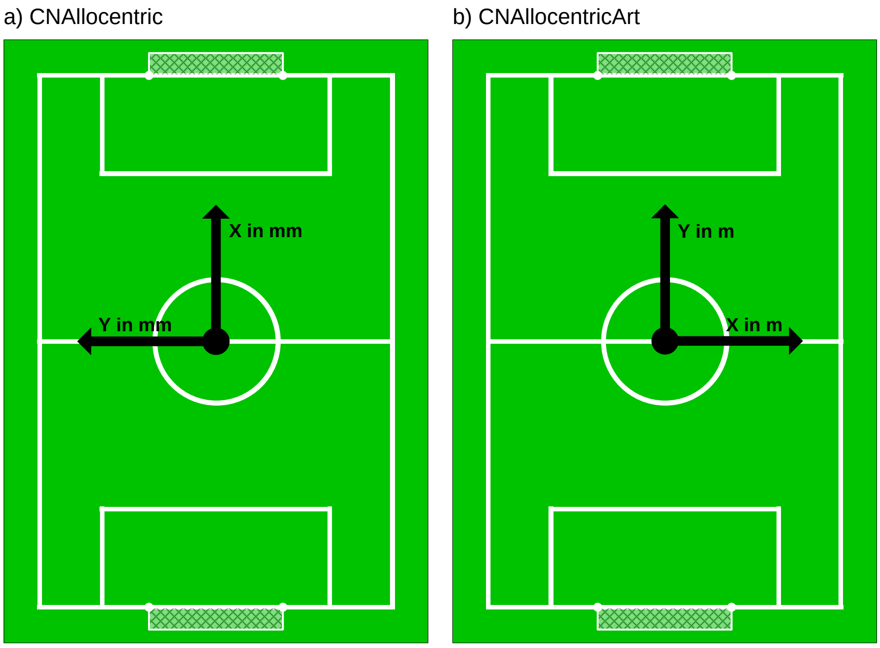

Figure 12.2: Two Possible Global Coordinate Systems in RoboCup

since for an omni-directional vision system adopted by most RoboCup Middle-Size teams polar coordinates are the most natural representation for detected objects.

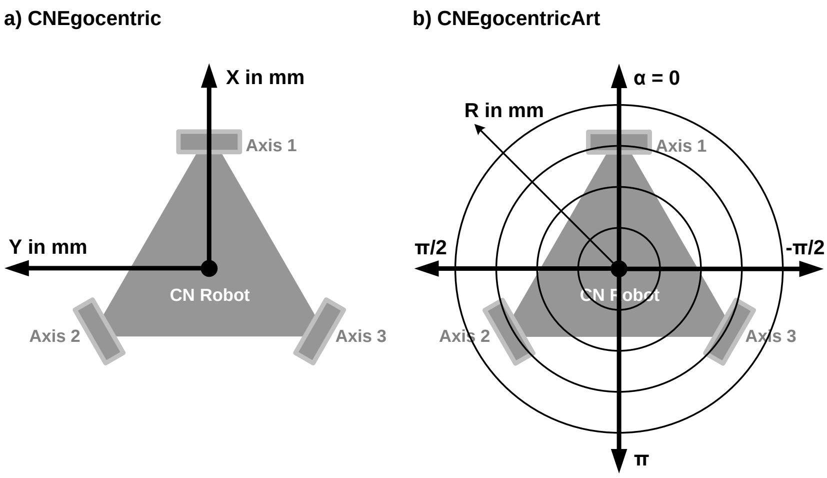

Figure 12.3: Two Possible Relative Coordinate Systems in RoboCup

In order to convert between relative and global coordinates, the estimated position of the robot on the field is also required and has to be communicated to the team members. This gives rise for another heterogeneity issue we introduce in our case study: One team may decide to communicate ball positions and robot positions separately, whereas other teams prefer to communicate combined messages that include both, the ball position and the robot position. Table 12.1 presents the three types of robots with heterogeneous message organizations and representations as used in this case study.

	Ball Position	Robot Position	Message Organization
Type A	CNEgocentric	CNAllocentric	combined
Type B	CNEgocentricArt	CNAllocentric	combined
Type C	CNEgocentricArt	CNAllocentricArt	separate

Table 12.1: Example of Heterogeneity Issues in Robot Communication

Although the communicated data are represented differently among the robots, only quite homogeneous information sources have been incorporated so far. All the communicated data originate from a vision system and a corresponding object detection and self-localization approach. In order to further emphasize heterogeneity in our case study, we also incorporate the *Referee Box Client* of the team as additional sensor for the ball position. During a RoboCup game, the *Referee Box Client* is responsible for receiving referee signals from the *Referee Box* and to propagate them to the connected clients of the teams. The *Referee Box* is a graphical tool that allows a referee assistant to enter referee signals, like e.g. *Kickoff*, *Throw In*, *Free Kick*, *Corner Kick*, etc. As in *Kickoff*, *Corner Kick* and *Throw In* situations the area for possible ball positions is quite restricted, we use the signals not only for preparing the team for the corresponding situation but also as additional sensor for the ball position. For this purpose, the *Referee Box Client* provides information about the ball position in the representation *CNAllocentric* as depicted in Figure 12.4.

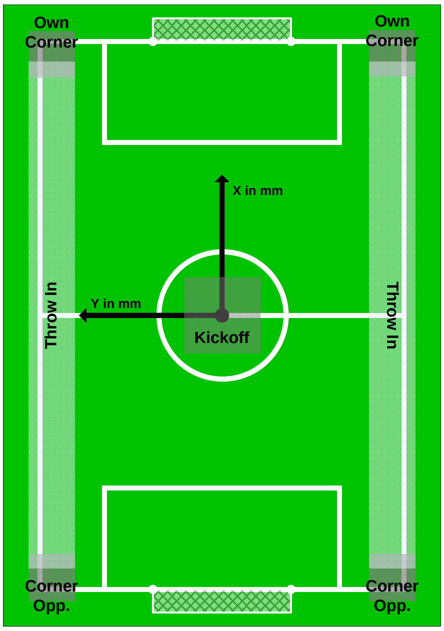

Figure 12.4: Referee Box Client as Sensor for Ball Position

12.2 Purpose of the Case Study

The general purpose of this case study is to demonstrate the applicability and viability of the proposed solution comprising the four main building blocks, *Information Model*, *Inter-Representation Operations*, *DST-based Information Fusion* and *Reasoning*, in a real-world scenario. The robots involved in this case study have to dynamically discover the information offers of other robots in the team and have to exchange and fuse heterogeneously represented information. Thus, in this case study we make heavy use of the *Information Model*, of *Inter-Representation Operations* and *DST-based Information Fusion* as illustrated in Figure 12.5. How to further exploit the exchanged and fused information for the coordination of the team is not in the focus of this case study and will be discussed only very briefly. A detailed investigation of this question, however, can be found in the Master's thesis of Triller [139], which was co-supervised as part of this work.

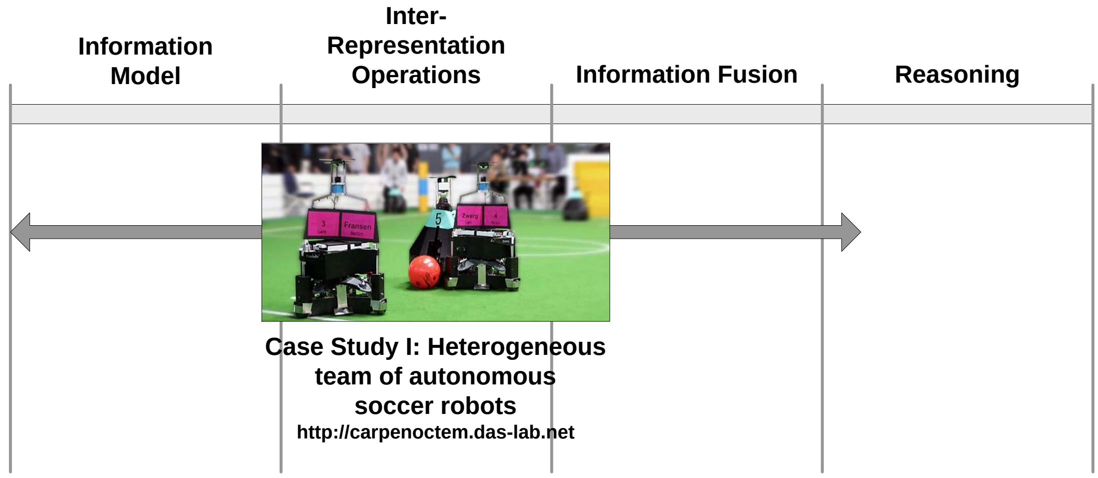

Figure 12.5: Purpose of Case Study I

The robots are required to make decisions and coordinate in the team within very short time intervals. Thus, the **performance of the proposed solution** with regard to CPU time required to realize the information exchange and fusion is a major issue. Here, it is also important to analyze how the performance is affected by the number of robots participating in the team and the number of hypotheses a robot contributes to the estimation of the ball position, i.e. the complexity of the involved belief functions. This leads us to the question of the **scalability of the proposed methods**.

In our review of related works, we have also seen that state-of-the-art approaches [125, 124] for fusing the observations of different robots to get an estimate of the ball position on the field are based on traditional probability theory and apply Bayesian filtering methods. Here, the question arises **how our proposed solution performs in comparison with the state-of-the-art approaches** and whether there are **benefits from the application of the Dempster-Shafer theory** instead of traditional probability theory.

In summary, this case study is intended to provide answers to the following questions:

1. Can the methods for information exchange and fusion proposed in this dissertation be successfully applied in a real-world scenario?
2. What is the performance of the proposed solution in a real-world application?
3. How do the proposed methods scale with respect to the number of participating robots and the complexity of the belief functions?
4. How do the methods perform in comparison to state-of-the-art approaches for cooperative object localization in the RoboCup community?

5. Are there benefits from the application of the Dempster-Shafer Theory or does it just increase complexity?

12.3 Implementation

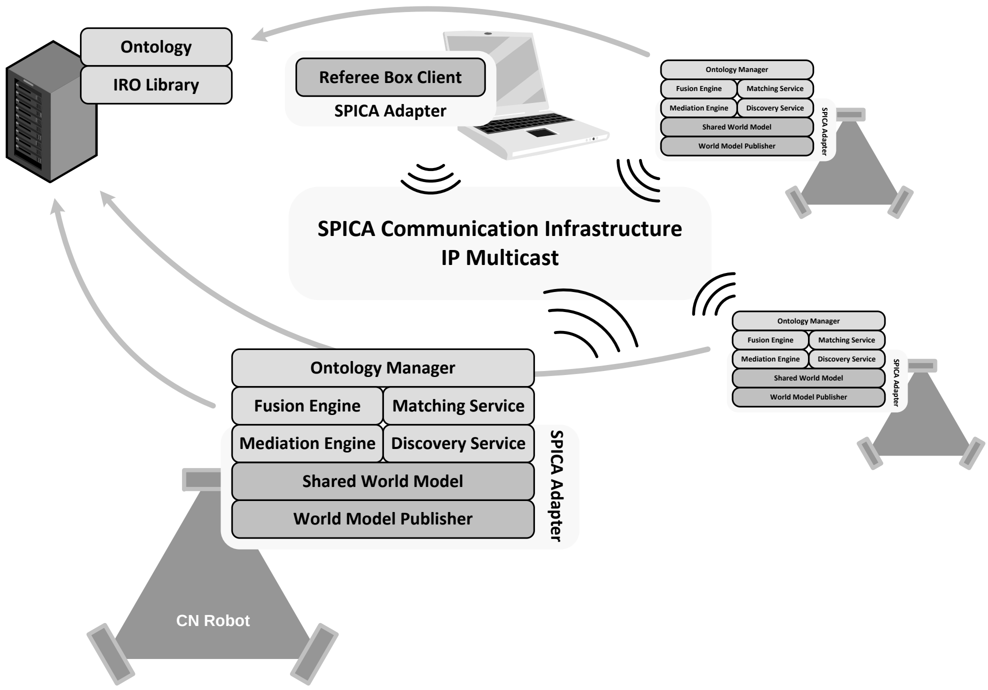

Figure 12.6: Architecture Realization for Autonomous Soccer Robots

For the realization of the case study, we created an instance of the abstract architecture presented in Chapter 11 and implemented almost all of the introduced architectural elements. There are only three minor simplifications but these should not be decisive for the overall system behavior: 1) The different robots participating in the team communicate the estimates of their localized position and the ball position on the filed, receive the corresponding information from their teammates and fuse them to a consistent ball estimate of the team. This means that we have information providers and consumers but no real information reasoner which registers information offers and requests with the discovery service at the same time. 2) In this scenario, there is no need for a general purpose ontology reasoner and thus it was not considered in the implementation. 3) As the robots communicate their estimated own position and the estimated ball position in very short time intervals of 33ms[1], we have abandoned the implementation of an information repository. Instead, the robots always process the most recently received data but consider the corresponding timestamps in the fusion process (see also Section 12.4).

As mentioned in Chapter 11, the abstract architecture identifies the main architectural elements and their relationships but leaves enough room for tailoring the architecture to a

[1]For this case study, the communication interval has been reduced from 100ms to 33ms in comparison to the actual Carpe Noctem control software. This allows to prepare for future challenges arising from faster robots currently developed in the RoboCup Middle Size League.

specific application domain. We have the option to decide for a centralized or decentralized realization of functionalities and may adjust the distribution of the functionalities over the different nodes of the computing environment as desired. An overview of the realization of the architecture for this case study is provided in Figure 12.6.

In general, the different nodes of the distributed computing environment, i.e. the autonomous mobile robots, the *Referee Box Client* and an external server, communicate via WLAN. In order to minimize communication overhead, each robot is equipped with its own *Ontology Manager*, *Discovery Service*, *Matching Service*, *Mediation Engine* and *Fusion Engine*. Interaction among them is realized through local method invocations. At runtime, communication is only required for registration of information offers and needs with the Discovery Services and for the actual information exchange with regard to the estimated robot positions and ball positions. The *Referee Box Client* and the *World Model Publishers* of the different robots act as information providers; the *Shared World Model* components of the robots are information consumers. Communication among the information providers, consumers and discovery modules is realized with the help of a SPICA generated communication infrastructure in form of *SPICA Adapters* and is based on IP multicast. The group communication scheme IP multicast has been selected instead of establishing peer-to-peer connections as information coming from the robots and the *Referee Box Client* has to be communicated to all robots in the team, which is naturally supported by IP multicast. This also means that the *Mediation Engine* is not responsible to establish communication channels between information consumers and providers in this scenario. At start-up, the *Ontology Managers* of the robots connect to the external server in order to fetch the *Ontology* using the standard HTTP protocol on top of TCP/IP. The external server also hosts the *IRO Library*, which is fetched by the *Matching Service* when reasoning about the required IROs is performed and their availability has to be checked.

All architectural elements were implemented using C# on top of Mono for Linux [85]. It was a kind of natural choice as most parts of the robot control software are written in this language. However, an issue occurred with the implementation of the *Ontology Manager* as there was no suitable API for OWL available for C#/Mono. Therefore, we used the Java library OWLApi2 [100] and converted it with the help of IKVM.NET [63] into a Mono-accessible assembly. It also turned out that the matrix operations required for the *Unscented Transformation*, for determining the *Mahalanobis* and *Bhattacharyya* distances, and for fusing two Gaussian distributions require too much CPU time if realized in C#. Thus, we utilized the Eigen C++ template library for linear algebra [34] and implemented appropriate wrappers to make it accessible from C# code.

Integration of the architecture into the other parts of the robot control software was straightforward. Instead of directly communicating its information to the other robots, the *World Model* component of the robot control software forwards the localized position and the estimates for the ball position to the *World Model Publisher*, which is responsible to perform the transformations to achieve heterogeneous representations and to send the information to the other robots. The *World Model Publisher*, as information provider, is also in charge of registering the information offers with the *Discovery Services* available in the environment. The *Shared World Model* component, which was already part of the robot control software, was extended to register its information needs with the *Discovery Services* and to receive the fused ball position from the *Mediation Engine*. Finally, the *Referee Box Client*, which is also part of the robot software framework, was prepared to act as information provider for the ball position and enhanced with support for registering the corresponding information offer.

The required IROs are provided as part of a C#/Mono IRO Library made available by the external server. They implement a simple standard interface providing just a method *transform* and are accessed by the *Mediation Engine* via *Reflection*. In order to avoid unnecessary access of the external server, the Matching Service and the Mediation Engine first check whether the corresponding library has already been downloaded and is available in a predefined local directory.

Uncertainty and impreciseness of the information to be exchanged is represented using Dempster-Shafer belief functions as described in Chapter 7, Chapter 8 and Chapter 9, which have to be maintained across the non-linear transformations required to adjust the different coordinate systems. Here, the belief functions for the ball and own position of the robots utilize the primitive *mean/covariance matrix* for the basic hypotheses. The belief function for the ball position from the *Referee Box Client* utilizes the primitive *value range*.

12.4 Application of the Dempster-Shafer Theory of Evidence

In our review of related work (see Chapter 6), it was described that state-of-the-art solutions in RoboCup to estimate the ball position from observations made by different robots of the team utilize Bayesian approaches [125, 124]. Thus, in this section we will show how our Dempster-Shafer-based method relates to Bayesian inference and that it has advantages over probability-based approaches in the scenario of our case study.

As already mentioned in Section 3.4, Bayesian inference updates an a priori estimate of the ball position x on the field with the likelihood of x with respect to an observation o of a robot:

$$p(x|o) = \alpha \cdot p(x) \cdot p(o|x) \tag{12.1}$$

In fact, this is a pointwise multiplication of the prior of x with the likelihood $p(o|x)$ and a subsequent normalization. The likelihood $p(o|x)$ results from the corresponding sensor model, which in our case is the sensor model of the camera system including the applied image processing steps used to detect the ball. Here, the sensor model often results in a multivariate Gaussian density function with mean x and covariance matrix Σ_x.

$$p(o|x) = \frac{1}{\sqrt{(2\pi)^n \cdot det(\Sigma_x)}} e^{-\frac{1}{2}(o-x)^T \Sigma_x^{-1}(o-x)} \tag{12.2}$$

The sensor model described in Equation 12.2 is a function of o for a given x, but the required likelihood is a function of x with respect to o. Assuming, however, that Σ_x is nearly constant in a certain neighbourhood of x, the likelihood with respect to o approximately represents a multivariate Gaussian again, with mean o and covariance matrix Σ_x. Furthermore, if we assume that the camera system provides a number of hypotheses $o_1, \ldots, o_m$ with corresponding confidences expressed as probabilities $P(o_i)$ with $\sum_i P(o_i) = 1$, the likelihood of x with respect to the observation constitutes a *Gaussian Mixture Model* (GMM) with weighting factors $P(o_i)$. These considerations are completely in line with the approach of Santos and Lima [124], where belief with regard to the ball position exchanged between the robots is also represented as GMM.

Updating a uniform prior of x with a GMM likelihood according to Equation 12.1 yields the GMM again. Now we can use this posterior estimate as prior for the next update step and combine it with the GMM received from another robot. The combination of the two GMMs is achieved by performing what Santos and Lima refer to as *Covariance Intersection* extended to GMMs [124].

In our approach, observations of different robots are fused by combining Dempster-Shafer belief functions of the form

$$m(x) = \sum_{H \in \mathcal{H}} m(x|H) \cdot m(H) \tag{12.3}$$

with Dempster's Rule of Combination. If there are no complex hypotheses but only basic hypotheses with the primitive mean/covariance matrix and $m(\Omega^X) = 0$, the mass assignment of Equation 12.3 happens to be a probability function representing a GMM. According to our fusion approach, combination of two GMMs has to be performed with

$$m_{1,2}(x) = \frac{1}{1-K} \sum_{H_1 \in \mathcal{H}_1} \sum_{H_2 \in \mathcal{H}_2} m_1(H_1) \cdot m_2(H_2) \cdot (m_1(x; \overline{x_1}, \Sigma_{H_1}) \cdot m_2(x; \overline{x_2}, \Sigma_{H_2})) \tag{12.4}$$

which actually corresponds to the Covariance Intersection extended to GMMs as described by Santos and Lima [124]. This is because Dempster's Rule of Combination reduces to a pointwise multiplication if the mass assignments are probability functions. Compared to Santos and Lima, however, we do not use an additional parameter γ to adjust the determinant of the result and apply a different method for calculating the weights of the components of the new GMM:

$$m_{1,2}(H_{ij}) = \alpha \cdot A^*_{H_i \oplus H_j} \cdot m_1(H_i) \cdot m_2(H_j) \tag{12.5}$$

where α is a normalization factor to ensure that the $m_{1,2}(H_{ij})$ sum up to 1. $A^*_{H_i \oplus H_j}$ denotes the agreement of the hypotheses H_1 and H_2 (see also Chapter 9). Santos and Lima calculate the new weight as $\frac{1}{N}(m_1(H_1) + m_2(H_2))$ (with N being the number of components of the GMMs) [124], which has to be regarded as an ad-hoc approach and is questionable to have a direct justification in probability theory.

In [124], the authors argue that it is very important to consider the imprecision of the self-localization approach of the robot when an estimate of the ball position in egocentric view has to be transformed to the world view, which is neglected in most related works. In our approach, this is naturally supported by IROs that have a dependency to an estimate of the robot's localized position. Furthermore, we argue that also network latency has to be taken into account when fusing the observations for the ball of the different robots. The ball moves with up to 5m/s if passed or dribbled by a robot, and a network latency of 100ms can often be observed during RoboCup tournaments. This causes an estimation error of about 50cm even if the robots make perfect observations and process images which are captured at exactly the same time.

Actually, Santos and Lima propose a Bayesian filtering approach that applies an object motion model to predict the probability distribution into the future and then uses this prediction as prior for the update with the corresponding observations of the robots. Thus, in principle the Bayesian filtering proposed by Santos and Lima naturally supports consideration of network latencies. However, this is not discussed at all in [124]. Our method is not tailored to application in RoboCup but constitutes a general method for information exchange and fusion in heterogeneous distributed environments. Consequently, we do not have a ball motion model or, more generally speaking, no object state transition model is available. Still, our approach is able to consider network latency as illustrated in the following paragraphs.

In Section 9.2.1 and Section 9.2.2 it is described how *discounting* can be used to adjust Dempster-Shafer belief functions in order to take into account sensor reliability and the up-to-dateness of the information to be fused. With discounting, a portion of the mass assignments, determined by the discounting factor α, is shifted to the mass of the whole frame of discernment. In order to take into account network latency, we discount the belief functions with regard to its timestamps according to the following equation:

$$\alpha = 1 - e^{-\frac{1}{50ms} \cdot (t_{mr} - t_o)} \tag{12.6}$$

where t_{mr} denotes the timestamp of the most recent observation and t_o the timestamp of the observation, whose belief function is adjusted by discounting. The factor $\frac{1}{50ms}$ results from the consideration that with an assumed speed for the ball and the robots of up to 5m/s (goal kicks excluded), the ball position changes up to 25cm every 50ms, which has to be regarded as significant change. Meaningful timestamps are an essential prerequisite of this approach, and thus the local system time of the robots and the Referee Box Client has to be synchronized. In our case study, this has been achieved with the help of NTP Version 4 [95].

For easier illustration of the effect of our discounting approach, we will restrict our description to two robots observing the ball. Thus, we assume the following situation, which is illustrated in Figure 12.7. The first robot observes the ball at *CNAllocentric* coordinates $(-318.71, -76.31)$ with belief mass 0.89 (magenta observation). The second robot observes the ball at *CNAllocentric* coordinates $(241.77, -79.88)$ with belief mass 0.91 (blue observation). The covariance matrices are

$\begin{pmatrix} 4.2861 \cdot 10^4 & 3.4745 \cdot 10^4 \\ 3.4745 \cdot 10^4 & 8.2981 \cdot 10^4 \end{pmatrix}$ for the first robot (magenta observation) and,

$\begin{pmatrix} 6.9004 \cdot 10^4 & -5.1615 \cdot 10^4 \\ -5.1615 \cdot 10^4 & 1.2860 \cdot 10^4 \end{pmatrix}$ for the second robot (blue observation)

and are depicted by the magenta and blue ellipses corresponding to the 2σ-contours.

In Figure 12.7, the black ellipses describe the hypotheses resulting from combining the magenta and blue observations through Covariance Intersection. The red ellipses illustrate the hypotheses that abstract the GMMs resulting from combination of the observations from the two robots through a single Gaussian. Each sub-figure shows the results for a different time difference between the magenta observation (most recent) and the blue observation. It can be observed that by our discounting approach the blue observation has less effect on the result (red ellipse) the bigger the time difference gets between the observations. Thus, the robots consider more recent information as more reliable. Too old information (time difference $>$ 100 ms) has almost no influence at all.

The approach presented above heavily relies on assigning a belief mass to the whole frame of discernment, which expresses that the ball position is unknown to a certain degree/belief.

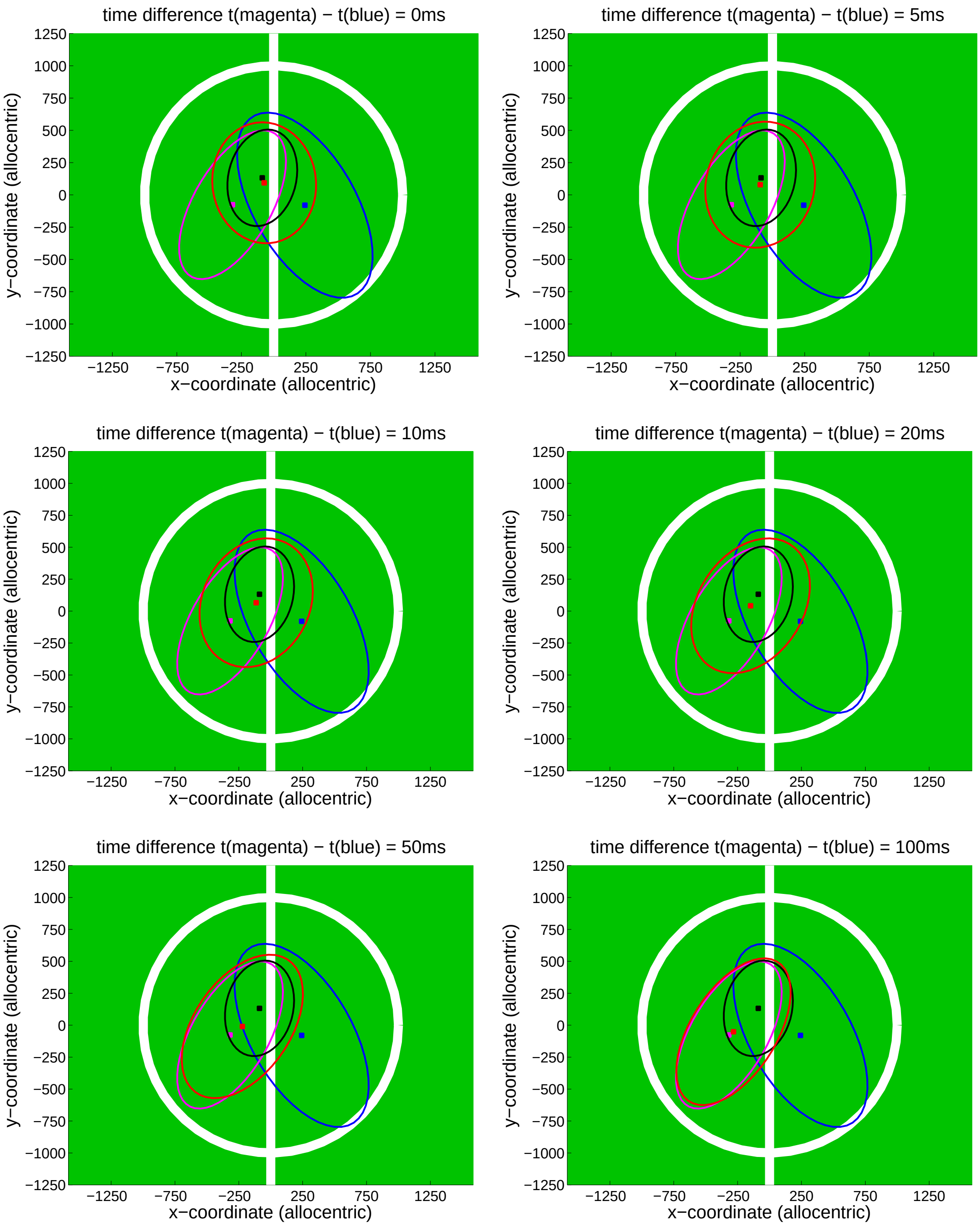

Figure 12.7: Effect of Discounting with regard to the Time Difference of Observations

Modelling such a kind of ignorance, however, is of big importance if sensor reliability and confidences in certain observations should be considered. For example, if a robot detects the ball at position x with confidence 0.6, we cannot assume that with confidence 0.4 the ball is not at position x. Instead, with confidence 0.4 we have no idea of the ball position. In a Bayesian framework, ignorance is commonly expressed by a uniform distribution, which in many cases yields the desired result. So for example, total ignorance could be expressed in the GMMs of Santos and Lima by introducing a Gaussian mixture component with mean $(0.0, 0.0)$ and a covariance matrix that has very large eigenvalues, for example $25 \cdot$ *(field length in mm)*^{2}. With this method, the technique of discounting could also be applied within their approach. However, a problem arises with the IROs required to adjust the representations (coordinate systems) in a heterogeneous mixed team as envisaged in this case study. If modelling ignorance through the uniform distribution, pure Bayesian probability theory does not provide means to distinguish between ignorance of the ball position and the knowledge that the ball position is uniformly distributed across the field.

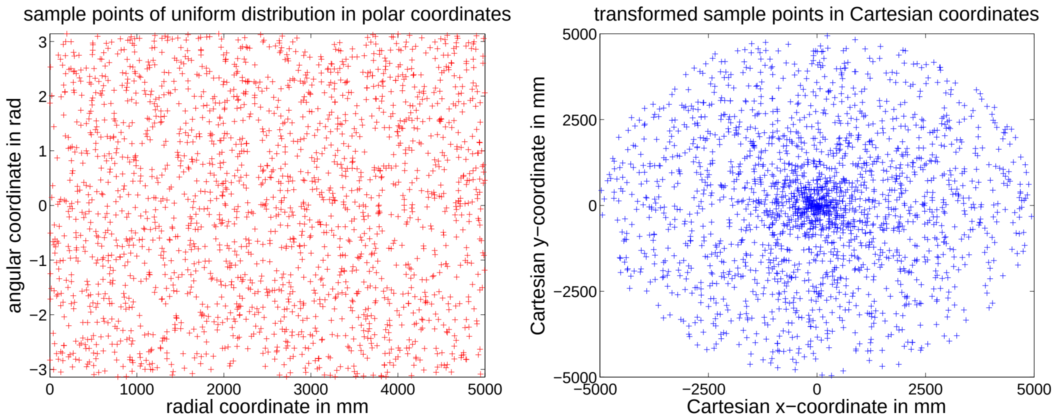

Figure 12.8: Non-linear Transformation of a Uniform Distribution

Assume, for example, an IRO that converts from polar coordinates to Cartesian coordinates as required in this case study. A uniform distribution in polar coordinates used to express ignorance would not yield a uniform distribution in Cartesian coordinates as illustrated in Figure 12.8. Thus, the application of IROs as envisaged in this thesis in a pure Bayesian probability framework cannot guarantee to maintain the modelled ignorance. Instead, using Dempster-Shafer belief functions, ignorance can be modelled by assigning masses to the whole frame of discernment, which are naturally maintained by IROs. With Dempster-Shafer Theory it is even possible to distinguish between complete and partial ignorance by utilizing mass assignments for complex hypotheses instead of assigning a mass to the whole frame of discernment. With the help of the example illustrated in Figure 12.9, we investigate the effect of assigning masses to complex hypotheses instead of distributing the masses uniformly among the involved basic hypotheses as it would be done in a Bayesian framework. Furthermore, we show how domain knowledge can easily be incorporated in our fusion method.

Figure 12.9 depicts a typical situation during a RoboCup game. A Carpe Noctem robot has succeeded to dribble the ball into the opponent half of the field and kicked it towards the goal. However, the ball hit one of the goal posts and was moved by an opponent defender out of the field. This is why the Carpe Noctem team gets a corner kick and the ball is manually positioned by the referee to the corresponding corner of the field. Only three of the four Carpe Noctem field players (CN 2, CN 3, CN 4) are correctly self-localized (depicted by confidences greater than 0.97) and only two of them (CN 1 and CN 2) detect a ball. Unfortunately, robot CN 1 is de-localized, detects a ball in the audience (false positive) with a high confidence of 0.87 and maps the ball using a wrong estimate for its own position to the field. Although CN 2 knows that his estimate of the position is not very reliable (confidence 0.60), without incorporation of additional knowledge the robots would agree on the wrong ball position (belief mass $0.60 \cdot 0.87 = 0.522$ for the wrong ball position, belief mass $0.98 \cdot 0.32 = 0.3136$ for the correct ball position) and adjust the team play accordingly.

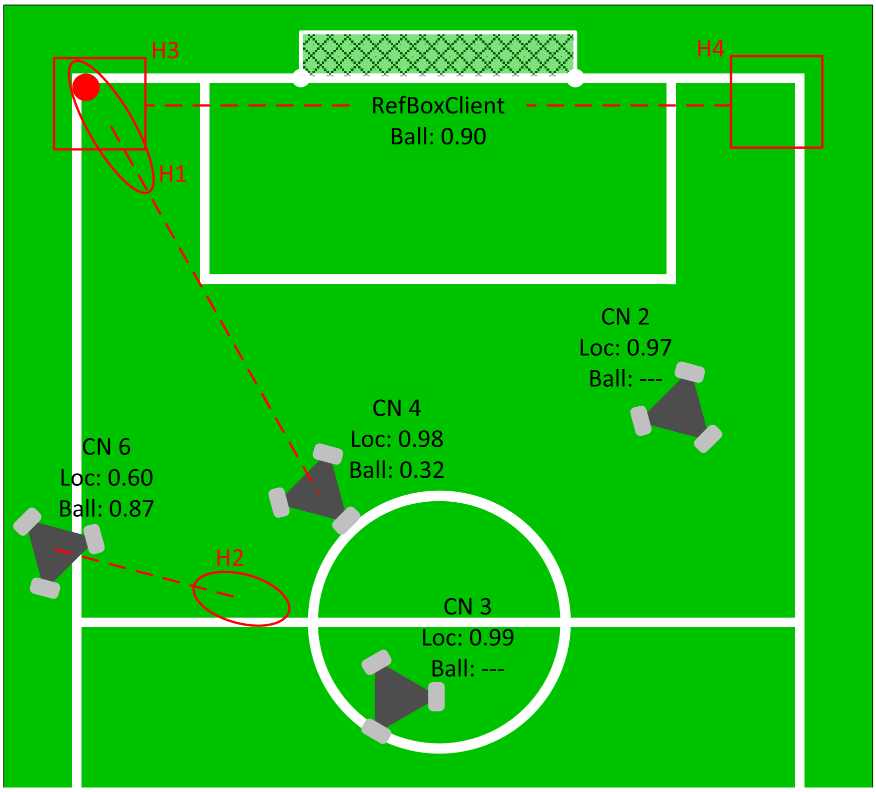

Figure 12.9: Referee Box Client used to Stabilize the Ball Position Estimation

In such a situation, the Referee Box command can be exploited as additional sensor for the ball position. During a corner kick, as shown in the example of Figure 12.9, there are only two possible ball positions on the field, namely the two corners of the opponent half. However, there is no possibility to distinguish between the two corners only from the Referee Box command. In our example, this is modelled by assigning a mass of 0.9 to a complex hypothesis which is the union of two basic hypotheses each covering a corner with a corresponding *value range* (red rectangles). In a Bayesian framework, the mass of 0.9 would be distributed equally among the involved basic hypotheses, i.e. the two basic hypotheses would be assigned 0.45 each.

Table 12.2 shows the results of the fusion process. The row for m_C contains the result if the mass of 0.9 is assigned to the complex hypothesis and the row for m_B shows the result for equally distributed masses. In the first case, the complex hypothesis $H3 \cup H4$ has the highest mass assignment whereas in the second case the two basic hypotheses $H3$ and $H4$ have the highest mass values. For decision making, Cobb and Shenoy [24] have proposed the plausibility transformation introduced in Section 3.5. Basically, the plausibility transformation calculates the plausibility for each element of the frame of discernment and normalizes the values to sum up to 1. The elements of the frame of discernment are atomic in the sense that no masses can be assigned to a part of the element. In our case, however, we calculate mass assignments also for $H1 \cap H3$, $H1 \cap \overline{H3}$ and $\overline{H1} \cap H3$, which are part of $H1$ and $H3$ respectively. Thus, we have to consider $H2$, $H4$, $H1 \cap H3$, $H1 \cap \overline{H3}$ and $\overline{H1} \cap H3$ as the actual elements of the frame of discernment. Applying the plausibility transformation in this way to the mass assignments m_C and m_B yields the rows Pl_P_C and Pl_P_B. In both cases, we would decide for the complex hypothesis $H1 \cap H3$, which corresponds to the correct ball position. The resulting confidence values are 0.38 and 0.35 respectively. At the first glance, these confidence values are surprising as there is almost no conflict between the hypothesis $H1$ and $H3 \cup H4$ if assigning the whole mass to $H3 \cup H4$, whereas there is only an agreement of $H1$ and $H3$ if the masses are equally distributed among the two involved basic hypotheses, and total conflict for $H1$ and $H4$. The reason for the almost identical confidence values can be found in the normalization factors used in Dempster's Rule of Combination and in the plausibility transformation. Consequently, in our example situation there is no big improvement of expressing partial ignorance through assigning the mass to the complex hypothesis as allowed in Dempster-Shafer theory instead of equally distributing the masses among the involved basic hypotheses as in a Bayesian framework.

	H1	**H2**	**H3**	**H4**	**H3 ∪ H4**	**H1 ∩ H3**	**H1 ∩ $\overline{\textbf{H3}}$**	**$\overline{\textbf{H1}}$ ∩ H3**	Ω^H
m_C	0.03	0.07	–	–	0.58	0.26	–	–	0.06
m_B	0.03	0.08	0.33	0.33	–	0.15	–	–	0.08
Pl_P_C	–	0.06	–	0.26	–	**0.38**	0.04	0.26	–
Pl_P_B	–	0.10	–	0.24	–	**0.35**	0.07	0.24	–
Pl_C^{CN}	**0.87**	0.07	0.87	0.58	–	–	–	–	–
Pl_B^{CN}	**0.51**	0.08	0.51	0.33	–	–	–	–	–

Table 12.2: Results of the Fusion for the Example Situation

Applying the plausibility transformation as described above yields low confidence values although a hypothesis with a confidence value of 0.31 is confirmed with a hypothesis of 0.9. It also requires the determination of the refined frame of discernment, which can be quite complex if the number of involved hypotheses gets high. For these reasons, we use another approach for decision making in this case study. We just calculate the plausibility of the basic hypotheses $H1$, $H2$, $H3$ and $H4$ without considering the mass assignment of Ω^H, omit the normalization and decide for the basic hypothesis with the highest plausibility value. If two basic hypotheses have the same value, the hypothesis originating from an observation of a robot is favored. The mass assignment to Ω is not considered as it expresses total ignorance and thus should not contribute to the confidence value. Furthermore, consideration of the mass assignment to Ω^H would always yield a confidence value of 1.0 if only one basic hypothesis is involved. Our new approach for decision making results in the rows Pl_C^{CN}

and Pl_B^{CN} of Table 12.2. A high confidence value of 0.87 for $H1$ is obtained if assigning the masses to the combined hypothesis and a medium confidence value of 0.51 if the masses are equally distributed among the involved basic hypotheses. This new approach for decision making provides reasonable results in this case study, but it still has to be investigated if this approach can also be used and behaves properly in other application domains.

12.5 Performance and Scalability

Whereas the previous section was concerned with evaluating our approach applied to the case study from a qualitative point of view, this section assesses the performance and scalability of our prototype implementation described in Section 12.3. Good performance is very important for the application in the domain of cooperative autonomous robots as they have to make decisions and to adapt their behavior in very short time intervals.

	CN Ontology	**MUSIC Ontology**
DL Expressivity	ALCROIN(D)	ALCROIN(D)
classes	89	228
object properties	46	102
data properties	12	13
sub-class axioms	123	337
sub-properties	22	65
individuals	112	110
class assertion axioms	124	110
object property assertion axioms	150	151
data property assertion axioms	86	75

Table 12.3: Metrics of the Used Ontologies

Generally speaking, parsing ontologies and reasoning with the represented knowledge is quite resource consuming. Thus, in our prototype implementation we preprocess the concepts defined in the ontology once at start-up. The contained information is stored in data structures, which are tailored to facilitate fast matching of information offers and requests and reasoning about the required mediation tasks with minimal overhead. We measured the time required for ontology loading (fetching and parsing with OWLApi2 [100]) and ontology preprocessing on three different computer systems. We used two different ontologies, whose metrics are presented in Table 12.3. The *Carpe Noctem Ontology (CN Ontology)* is the ontology actually used in this case study and the *MUSIC Ontology* forms the baseline for the MUSIC Context Model. Both ontologies are defined according to the *Information Model* presented in Chapter 7, but differ in the number of the contained concepts and their relationships. For example, the MUSIC Ontology defines 228 classes and 102 object properties, whereas the CN Ontology only defines 89 classes and 46 object properties. The results of our measurements on a Core 2 Duo 2.53GHz with 4GB RAM, a Pentium IV 3.0 GHz with 512MB RAM and a Core Duo 1.66GHz with 2GB RAM are presented in Figure 12.10, which depicts the mean values of 20 trials.

It can be observed that ontology loading requires up to a couple of seconds, whereas ontology preprocessing only requires from 29ms (CNOntology on the Core 2 Duo) up

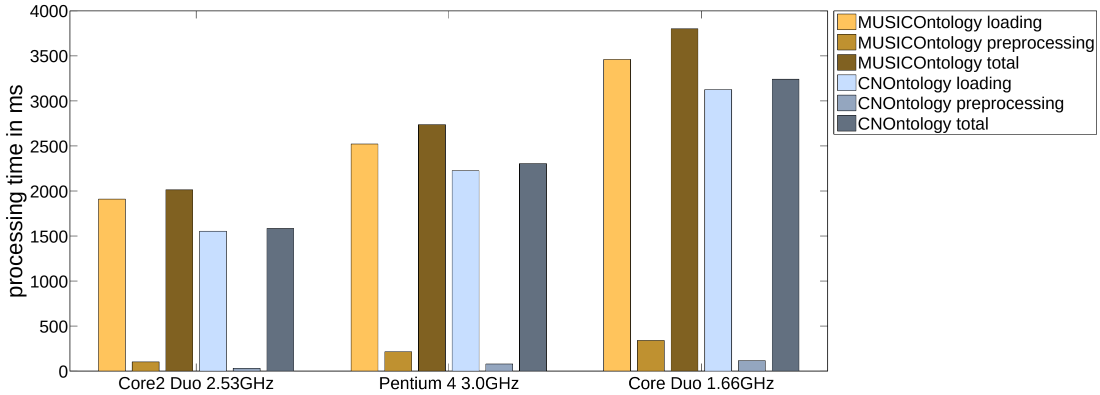

Figure 12.10: Processing Time of the Ontology Manager for Remote Ontologies

to 281ms (MUSICOntology on the Core Duo). However, the ontologies as well as their imported ontologies were fetched from the Web, and thus network latency and delays caused by the involved servers influenced the measurements. Therefore, we repeated the test with ontologies already available on the local file system. The corresponding results of the measurement are illustrated in Figure 12.11.

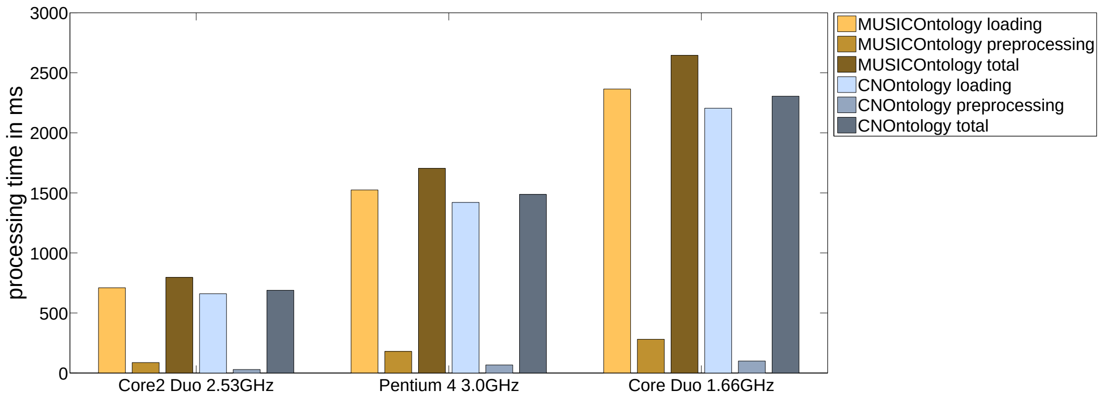

Figure 12.11: Processing Time of the Ontology Manager for Local Ontologies

With locally available ontologies the time for ontology loading decreased by about 1s, whereas the time required for ontology preprocessing, just as expected, remains the same. With the help of the data structures resulting from the preprocessing step, registration of information requests with the discovery service, their matching against information offers and reasoning about the required mediation tasks can be achieved in about 95μs on the Core Duo for this case study, where only a limited number of 8 IROs is available. An investigation how this time increases with the number of involved representations, IROs and their dependencies remains for future work.

More important for the performance of our prototype implementation than ontology loading and processing, which are only required once during start-up, are the tasks that have to be performed in each iteration of the robot control cycle, i.e. at a frequency of 30Hz:

serialization and deserialization of the messages, performing the IROs and fusing the estimates. We measured the time required for the different tasks using the following test setup: Five robots play soccer with the Carpe Noctem team strategy used during the RoboCup world championships 2009. They communicate the estimated ball position and their own position. The exchanged messages are organized and utilize the representations according to the three types of robots defined in Table 12.1. One robot is of Type A, two robots of Type B, and also two robots are of Type C. In addition, the Referee Box Client provides information about the ball position during a standard situation using a symbolic representation. Information with regard to the ball position is requested on all robots in the representation *CNAllocentric*. Thus, in one control cycle, our prototype implementation has to serialize one or two messages, to deserialize up to 9 messages, to perform up to 12 IROs and to fuse 6 estimates for the ball position. Figure 12.12 presents the results of our measurements (mean values for 200 control cycles) on three different computer systems. As the actual Carpe Noctem robots used in this scenario are equipped with a Core Duo 1.66GHz, we could measure the time under real conditions only on this system. For the other configurations we had to perform the measurements with the help of the Carpe Noctem robot simulator.

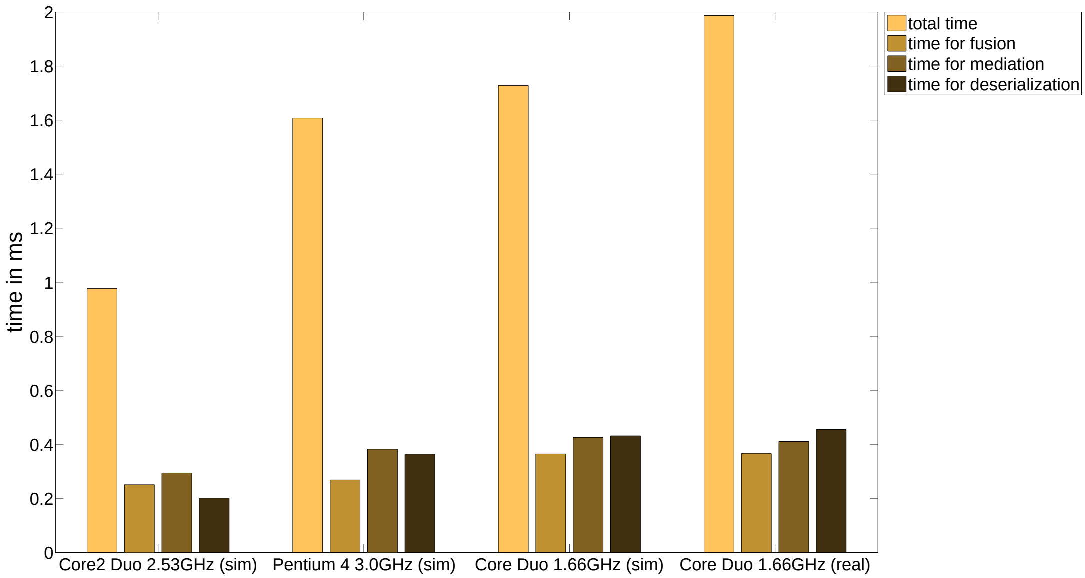

Figure 12.12: Performance of the Prototype Implementation

Figure 12.12 shows that under real game conditions our prototype implementation requires in total about 2ms in one control cycle. The average time for deserialization is 0.45ms, the average time for the mediation tasks 0.41ms and the average time for fusing the ball estimates 0.36ms. Under real game conditions about 80% of one core of the CPU is occupied by the image processing module, which is not running in our simulation environment as all information on the world model objects are provided by the simulator. It can be observed that the time for fusion, mediation and deserialization is almost constant, whereas the average total time decreases from 1.99ms to 1.73ms. This decrease is explained by the less frequent interruption of our routine by the scheduler, when the background workload is small. The measurements on the other computer systems reveal no surprising results and just reflect the better performance of the CPUs. So, for example, on a system with an up-to-date 2.53GHz Core 2 Duo 2.53GHz, the average total time required by our prototype implementation in one control cycle decreases to 0.97ms.

In the previous paragraphs, we have assessed the performance of our prototype implementation for a current RoboCup game situation with five robots and the Referee Box Client. The following paragraphs will use our prototype implementation to investigate the scalability of the concepts proposed in this dissertation. All measurements were performed on the 1.66GHz Core Duo with 2GB RAM in a simulated environment that allows us to tune the different parameters which are of interest for the scalability tests.

The time required for deserialization of messages is mainly determined by the complexity of the belief function to be represented, i.e. by the number of defined basic and complex hypotheses and the number of dimensions (number of contained scopes) in a basic hypothesis. Figure 12.13 presents the results of the corresponding scalability test. All values depict the average time for 20 trials.

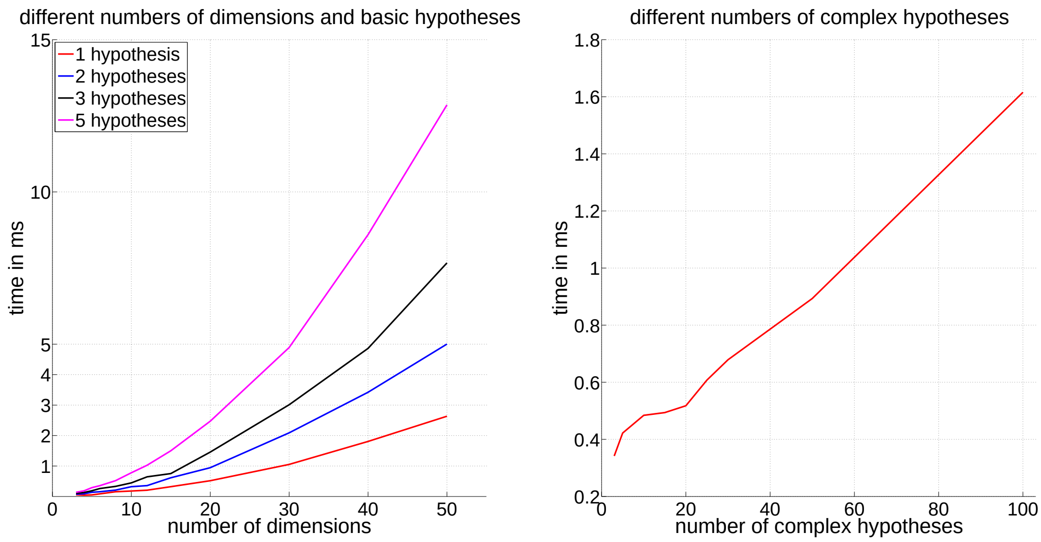

Figure 12.13: Results of the Scalability Test for the Deserialization Method

The left part of Figure 12.13 shows that the average time for serialization/deserialization is more or less proportional to the number of basic hypotheses (number of complex hypotheses was set to zero in this test) and grows quadratically with the number of dimensions. This is completely in line with the fact that also the message size is proportional to the number of basic hypotheses in this case, and quadratic to the number of dimensions. This is because the size of the covariance matrices used to express uncertainty also grows quadratically with the number of dimensions. The right part of Figure 12.13 illustrates the results of the scalability test with regard to the number of complex hypotheses. The number of dimensions is set to 5 and the number of basic hypotheses was fixed to 7. Obviously, the required time for deserialization grows linearly to the number of complex hypotheses if the representation of the complex hypotheses contributes most to the message size.

The time required to perform an IRO is mainly determined by the following four factors: 1) time for a single point-to-point conversion, 2) number of additional dependencies of the IRO, 3) number of dimensions of the involved basic hypotheses and 4) number of basic hypotheses. The results of the corresponding scalability test are presented in Figure 12.14. Here too, all values depict the average time for 20 trials. It is also noteworthy that the method which realizes the point-to-point conversion is fixed for all tests, and thus the required time for the method was nearly constant.

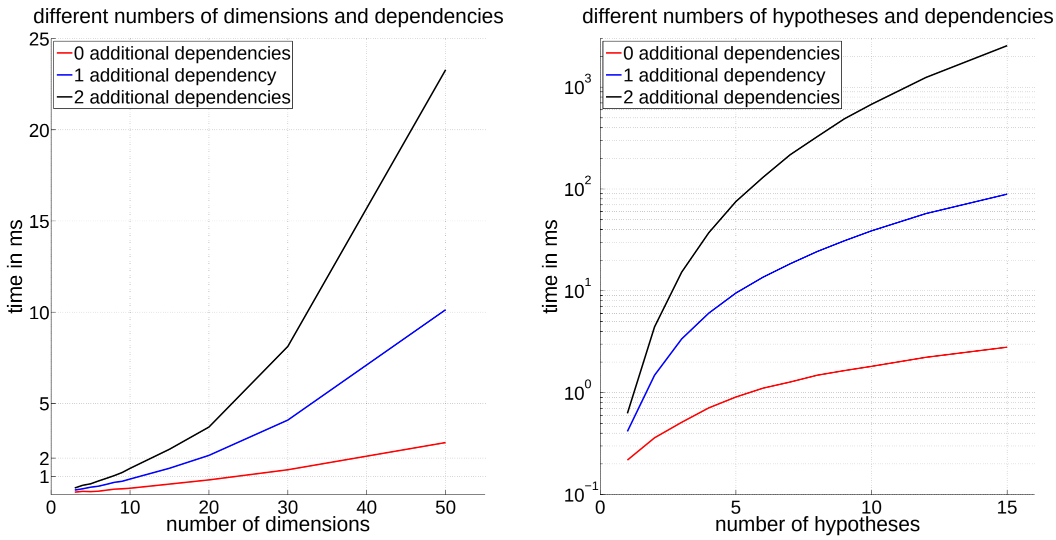

Figure 12.14: Results of the Scalability Test for the IROs

From the left part of Figure 12.14 we observe that the required time grows nearly proportionally in the range of 3 to 40 dimensions if the IRO has no additional dependencies and only one basic hypothesis has to be transformed. This is in line with the consideration that the *Unscented Transformation* used to propagate covariance matrices across non-linear functions requires the transformation of $2n + 1$ sample points, where n is the number of dimensions. For more than 40 dimensions, however, the effort for the Cholesky Decomposition, which has to be performed in order to determine the sample points, has more and more impact. It requires about $\frac{1}{6}n^3$ multiplications for a covariance matrix of size $n \times n$, which causes the time to grow cubically with the number of dimensions. If also additional dependencies are involved, the number of 'effective' dimensions is increased by the sum of the number of dimensions of the additional dependencies. In our test setup, we used the same number of dimensions for the additional dependencies as for the real basic hypothesis. Thus, the time required for 30 dimensions and no additional dependency corresponds to the time for 10 dimensions with 2 additional dependencies. The time for 40 dimensions and no additional dependency is nearly the same as for 20 dimensions and one additional dependency.

The right part of Figure 12.14 shows the influence of the number of basic hypotheses on the time required for performing the IRO (average of 20 trials). In our experiment, the number of dimensions was fixed to 5 and the number of basic hypotheses was constant for the actual information element to be transformed and for the additional dependencies. From the plots we observe that the required time grows linearly with the number of hypotheses if there is no additional dependency, quadratically for one additional dependency and cubically for two additional dependencies. This is due to the fact that each combination of the hypotheses has to be transformed. For 5 hypotheses this means 5 transformations if there are no dependencies, $5 \cdot 5 = 25$ transformations with one additional dependency, and $5 \cdot 5 \cdot 5 = 125$ transformations with two additional dependencies.

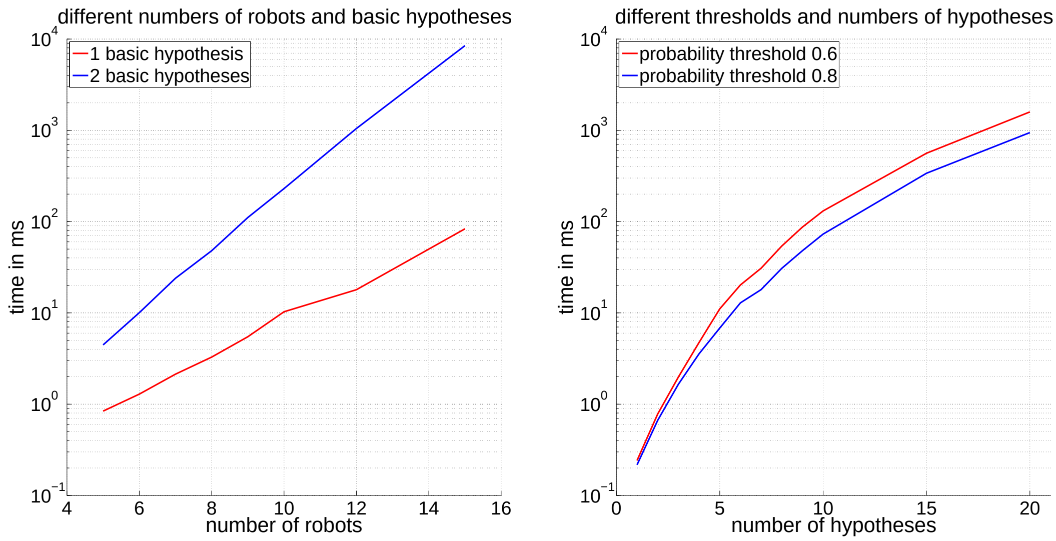

Figure 12.15: Results of the Scalability Test for the Fusion Approach

The time required for our Dempster-Shafer-based information fusion method is mainly determined by the number of participating robots, i.e. the number of information providers, and the number of basic hypotheses used to define the corresponding belief functions. In order to evaluate the scalability of the fusion approach, we generated belief functions with basic hypotheses based on the primitive mean/covariance matrix. The mean values are uniformly distributed in a rectangular area of $5000mm \times 5000mm$ on the field. The covariance matrix was fixed and has only elements unequal to 0 on its diagonal. All the variances are set to $300mm \times 300mm$. This guarantees a high number of intersections of the basic hypotheses, which results in new basic hypotheses for the next fusion step.

For random generation of unions of basic hypotheses, i.e. complex hypotheses, we applied the following approach: If a belief function with n hypotheses (basic or complex) is desired, we first generate n basic hypotheses as described above. Afterwards, we generate a vector of n random numbers $(x_1, \ldots, x_n)$ in the interval of $[0, 1]$. If x_i is greater than a certain threshold, the basic hypothesis i is part of the union. A union with only one basic hypothesis results in a belief assignment to the basic hypothesis itself. If a union of hypotheses is the same as a previously generated union, a new vector of random numbers is used. This way, we can adjust the number of belief assignments to basic and complex hypotheses by tuning the threshold. These steps are repeated until n belief assignments to basic or complex hypotheses have been realized.

The results of our scalability test are presented in Figure 12.15. The left part of the figure shows the time required for information fusion in our prototype implementation depending on the number of participating robots for one and two basic hypotheses (no complex hypothesis). It is obvious that the time grows exponentially with the number of robots, i.e. information providers. This is due to the fact that combining two belief functions with n basic hypotheses each, results in a belief function with at most $(n+1)^2 - 1$ basic hypotheses (each basic hypothesis of the first belief function has intersections with all basic hypotheses of the second belief function). If this belief function is combined with a further belief function with n basic hypotheses, we get a maximum number of $(n+1)^3 - 1$ basic hypotheses. Generally speaking, the combination approach results in a maximum number of $(n+1)^m - 1$ basic hypotheses for m robots.

The right part of Figure 12.15 illustrates the influence of the number of hypotheses (basic and complex) on the time required for the fusion. In this experiment, we fixed the number of information providers (robots) to three. It can be observed that the time grows roughly like n^3, which is in line with the considerations of the previous paragraph, and also that by trend less time is required the more complex hypotheses (unions of basic hypotheses) are used in the belief function. At first, this might be surprising as the belief functions define the same number of basic hypotheses that have to be checked for intersection. However, due to the fact that the distributive law does not hold for the general combination of two unions of basic hypotheses, we have complex hypotheses that cannot be simplified and thus do not result in new basic hypotheses by Covariance Intersection.

In summary, the number of participating robots and the number of basic hypotheses used in the belief functions are the most limiting factors of our solution approach consisting of serialization/deserialization of messages, performing the IROs and fusing the different belief functions. In particular, the scalability of our fusion approach with regard to the number of information providers has to be considered as problematic. We have to expect an exponential growth in the number of involved basic hypotheses. With regard to this case study, our analyses revealed that the approach is feasible for about 12 robots if only one basic hypothesis is used for the ball position and for about 6 robots if a belief function for a ball estimate defines two basic hypotheses.

12.6 Summary and Discussion

As part of this case study, we have shown how our Dempster-Shafer-based information fusion method relates to a current state-of-the-art approach in RoboCup [121] which apply Bayesian methods to fuse estimates of the ball position from different robots. Very similar to what was proposed by Santos and Lima, our approach results in a combination of Gaussian Mixture Models if the belief functions are specified with basic hypotheses based on the primitive mean/covariance matrix. Our concept of IROs naturally supports consideration of the precision of the self-localization of the robot when converting between coordinates relative to the robot and absolute world coordinates. This was claimed by Santos and Lima as one of the highlights of their work and they argue that this important issue has been neglected by most other approaches so far. Besides, we have shown how network latencies can be considered very easily in our information fusion method. This is not discussed in [124] at all even though their framework would allow for it. Here, the approach of Santos and Lima can be expected to achieve even more precise results than our method as they have available an object motion model. However, we have to keep in mind that the framework of Santos and Lima has been designed specifically for the RoboCup domain and the requirements of fusing ball position estimates. In contrast, our generic framework for information exchange and fusion is not tailored to a particular application domain. Nevertheless, it can compete with the state-of-the-art approaches in the domain of RoboCup.

The Dempster-Shafer Theory of Evidence is more expressive than pure Bayesian probability theory. It allows to explicitly model partial or complete ignorance. In probability theory, this can only implicitly be expressed by special distributions such as the uniform distribution. Thus, in principle our approach is also more expressive than the one proposed by Santos and Lima. Here, our analyses revealed that due to the normalization factors in Dempster's Rule of Combination and in the plausibility transformation used for decision making, the increased expressiveness only yields slightly better results, which would not justify the increased complexity. However, if non-linear IROs have to be performed to overcome mismatches in the requested and provided representations as envisaged in this scenario, implicit modelling of ignorance as done in probability theory is not appropriate anymore. This is because the modelled ignorance is not properly maintained across the IRO. Using our Dempster-Shafer-based method instead, this is naturally supported.

The performance evaluation of our prototype implementation has shown that the solution proposed in this dissertation is applicable for the RoboCup domain where decisions have to be made in very short time intervals of some milliseconds. However, our solution does not scale with the number of information providers: The time required for the fusion step grows exponentially with the number of information providers in the worst case. It is noteworthy here that the same phenomenon is observed in the combination of GMMs if all components of the fused GMM are maintained. With GMMs and probability theory a common approach is to reduce the number of components through approximation of the distribution by a smaller number of components. However, this may cause the result of the fusion to be dependent on the order of the fusion steps. For our Dempster-Shafer-based approach, approximation is even more difficult as we also allow specification of complex hypotheses that have to be maintained in the approximation. The search for appropriate methods that reduce the number of involved basic hypotheses remains open for future work.

It has also to be noted here that the Dempster-Shafer Theory of Evidence in general is characterized by scalability issues as it operates on the power set of a frame of discernment. In this case study, the frame of discernment for the ball position is actually continuous and contains all points in and around the area of the football field. By defining the belief functions on some predefined primitives as proposed in this dissertation, however, the expressiveness is limited in favor of a small set of high-level focal elements which is formed by the hypotheses defined in the involved belief functions. Consequently, adding a new basic hypothesis that results from the combination of two basic hypotheses in fact increases the number of high-level focal elements. Here too, it is vitally important for the applicability of the approach to find methods that reduce the number of basic hypotheses and in this way keep the number of high-level focal elements as small as possible.

13 Case Study II: Activity Recognition for Context-aware Systems

13.1 Overall Description

In many ubiquitous computing scenarios, the user is assumed to always carry a mobile device supporting her in performing the activities of daily living. For this purpose, the mobile device interacts with a number of diverse computing devices ranging from large computers to small, invisible processing units contained in objects of our daily life. Communication and data exchange between the involved devices is realized through wireless networking. A key characteristic of such a scenario is the mobility of the user. It implies confrontation with different context situations the devices and applications have to care for by dynamic reconfiguration in order to always provide an appropriate quality of service. For example, the applications and devices have to adapt to changing networking facilities in different environments, have to consider scarce device resources like battery power and have to react to changes in the user situation. So it may be required that the application switches to hands-free mode while the user is driving or that different privacy policies are activated depending on whether the user is in a meeting, at work or having dinner. All the context changes, either with regard to the user or with regard to the computing environment, have to be detected by appropriate sensors or derived by reasoning approaches.

In particular, adaptation to different high-level user situations is a challenging task as no sensors exist which are able to sense the user situation directly. For example, there exists no sensor that can directly tell whether the user is having dinner or sleeping. Instead, the corresponding information has to be derived by appropriate reasoning approaches from lower-level sensor information.

This simulated case study illustrates the application of Dempster-Shafer-based reasoning with HMMs as proposed in Chapter 10 for user activity recognition. It is assumed that a user moves around the city and the applications running on her mobile device have to adapt to the current activity as already mentioned above (see Figure 13.1).[1]

We consider the following five activities to be recognized: ‘Going to Cinema’, ‘Having Dinner’, ‘Doing a Presentation’, ‘Going Shopping Alone’ and ‘Going Shopping with Friends’. Of course, the location of the user is a strong indication for the corresponding activity. However, the sensors have to be expected to be unreliable to a certain extent and may also provide wrong observations. Furthermore, the location of the user may not be known precisely. For example, it might only be known that the user is currently in a big shopping center with a lot of shops but also with a number of restaurants and a cinema. In this case, further sensors are employed in order to resolve ambiguities and to stabilize the recognition.

[1] Concepts for the realization of the adaptive behaviour of the application are not discussed here and are out of the scope of this thesis.

In order to be able to perform the necessary reasoning, the user is assumed to wear a small **microphone** clipped on the jacket and **two accelerometers** hidden in the belt and in the watch. The microphone is used to provide information about persons currently speaking and the surrounding sound. A classification approach is applied to derive, for example, whether only the user speaks, only other persons speak, the user talks to other persons, etc. The measurements of the two accelerometers are used as input for a posture recognition approach. This means, information is derived from the accelerometers whether the user is currently sitting, standing, walking, etc. In addition, the mobile device can utilize a GPS sensor and/or WiFi connectivity and/or discoverable sensors in the environment to collect location information. This setting more or less corresponds to the one which was realized in the Master's thesis of Mark Blum on '*Real-time Context Recognition*' at ETH Zürich [11]. The only difference is that location information was derived using audio information, intentionally abandoning information from location sensors as we do assume.

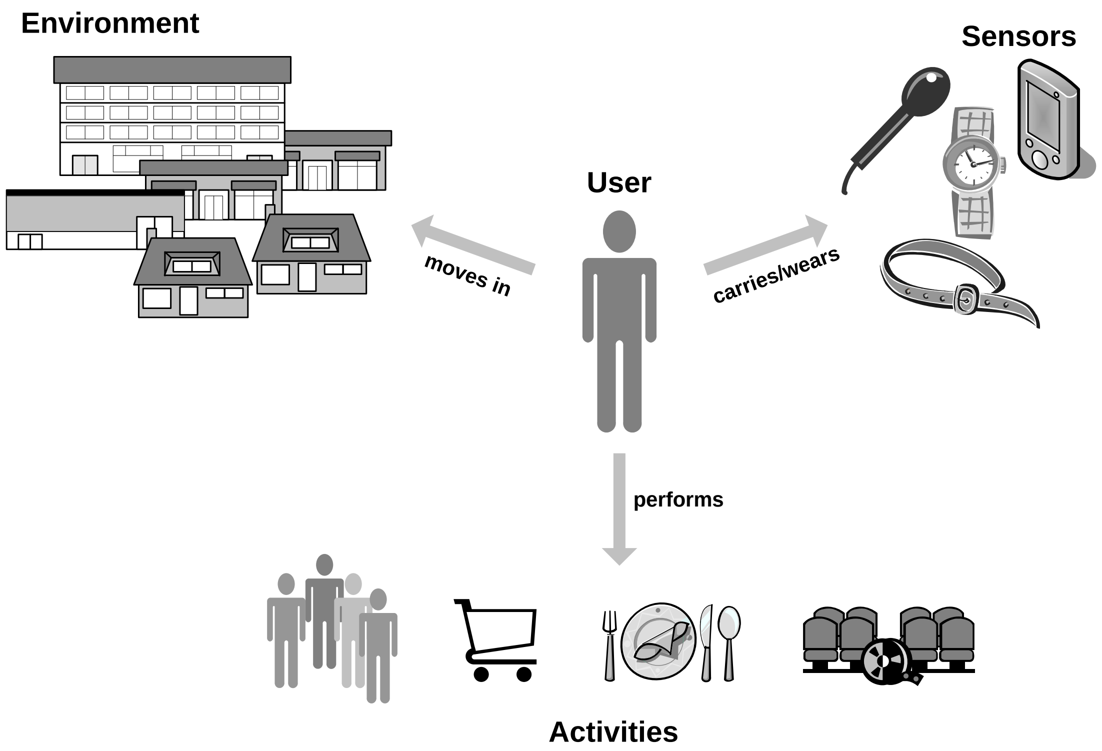

Figure 13.1: Activity Recognition for Context-aware Systems

In comparison to Blum, we adjusted the alphabets for the observations of the three sensors to better match the different activities to be recognized. The observation alphabets we used are shown in Table 13.1. Of course, these outputs are not directly provided by the sensors, but they have to be derived from the raw sensor data by possibly applying a number of preprocessing and classification steps.

As HMMs have been successfully applied for user activity recognition in many related works [123, 168, 98, 114], we decided to use them as underlying reasoning approach for this case study as well. We not only show how these models can be used to classify observation sequences as activities based on the traditional Bayesian probability theory but also how HMMs can deal with uncertain, imprecise and unreliable information represented

Location	Audio	Posture
AtHome	NoSpeech	Lying
AtOffice	MeSpeaking	Sitting
Street	MeTalkingToOthers	Standing
Shop	OthersSpeaking	Walking
Restaurant	LoudCrowd	Running
Cinema	DistantVoices	Driving
Car	unspecific	unspecific
Subway		
unspecific		

Table 13.1: Observation Alphabets for the three Employed Sensors

as Dempster-Shafer belief functions. In our case study, heterogeneity and different quality levels of sensor information are considered with regard to location information. Location information is mainly provided by the GPS sensor of the mobile device. However, inside a building, e.g. in a shopping center with several floors, GPS is unlikely to work. Here, location information is gathered by exploiting the WiFi connectivity and knowledge about the location of the corresponding access points, or by utilizing an appropriate person tracking service provided by the computing infrastructure of the building. It is obvious that these three different approaches provide information of different granularity, different precision and reliability. Also the precision and reliability of the speech sensor is heavily affected by the current environment. For example, it is difficult to decide if a user speaks alone or talks to other persons if there is much surrounding noise. All these aspects highlight the need for reasoning approaches that are able to deal with all the characteristics of heterogeneous sensor information.

13.2 Purpose of the Case Study

Whereas the first case study has demonstrated the applicability of the *Information Model*, the *Inter-Representation Operations* and the *DST-based Information Fusion* to the domain of RoboCup, this case study focuses on the *DST-based Reasoning* with heterogeneous and imperfect sensor information (see Chapter 10) and its application to the area of context-aware adaptive applications.

As already mentioned above, we use HMMs to classify observation sequences as different activities. In our framework, however, the observations of the sensors are represented by means of Dempster-Shafer belief functions, and thus the HMMs have to work on belief functions instead of probabilities. Ramasso et al. have shown in [114] how HMMs can be extended to the Dempster-Shafer Theory of Evidence and how better classification results can be obtained in comparison to their probability-based counterparts. However, Ramasso et al. have exploited the approach of *contextual discounting* [84] for creating the extended transition and observation matrices, which in fact corresponds to the incorporation of additional knowledge. Furthermore, they have proposed an adjusted classification method exploiting the additional expressivity of Dempster-Shafer Theory.

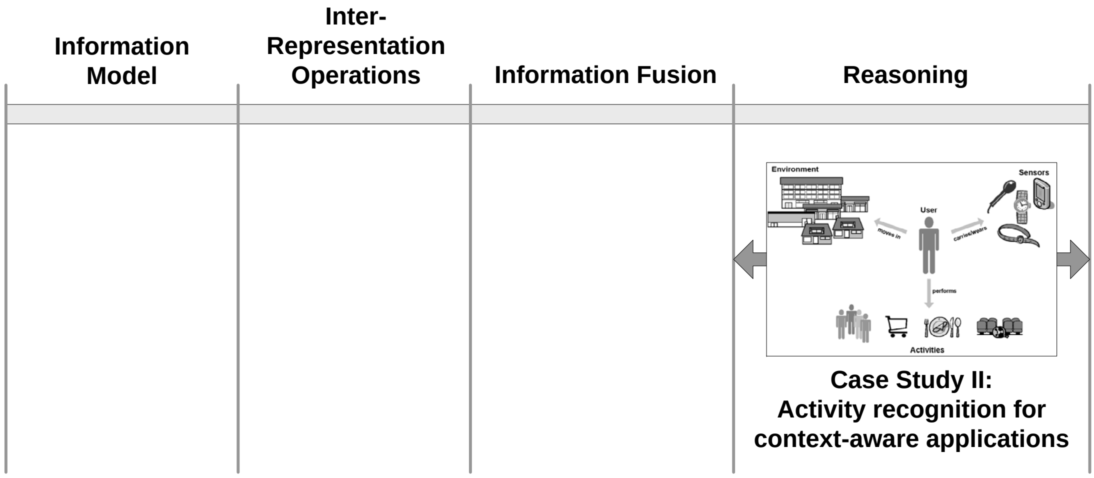

Figure 13.2: Purpose of Case Study II

In this case study, we analyze how observations represented as Dempster-Shafer belief functions perform in comparison to observations represented through probability vectors, if no contextual discounting is applied and the standard classification algorithm is used.

Another important challenge in our scenarios arises from the incorporation of a priori unknown sensors which are discovered in the ubiquitous computing environment. These dynamically incorporated sensors are likely to show different characteristics with regard to reliability and precision of the measurements compared to the sensors used in the training phase of the HMM. Therefore, we also analyze the robustness of the trained HMM with respect to changing quality of observations.

In summary, this case study is intended to provide answers to the following questions:

1. How can context-aware systems in ubiquitous computing environments profit from the approach proposed in this thesis?
2. How do observations represented as Dempster-Shafer belief functions perform in comparison to observations represented through probability vectors in a HMM-based reasoning scheme?
3. How do different quality levels of the used sensor information influence the classification results?

13.3 Implementation

Realizing a scenario as described above in real world and performing the data collection and evaluation with real sensors and real persons would require a lot of personnel resources and much time. Thus, we decided to develop a simulation that provides us with the data required for the training of the Hidden Markov Models and for testing the influence of a decreased sensor performance on the classification results. The overall approach for realizing this case study is depicted in Figure 13.3.

The simulation environment comprises manually designed HMMs, which provide models of the different activities and are used to create 'ideal' observation sequences. These 'ideal'

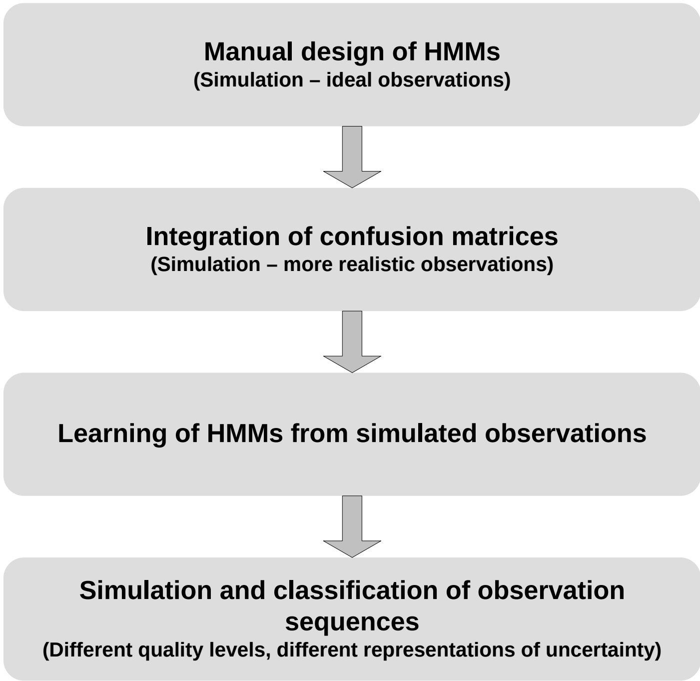

Figure 13.3: Approach for Simulation and Classification of Activities

observation sequences are then disturbed by confusion matrices in order to achieve more realistic observations. Using this kind of simulation, 100 observation sequences are generated for each of the activities which serve as a basis for the training of HMMs that are used for classification in the further analyses. It is important to note here that the trained HMMs are different from the manually designed HMMs. The rationale behind this is the fact that in a real-life scenario we would not be aware of the real model of the activities (manually designed HMMs) and would start to estimate models (trained HHMs) from a number of observation sequences. The trained HMMs are then used to investigate the effect of observations represented as Dempster-Shafer belief functions instead of probability vectors and the robustness of the HMMs with regard to different quality levels of the involved sensors. The following sections introduce the simulation environment and describe the training of the HMMs.

13.3.1 Simulation Environment

As HMMs have shown a good performance in projects with real data [123, 168, 98, 114], it is justified to assume that HMMs are also well suited to model real-life user activities. Hence, we base our simulation on manually designed HMMs, i.e. for each of the different activities to be classified a corresponding HMM is created by manually providing its parameters (number of states, initial state distribution, transition matrix and observation matrices). The different parameters are manually tuned so that the provided output reflects our subjective expectations about the particular activity. It is noteworthy here that the observation matrices are specified to simulate an ideal sensor. The resulting HMMs for the activities ‘Having Dinner’, ‘Going to Cinema’, ‘Doing a Presentation’, ‘Going Shopping Alone’, and ‘Going Shopping with Friends’ are presented in Appendix A.

In order to integrate a more realistic sensor model, we incorporate confusion matrices[2] for the different sensors. For speech and posture, the matrices are guided by the confusion matrices resulting from the classification experiments on real data performed by Blum [11]. However, it was not possible to reuse the confusion matrices of Blum directly as the sensor outputs required for the activities mentioned above are slightly different. Thus, we manually adjusted the confusion matrices but kept the overall performance of the sensors. The confusion matrix for location is not based on experiments on real data. Instead, we estimated a confusion matrix for location based on an assumed accuracy of the corresponding sensor of 10m. This roughly corresponds to the accuracy of current GPS sensors [52]. Furthermore, we estimated the size and arrangement of buildings. The different confusion matrices for speech, posture and location observations are presented in Appendix A.

Having integrated the confusion matrices, the resulting HMMs can be used to simulate an arbitrary number of observation sequences. Just as in a real-world scenario, where the data would have been collected by real persons, the simulated data are now used to train and evaluate the HMMs. Although the data have been created by simulation, they already reflect some characteristics of sensor information due to the integration of the confusion matrices. However, the influence of a decreased sensor performance on the classification results should also be investigated in this case study. Therefore, the confusion matrix for the location sensor is adjusted to provide more unreliable and more uncertain observations.

Whereas for the training the observations are assumed to be 'sharp', i.e. an observation corresponds to a single symbol (e.g. the current location is *'Restaurant'*), the classification of activities is performed using 'sharp' observations, observations represented as probability vectors as well as observations represented by Dempster-Shafer belief functions. The first two cases can be handled with traditional Bayesian probability theory and traditional HMMs. The third case illustrates the usage of HMMs which are extended so that Dempster-Shafer belief functions can be incorporated.

For the generation of the probability vector which is intended to reflect the probabilities resulting from a corresponding classification method, we applied the following approach:

- Generate the ideal observation from the manually designed HMM and select the column of the confusion matrix that belongs to the ideal observation. This column contains the relative frequencies of occurrences of the symbols to be expected.
- Repeat the following steps until all symbols have been assigned with a probability:
    1. Randomly choose an element from the set S_{na}, which comprises all symbols that have not yet been assigned a probability according to the relative frequencies of all elements of S_{na}.
    2. Let s be the selected symbol and f_s its relative frequency. Generate n random numbers which are distributed according to $\mathcal{N}(1/|S_{na}|, \alpha \cdot f_s)$, with e.g. $\alpha = 1$. Set all random numbers ≥ 1.0 to 1.0, all random numbers ≤ 0.0 to 0.0, and choose the maximum m.
    3. If $m < 1/|S_{na}|$ go back to step 2.

[2] A confusion matrix describes the relative frequencies of confusing the correct symbols with other symbols.

4. Assign symbol s with the probability $p = m \cdot r$, where $r = \min(p_{min}, 1.0 - s_p)$ (p_{min} is the smallest probability already assigned and s_p is the sum of all probabilities assigned so far).

- Normalize the resulting probability vector to sum up to 1.

From the probability vector generated in this way, the 'sharp' observation can be retrieved by simply selecting the symbol with the maximum probability. Here, steps 1 and 4 of the probability generation approach described above ensure that the relative frequencies of the symbols defined by the confusion matrix are retained. Step 1 selects the first symbol according to the relative frequencies and step 4 guarantees that it is assigned with the highest probability.

Step 2 has the effect that symbols with a high relative frequency are assigned with probabilities that differ more from the uniform distribution than the probabilities assigned to symbols with low relative frequency. This way, we model that the decision for symbols with a high relative frequency can often be made with high confidence (big difference in the classification probabilities), whereas the confidence in decisions for symbols with a small relative frequency is often quite low (nearly the same probability values).

Dempster-Shafer belief functions are generated from the probability vector by applying a simple clustering approach. All symbols with classification probabilities that differ only to a certain extent (defined by a threshold) are included in a cluster. Then the minimal probability of the elements multiplied with the number of elements is assigned as belief mass to the union set of the symbols in the cluster. The differences to the minimal probability are assigned as belief masses to the single symbols. In this way we model that we cannot really decide for a symbol if the classification probabilities are nearly the same.

13.3.2 Training of the Hidden Markov Models

Training of Hidden Markov Models is performed in this case study as usual with the help of Expectation Maximization (EM) [113]. A remaining problem, however, is the need for an initial estimate of the parameters of the HMM (number of states, initial state distribution, transition matrix and observation matrices) as starting point for the iterative EM approach. It is also noteworthy here that the initial parameters can heavily influence the quality of the training result. Of course, we could use the manually designed HMMs, but this would not reflect the reality where the ideal model is typically unknown.

As initial parameter estimation for Hidden Markov Models is not in the focus of our work, we applied a very simple approach based on the manual analysis of the histograms of the observation sequences for an activity. The histograms depict the relative frequency of observation triples, comprising the observed symbols for location, speech and posture. Consequently, peaks in the histogram correspond to triples that can be observed with high frequency. If we assume that a state is characterized by a frequently observable triple, the peaks correspond to the different states of the HMM. However, for a certain state, not only such a characteristic triple is likely to be observed but also triples which are very similar to it, i.e. with only one different symbol. This way, we get an idea of the different states and the frequencies of characteristic observations, and thus are able to build the observation matrices. The remaining parameters, the initial state distribution and the state transition matrix are estimated by creating 5 histograms covering subsequent parts of the observed sequences and examining the histograms for occurrences of the previously identified states.

Training of the Hidden Markov Models is performed using 'sharp' observations only and based on probability theory. Consequently, we have to extend the resulting models to be able to incorporate observations represented as Dempster-Shafer belief functions. This can easily be achieved by extending the transition matrix and observation matrices using the **Disjunctive Rule of Combination**. In this process, however, we do not use contextual discounting as it is done in [114]. As the Dempster-Shafer Theory works on the power set of the frame of discernment, the matrices are significantly increased in size. A conditional probability table of size $m \times n$ results in a matrix for the conditional mass assignments of $2^m \times 2^n$. Thus, it is vitally important to keep the number of states and the number of observable symbols as small as possible. Seen from the other direction, our approach is only feasible for HMMs with a small number of states and small observation symbol alphabets.

13.4 Evaluation

As basis for the evaluation we generated 100 observation sequences for each of the activities 'Cinema', 'Dinner', 'Presentation', 'Shopping Alone' and 'Shopping with Friends' using our simulation environment. These observation sequences were used to perform initial parameter estimation and training of the HMMs as described in the previous section.

Afterwards, 50 test sequences were generated for each of the different activities. In order to investigate the robustness of the trained HMMs with regard to a decreased sensor quality, 25 out of the 50 observations were generated with an adjusted confusion matrix for the location sensor in order to simulate a decreased sensor performance: the elements on the diagonal of the confusion matrix (number of correctly classified observations) were multiplied with $\alpha = 2/3$.

	sharp					**prob. vector**					**DST belief function**				
	Cinema	Dinner	Presentation	Shop. Alone	Shop. Friends	Cinema	Dinner	Presentation	Shop. Alone	Shop. Friends	Cinema	Dinner	Presentation	Shop. Alone	Shop. Friends
Cinema	25	0	0	0	0	25	0	0	0	0	25	0	0	0	0
Dinner	1	24	0	0	0	0	24	0	1	0	0	24	0	1	0
Presentation	0	0	25	0	0	0	0	25	0	0	0	0	25	0	0
Shop. Alone	2	0	0	23	0	0	0	0	25	0	0	0	0	25	0
Shop. Friends	1	3	0	21	0	0	3	0	21	1	0	3	0	21	1

Table 13.2: Confusion Matrix for Normal Sensor Quality

In this case study, we investigate whether the classification results of the HMMs improve if the sensor observations are represented as probability vectors or as Dempster-Shafer belief functions, instead of using only 'sharp' observations. Consequently, the generated sequences contain the location information represented as 'sharp' observations, as probability vectors and as Dempster-Shafer belief functions. The audio and posture information, however, is only integrated in terms of 'sharp' observations in all tests to keep the experiments traceable.

Table 13.2 and Table 13.3 illustrate the classification results for a normal location sensor quality and a decreased location sensor quality in terms of confusion matrices. The labels for the rows depict the ground truth, i.e. actual class of observations, whereas the labels for the columns exhibit the classification result for sharp observations, probability vectors and Dempster-Shafer belief functions. Hence, an entry of the table corresponds to the number of sequences of the activity defined by the row label, classified as the activity depicted by the column label.

From Table 13.2 it gets obvious that for normal sensor qualities the HMMs can discriminate reliably between the activities 'Cinema', 'Dinner' and 'Presentation' for all three representations of the location observations. Most sequences of 'Shopping Alone' are correctly classified, whereas almost all 'Shopping with Friends' sequences are mapped to 'Shopping Alone'. Thus, the HMMs are not able to discriminate between the two shopping activities. This situation is almost the same for all three possible representations of the location observations. In general, the representation of the location observations as probability vectors or as Dempster-Shafer belief function yields only slightly better results in comparison to the integration of sharp observations: with location observations based on probability vectors or belief functions, all 'Shopping Alone' activities are classified correctly.

	sharp					**prob. vector**					**DST belief function**				
	Cinema	Dinner	Presentation	Shop. Alone	Shop. Friends	Cinema	Dinner	Presentation	Shop. Alone	Shop. Friends	Cinema	Dinner	Presentation	Shop. Alone	Shop. Friends
Cinema	22	2	0	1	0	22	0	0	3	0	22	0	0	3	0
Dinner	1	19	0	4	1	0	20	0	3	2	0	20	0	3	2
Presentation	1	11	3	10	0	0	0	25	0	0	0	0	25	0	0
Shop. Alone	3	0	0	22	0	0	0	0	25	0	0	0	0	25	0
Shop. Friends	2	1	0	22	0	0	1	0	24	0	0	1	0	24	0

Table 13.3: Confusion Matrix for Decresed Sensor Quality

In case of a decreased performance of the location sensor, however, representation of the location observations as probability vectors or belief functions yields significantly better classification results (see Table 13.3). With probability vectors or belief functions, the HMMs are still able to discriminate between the activities 'Cinema', 'Dinner' and 'Presentation', whereas only 10 out of 25 'Presentation' sequences can be classified correctly when integrating sharp location observations. Consequently, the HMM-based classification is more robust against decreasing sensor performance if representing the observations as probability vectors or as belief functions, i.e. if uncertainty in the observations is taken into account during the classification.

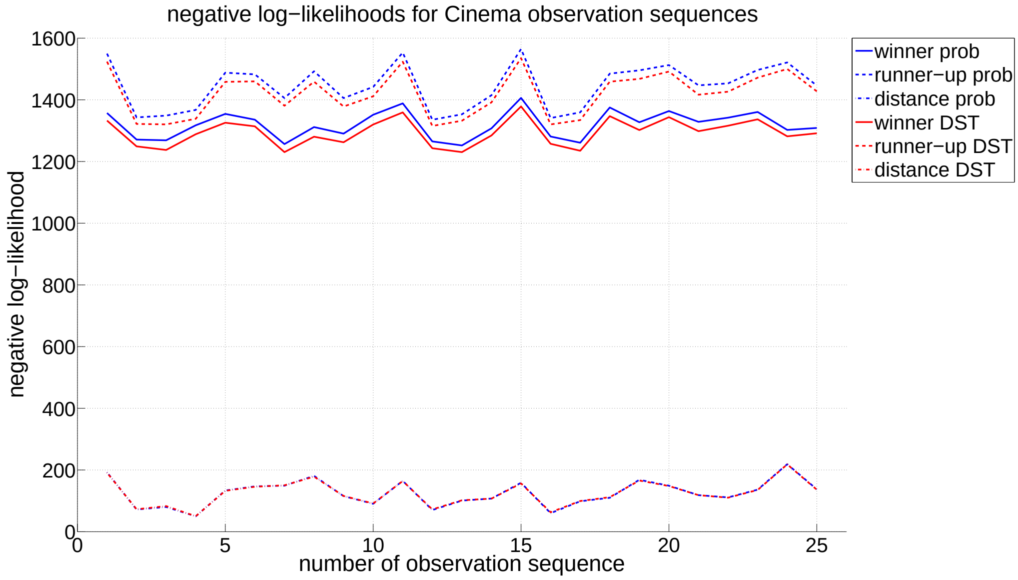

Figure 13.4: Analysis of the Negative Log-likelihoods for 'Cinema' Observation Sequences

We actually expected that Dempster-Shafer belief functions can handle a decreased sensor performance even better than probability vectors. However, probability vectors and belief functions show an identical performance and even identical confusion matrices, for normal sensor quality as well as for a decreased performance of the location sensor. At the first glance, this is a surprising observation. In order to further investigate this phenomenon, we used 25 observations of type 'Cinema' for normal sensor quality and analyzed the log-likelihoods of the HMMs that served as classification criterion.

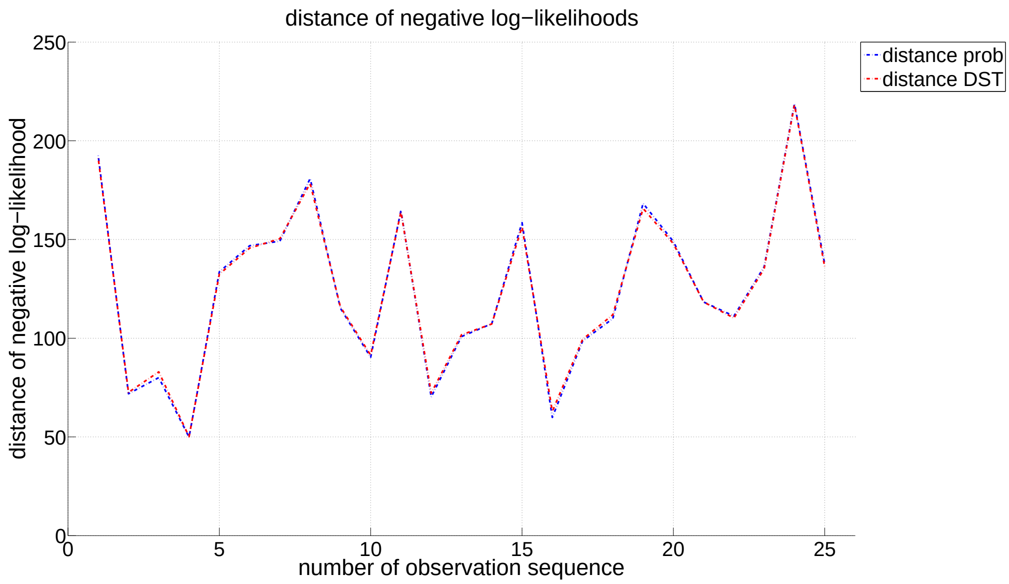

Figure 13.5: Differences of the Negative Log-likelihoods of the Winner HMM and the Runner-up HMM

Figure 13.4 illustrates the negative log-likelihoods of the winning HMM (HMM with the smallest negative log-likelihood) and of the HMM that was the runner-up in the classification process (HMM with the second smallest negative log-likelihood) for probability vectors and Dempster-Shafer belief functions. It can be observed that the curves for Dempster-Shafer belief functions are more or less just the shifted curves of the probability vectors: all values for the belief functions are roughly smaller by 30 points than the corresponding values for the probability vector. Consequently, also the differences of the negative log-likelihoods of the winner HMM and the runner-up HMM are nearly the same. The deviations are that small that even in the enlarged illustration of Figure 13.5 no real discrepancies can be observed. With identical differences, however, it is also obvious that the classification results are the same for probability vectors and the belief functions.

Our further analyses have revealed that the phenomenon described above can be explained as follows:

1. During classification, the HMMs implicitly perform an iterative filtering, which results in a probability vector or belief function for the current state of the activity at each time step. When in the DST version of the HMM the integration of observations once leads to a belief function which more or less only assigns masses to singletons, i.e. the corresponding belief function reduces to a probability vector, then all belief functions for the states at subsequent time steps also assign masses only to singletons. This is due to the following two reasons: 1) Without contextual discounting all entries $m(s^{t=k}|s^{t=k-1})$ ($s^{t=k} \subseteq S$, $|s^{t=k}| > 1$ and $s^{t=k-1} \in S$) of the transition matrix extended to a Dempster-Shafer conditional mass assignment equal to 0. 2) A conjunctive combination of a probability vector with an arbitrary belief function yields a probability vector again. In our experiments, this phenomenon could be observed after 6 time steps in average (total length of the observation sequence ≥ 80), even if the a priori state distribution was initialized with $m(\Omega^S) = 1$.
2. The update of the belief function for the states (m_s) with the belief functions that result from the observations by applying the Generalized Bayesian Theorem (m_o), has to be performed with the Conjunctive Rule of Combination:

$$(m_s \oplus m_o)(s) = \frac{1}{1-K} \sum_{s_A \cap s_B = s \neq \emptyset, s_A \subseteq S, s_B \subseteq S} m_s(s_A) \cdot m_o(s_B) \tag{13.1}$$

   which reduces to:

$$(m_s \oplus m_o)(s) = \frac{1}{1-K} m_s(s) \cdot pl_o(s) \quad \forall s \in S \tag{13.2}$$

   if m_s only assigns masses to singletons.
3. The plausibility $pl_o(s) = pl(s|o)$ for singletons s, i.e. $s \in S$ and $o \subseteq O$ (O frame of discernment for the observation), calculates as:

$$pl(s|o) = pl(o|s) = \sum_{o_i \in o} pl(o_i|s) = \sum_{o_i \in o} pl(s|o_i) \tag{13.3}$$

   as $m(.|s)$ happens to be a probability vector if the observation matrices are extended to Dempster-Shafer conditional mass assignments without contextual discounting.

We illustrate the last point with the following example. Assume a HMM with three states s_1, s_2, s_3, three possible observations o_1, o_2, o_3, and an observation matrix as provided by Table 13.4.

	$P(O\|S)$		
	$P(.\|s_1)$	$P(.\|s_2)$	$P(.\|s_3)$
o_1	0.3000	0.1000	0.6000
o_2	0.2000	0.4000	0.1000
o_3	0.5000	0.5000	0.3000

Table 13.4: Example for an Observation Matrix

An extension of the observation matrix depicted in Table 13.4 using the Disjunctive Rule of Combination results in the conditional mass assignment which is shown in Table 13.5. All entries for $m(o|s)$, $|o| > 1$ and $|s| = 1$ equal 0 as no contextual discounting is performed. From the mass assignments, we calculate the plausibility matrix shown in Table 13.6.

	$m(O\|S)$						
	$m(.\|\{s_1\})$	$m(.\|\{s_2\})$	$m(.\|\{s_3\})$	$m(.\|\{s_1,s_2\})$	$m(.\|\{s_1,s_3\})$	$m(.\|\{s_2,s_3\})$	$m(.\|\{s_1,s_2,s_3\})$
$\{o_1\}$	0.3000	0.1000	0.6000	0.0300	0.1800	0.0600	0.0180
$\{o_2\}$	0.2000	0.4000	0.1000	0.0800	0.0200	0.0400	0.0080
$\{o_3\}$	0.5000	0.5000	0.3000	0.2500	0.1500	0.1500	0.0750
$\{o_1,o_2\}$	0.0000	0.0000	0.0000	0.1400	0.1500	0.2500	0.1490
$\{o_1,o_3\}$	0.0000	0.0000	0.0000	0.2000	0.3900	0.3300	0.3390
$\{o_2,o_3\}$	0.0000	0.0000	0.0000	0.3000	0.1100	0.1700	0.1690
$\{o_1,o_2,o_3\}$	0.0000	0.0000	0.0000	0.0000	0.0000	0.0000	0.2420

Table 13.5: Observation Matrix Extended to the DST

Updating the belief function for the states at a certain time step according to the extended belief propagation algorithm (see Chapter 10), however, needs the matrix $pl(S|O)$ instead of the matrix $pl(O|S)$ as presented in Table 13.6. Since it holds $pl(s|o) = pl(o|s)$, transposing the matrix of Table 13.6 is sufficient. The resulting matrix for $pl(S|0)$ is shown in Table 13.7.

With the help of Table 13.7, the effect of Equation 13.3 becomes obvious. Consider, for example, the bold marked entries for $pl(\{s_3\}|\{o_1\})$, $pl(\{s_3\}|\{o_2\})$ and $pl(\{s_3\}|\{o_1,o_2\})$. It can easily be verified that $pl(\{s_3\}|\{o_1,o_2\}) = pl(\{s_3\}|\{o_1\}) + pl(\{s_3\}|\{o_2\})$. Furthermore, it holds $pl(s_i|o_j) = p(o_j|s_i)$ (see Table 13.4).

	$pl(O\|S)$						
	$pl(.\|\{s_1\})$	$pl(.\|\{s_2\})$	$pl(.\|\{s_3\})$	$pl(.\|\{s_1,s_2\})$	$pl(.\|\{s_1,s_3\})$	$pl(.\|\{s_2,s_3\})$	$pl(.\|\{s_1,s_2,s_3\})$
$\{o_1\}$	0.3000	0.1000	0.6000	0.3700	0.7200	0.6400	0.7480
$\{o_2\}$	0.2000	0.4000	0.1000	0.5200	0.2800	0.4600	0.5680
$\{o_3\}$	0.5000	0.5000	0.3000	0.7500	0.6500	0.6500	0.8250
$\{o_1,o_2\}$	0.5000	0.5000	0.7000	0.7500	0.8500	0.8500	0.9250
$\{o_1,o_3\}$	0.8000	0.6000	0.9000	0.9200	0.9800	0.9600	0.9920
$\{o_2,o_3\}$	0.7000	0.9000	0.4000	0.9700	0.8200	0.9400	0.9820
$\{o_1,o_2,o_3\}$	1.0000	1.0000	1.0000	1.0000	1.0000	1.0000	1.0000

Table 13.6: Plausibility Matrix for Observations

When observations are integrated, a belief function m_s for the states of the HMM at a certain time step which only assigns masses to singletons is updated according to Equation 13.2 by multiplication with $pl_O(s)$:

$$pl_O(s) = \sum_{o \subseteq O} pl(s|o) \cdot m_{Obs}(o) \tag{13.4}$$

where m_{Obs} denotes the observation represented as belief function.

	$pl(S\|O)$						
	$pl(.\|\{o_1\})$	$pl(.\|\{o_2\})$	$pl(.\|\{o_3\})$	$pl(.\|\{o_1,o_2\})$	$pl(.\|\{o_1,o_3\})$	$pl(.\|\{o_2,o_3\})$	$pl(.\|\{o_1,o_2,o_3\})$
$\{s_1\}$	0.3000	0.2000	0.5000	0.5000	0.8000	0.7000	1.0000
$\{s_2\}$	0.1000	0.4000	0.5000	0.5000	0.6000	0.9000	1.0000
$\{s_3\}$	**0.6000**	**0.1000**	0.3000	**0.7000**	0.9000	0.4000	1.0000
$\{s_1,s_2\}$	0.3700	0.5200	0.7500	0.7500	0.9200	0.9700	1.0000
$\{s_1,s_3\}$	0.7200	0.2800	0.6500	0.8500	0.9800	0.8200	1.0000
$\{s_2,s_3\}$	0.6400	0.4600	0.6500	0.8500	0.9600	0.9400	1.0000
$\{s_1,s_2,s_3\}$	0.7480	0.5680	0.8250	0.9250	0.9920	0.9820	1.0000

Table 13.7: Plausibility of States for a Given Observation

At this point we have to recall how the mass assignment m_{Obs} is derived from the observation represented as probability vector p_{Obs}: if $n > 1$ observations show nearly the same probability value, the minimal probability value of these observations multiplied with n is assigned to the union set of these observations. Therefore, it approximately holds:

$$m_{Obs}(o) \approx \sum_{o_i \in o, 1 \leq i \leq n} p_{Obs}(o_i) \approx n \cdot p_{Obs}(o_1) \quad \text{for } |o| = n > 1 \tag{13.5}$$

Hence, if the mass is assigned to the union set $o = \bigcup_{1 \leq i \leq n} o_i$, the corresponding summand of $pl_O(s)$ is given by:

$$pl(s|o) \cdot m_{Obs}(o) \approx n \cdot p_{Obs}(o_1) \cdot pl(s|o) \approx n \sum_{1 \leq i \leq n} p(o_i|s) \cdot p_{Obs}(o_i) \tag{13.6}$$

If the observation is represented as a probability vector, the update is performed according to the theorem of Bayes by multiplication of $p_s(s)$ with $p_O(s) = \sum_{o \in O} p(o|s) \cdot p_{Obs}(o)$. In this case, the summands corresponding to Equation 13.6 are given by $\sum_{1 \leq i \leq n} p(o_i|s) \cdot p_{Obs}(o_i)$. Hence, the summands used for the update step of the HMM just differ by the constant factor n.

If we now assume[3] that there are a few dominating values in p_{Obs}, which results in a mass assignment to the union set of the corresponding singletons, all other summands can be neglected and also the plausibilities used for the update differ only by the constant factor n. Consequently, a corresponding update step always results in a decrease of the negative log-likelihood by $\log(n)$ in comparison to the probability vector-based integration. Considering our example of 'Cinema' observations, this also means that each assignment of a mass to the union set of n singletons results in a shift of the negative log-likelihoods for the winner HMM and also for the other HMMs by $\log(n)$. If the average number of such assignments and the relative frequencies of the values for n are approximately the same as in our experiments, even the total shift of the curves is constant for all observation sequences.

In summary, the considerations presented above revealed that representing (partial) ignorance by assigning masses to union sets of singletons instead of uniformly assigning probabilities to the singletons has no effect on the classification result in the experiments performed so far.

We now assume that the user for which activity recognition has to be performed, watches a movie in a cinema located in a shopping center with 15 shops, 3 restaurants and 1 cinema. GPS is not available in the shopping center. The only reliable information with regard to location is the fact that the user is inside the shopping center. We now represent the ignorance about the real location of the user by a uniform distribution over all different shops, restaurants and the cinema. This yields:

$$\begin{array}{lcll} p_{Obs}(\text{'Shop i'}) & = & 1/19, & 1 \leq i \leq 15 \\ p_{Obs}(\text{'Restaurant j'}) & = & 1/19, & 1 \leq j \leq 3 \\ p_{Obs}(\text{'Cinema'}) & = & 1/19 & \end{array}$$

[3]This assumption has proven to hold in our experiments but it is unlikely to be true in the general case.

If an IRO or reasoning step is performed that abstracts the concrete shops, restaurants and the cinema only by the corresponding categories, we get:

$$\begin{array}{lcl} p^*_{Obs}(\text{‘Shop’}) & = & 15/19 \\ p^*_{Obs}(\text{‘Restaurant’}) & = & 3/19 \\ p^*_{Obs}(\text{‘Cinema’}) & = & 1/19 \end{array}$$

If we represent the ignorance instead by

$$m_{Obs}(\{\text{‘Shop 1’}, \ldots, \text{‘Shop 15’}, \text{‘Restaurant 1’}, \ldots, \text{‘Restaurant 3’}, \text{‘Cinema’}\}) = 1$$

the ignorance is naturally maintained across the IRO to

$$m^*_{Obs}(\{\text{‘Shop’}, \text{‘Restaurant’}, \text{‘Cinema’}\}) = 1.$$

In order to show the effect of the different representations of ignorance on the classification result, we generated 25 observation sequences for the ‘Cinema’ activity. The location information inside the shopping center was given by the sharp observation ‘Shop’ (maximum probability in the probability vector p^*_{Obs}), the probability vector p^*_{Obs} and the belief function m^*_{Obs}. The corresponding classification results are presented in Table 13.8.

	sharp					**prob. vector**					**DST belief fucntion**				
	Cinema	Dinner	Presentation	Shop. Alone	Shop. Friends	Cinema	Dinner	Presentation	Shop. Alone	Shop. Friends	Cinema	Dinner	Presentation	Shop. Alone	Shop. Friends
Cinema	13	5	0	7	0	6	11	0	8	0	21	3	0	1	0

Table 13.8: Confusion of ‘Cinema’ Observations (adjusted experiment)

Table 13.8 shows that for sharp observations and probability vectors, the HMMs are not able to reliably recognize the ‘Cinema’ activity any more, whereas with belief functions, 21 out of 25 observation sequences are classified correctly. Very interesting is the fact that the number of correctly classified sequences is higher for sharp observations than for observations represented as probability vectors. However, this can be explained by the integration of the observations for audio and posture which do not match the shopping activities but vote in favor of the dinner activity. In the probability vector-based representation the location restaurant is also assigned with a probability 3/19. This is sufficient for shifting a number of classifications from ‘Cinema’ to ‘Dinner’.

As in this experiment, the mass assignment to the union set of singletons does not correspond to uniformly distributed probabilities, the constant shift of the negative log-likelihoods as in the previous experiments cannot be observed any more. Also the distances of the winner HMM to the runner-up HMM differ to a large extent (see Figure 13.6 and Figure 13.7). This is also in line with the observation that the representation of the observations influences the classification results in this experiment.

Figure 13.6: Analysis of the Negative Log-likelihoods for 'Cinema' Observation Sequences (adjusted experiment)

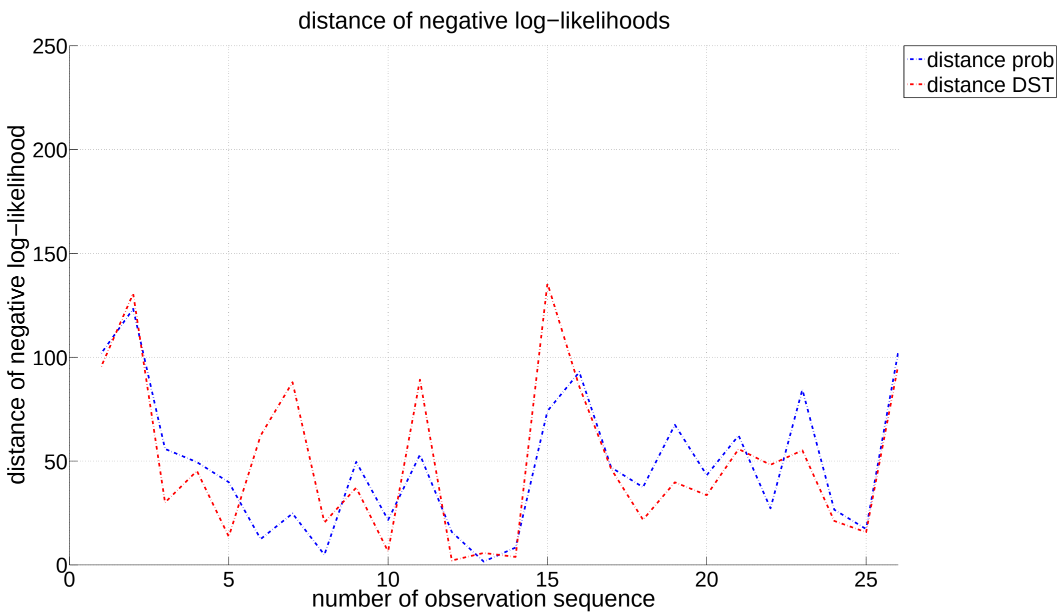

Figure 13.7: Differences between the Negative Log-likelihoods of the Winner HMM and the Runner-up HMM (adjusted experiment)

13.5 Summary and Discussion

In this case study, we have investigated the influence of representing observations in an HMM-based activity recognition approach as 'sharp' observations, as probability vectors and as Dempster-Shafer belief functions. Our experiments revealed that the classification with probability vectors and belief functions is more robust against decreasing sensor performance.

However, in our case study, where we have not applied contextual discounting [84], probability vectors and belief functions yield identical classification results, if a mass assignment to the union set of singletons corresponds to uniformly distributed probabilities of the singletons. In this case, the negative log-likelihoods for all HMMs are shifted by a constant value. This also means that if we do not apply contextual discounting for the extension of the transition and observation matrices, we do not have to use HMMs extended to the DST and can achieve classification results of the same quality with traditional HMMs. We just have to convert the Dempster-Shafer belief functions to a probability vector using the pignistic transformation (see Section 3.5), which distributes the masses assigned to union sets of singletons uniformly to the involved singletons.

Our last experiment, however, highlights again that it is more appropriate to represent ignorance through Dempster-Shafer belief functions instead of using a uniform distribution if IROs have to be expected. Exploiting the expressivity of the Dempster-Shafer belief functions, partial ignorance is naturally maintained across the IRO. Nevertheless, we can still do the actual reasoning with the traditional HMMs based on probabilities if we apply the pignistic transformation before the integration of the observations.

Still, reasoning with HMMs extended to Dempster-Shafer belief functions can yield better classification results if contextual discounting is applied and a modified classification criterion is used, as shown by Ramasso et al. [114]. Here, our case study revealed that contextual discounting is of major importance for the success of Ramasso's approach. An investigation, however, how contextual discounting and the modified classification criterion influence the classification results in the scenario of our case study remains for future work.

It is also noteworthy here that in this case study, we have classified whole observation sequences for different activities, i.e. we have classified only once at the end of an observation sequence. In a real context-aware application which has to adapt its behaviour according to the current user activity, classification has to be performed at each time step when new observations are integrated. This also means that the evolution of the negative log-likelihoods has to be tracked for the different models over time and criteria have to found that facilitate a stable and dependable adaptation decision. The search for solutions to these problems also remains for future work.

14 Conclusions

14.1 Requirements Revisited

In this dissertation, we have identified a number of unsolved challenges imposed by current distributed computing environments. A priori unknown information providers and consumers dynamically appear and disappear in the environment, and the involved software components and systems are likely to be independently developed with minimal interaction of the different development teams. Consequently, a number of heterogeneity issues arise which have to be handled at runtime. Furthermore, the need to incorporate imprecise, uncertain and unreliable sensor information constitutes an additional key characteristic of current distributed computing environments.

The identified challenges and key characteristics directly translate to a number of requirements which have been collected in Chapter 6. The main objective of our work was to provide a comprehensive approach fulfilling all requirements at the same time. Thus, in this section we summarize how the single requirements are addressed by the proposed solution.

1. **Dynamic integration of a priori unknown information consumers and providers**

   An abstract architecture specification has been presented in Chapter 11 which comprises services for semantic discovery and matching of information consumers and providers as well as a Mediation Engine responsible for establishing the links between them. Information offers and requests are defined with the help of the ontology-based Information Model and the IORL introduced in Section 7.3. The Mediation Engine is also concerned with bridging the heterogeneity issues that are caused by the independent development of the involved software components and services.

2. **Establishment of a common vocabulary**

   A common vocabulary which serves as baseline for the different development teams is established utilizing our Information Model (see Chapter 7). We have not presented a concrete ontology for a particular application domain, but instead our Information Model constitutes an ontology meta-model that can be used for the specification of a wide range of ontologies tailored for specific application domains. The Information Model has also been designed to allow easy integration of already available ontologies.

3. **Mechanisms for the semantic interpretation of data structures**

   The Information Model not only allows for the definition of a conceptualization of the real and logical entities of the world, but also supports the definition of information types (scopes) and different representations for them. Data structures are defined in terms of representations which comprise further scopes as their dimensions. Scopes and representations are associated with meta-data to describe the imperfect nature of the sensor data. In this respect, the data structures used in a particular application as well as the meta-data to define the quality of the information are semantically interpretable.

4. **Support for conversion between data representations**

   Conversions between different representations of a piece of information are realized with the help of Inter-Representation Operations (IROs), which are described in Chapter 8. Here, we have particularly focused on the problem of preserving the measures for impreciseness and uncertainty across the possibly non-linear transformations. The Unscented Transformation is applied to propagate multivariate normal distributions through non-linear functions. Furthermore, we have shown how (uniform) value ranges can be transformed based on a second-order Taylor approximation of the transformation function. The developer is only required to provide a method for a point-to-point conversion and is not confronted with the problem of maintaining the measures for uncertainty and impreciseness at all.

5. **Expression of information offers and needs**

   For expression of information offers and needs we have presented the IORL in Section 7.3, which is an XML-based language and which allows to define constraints on the entities, scopes, representations and the data values of the provided or requested information. Furthermore, it supports the specification of meta-data attributes like sensor reliability or the frequency with which the information is provided/requested. A corresponding approach for matching of information offers and requests has been discussed in Section 7.3.2.

6. **Support of competitive sensor fusion**

   Competitive sensor fusion is realized with the help of the Dempster-Shafer Theory of Evidence (see Chapter 9). As we base the specification of the belief functions on the primitives mean/covariance matrix, value range and uniform value range, the involved hypotheses can be considered as high-level focal elements. This allows us to apply the Dempster-Shafer Theory also on domains that involve continuous scales. We have also presented concepts for a consistent calculation of the conflict across several fusion steps. A difficult problem in the simplification of complex hypotheses is caused by the fact that the distributive law does not hold for the *Conjunctive Rule of Combination* and the *Disjunctive Rule of Combination*. Here, we have derived and proven a set of new rules that still allow a simplification in a number of cases.

7. **Provisioning of reasoning schemes that consider the imperfect nature of sensor data**

   In Section 10.1, we have shown how Dempster-Shafer belief functions can be incorporated in traditional probabilistic reasoning schemes like Hidden Markov Models, Naïve Bayes Classifiers and Polytree Bayesian Networks. For this purpose, Pearl's belief propagation algorithm was revisited in the realm of the Dempster-Shafer Theory and the Transferable Belief Model. Furthermore, it was described how belief functions can be considered in logic-based reasoning schemes and particularly in the evaluation of first-order logic formulas. In this respect, a number of approaches have been presented that allow a consideration of the imperfect nature of sensor data in different reasoning schemes.

From the discussion above, we can conclude that the solution proposed in this dissertation is able to address all challenges and requirements identified in Section 1.2 and Chapter 6 respectively. Thus, it is justified to claim that this dissertation provides a comprehensive approach for information exchange and fusion in dynamic and heterogeneous distributed environments.

14.2 Contributions

Although the theories serving as baseline for our methods are already well established, our work includes a number of enhancements and solutions to problems which arise if the different concepts have to be incorporated to a coherent approach. For example, we base the specification of the belief functions on some basic primitives that allow to work on a set of high-level focal elements formed by the hypotheses and facilitate the application of simple rules for the simplification of complex hypotheses. As another example, we have shown how belief functions can be handled in the evaluation of first-order logic formulas.

However, we do not see the main contribution of this work in specific theoretical concepts we have proposed. Instead, we claim the provisioning of a comprehensive solution which addresses all the identified requirements of current and future distributed computing environments as main contribution of this work. In particular, the state of the art in the area of *Context Management and Reasoning in Ubiquitous Computing* and in the area of *Cooperative Teams of Heterogeneous Mobile Robots* is enhanced.

Our analyses of related work have revealed that several current context management and reasoning systems address the challenges resulting from a dynamic computing environment with appearing and disappearing context providers and consumers. Other approaches facilitate the independent development of context components by establishing a common vocabulary based on ontologies. However, together with the ASC/CoOL model our Information Model stands out because they are, to the best of our knowledge, the only approaches that explicitly address the problem of heterogeneous representations of context information and conversions between them. All other approaches neglect this important issue. We have used many concepts of ASC/CoOL as a baseline for our Information Model, but we have enhanced it to allow a more elaborate specification of Inter-Representation Operations and a representation of uncertainty, impreciseness and unreliability of context information in terms of Dempster-Shafer belief functions.

Although some context management frameworks support competitive context fusion and provide reasoning schemes which consider the imperfect nature of context information, they lack the ability to represent partial or complete ignorance. The Dempster-Shafer Theory of Evidence supports modelling partial or complete ignorance. However, the analyzed works applying Dempster-Shafer Theory fail to incorporate it with well-established reasoning schemes required to derive high-level context information.

In the area of autonomous robots, state-of-the-art approaches for cooperative object localization are mainly based on probability theory and utilize Bayesian filtering techniques as underlying method for sensor fusion. Hence, they also suffer from the inability of traditional probability theory to explicitly model partial and complete ignorance. As shown in both of our case studies, this causes problems if non-linear transformations have to be performed. Besides, most approaches are tailored to a specific task or application domain and do not provide a generic sensor fusion approach. Generic and reusable sensor fusion and reasoning schemes in the area of autonomous mobile robots, however, assume a statically composed system and do not consider the additional challenges arising from a dynamic computing environment and from independent development of information providers and consumers.

Related works that address interoperability and cooperation of independently developed and heterogeneous autonomous mobile systems view interoperability and cooperation mainly from a high-level perspective and consider the exchange of information at the symbolic level only. Exchange of sensor data with its measures of uncertainty and impreciseness as well as appropriate sensor fusion and reasoning schemes are supported only to a very limited extent.

Closest to our vision of autonomous mobile systems, which are dynamically composed and configured at runtime is the PEIS approach (see Section 6.2.6). It meets many requirements arising from a dynamic computing environment and even presents approaches for data fusion. However, a common vocabulary serving as a baseline for independent development is provided only implicitly, and semantically interpretable data structures and the transformation of measures of uncertainty and impreciseness are supported only to a very limited extent.

In conclusion, we see four main contributions of our work to the state of the art:

1. **A comprehensive solution** for information exchange and fusion in dynamic and heterogeneous distributed environments has been designed combining a number of available theoretical concepts to a coherent approach.
2. Our ontology-based Information Model along with the associated IORL represents **a new context modelling approach** which is tailored to support heterogeneous context sensors and reasoners and which allows to express the imperfect nature of sensor information through Dempster-Shafer belief functions.
3. The proposed information fusion method along with the enhancement of the probabilistic reasoning schemes to the Dempster-Shafer Theory describes a **new generally applicable context aggregation and reasoning method**.
4. The work described in this dissertation introduces **new concepts for the realization of teams of heterogeneous autonomous robots**, which are composed dynamically at runtime and where interaction among the different development teams is minimal.

14.3 Insights from the Case Studies

The solution to realize information exchange and fusion in dynamic and heterogeneous distributed computing environments proposed in this dissertation has been evaluated with two case studies. The first case study was concerned with the dynamic establishment of a cooperative team of heterogeneous autonomous robots in the RoboCup environment. The second case study was settled in the area of activity recognition for context-aware adaptive applications.

As part of both case studies we have investigated whether the application of Dempster-Shafer Theory of Evidence has benefits over traditional approaches based on probability theory, or just increases the complexity of the involved conversion, fusion and reasoning steps. Both case studies have revealed that the application of the Dempster-Shafer Theory of Evidence, at least in the two specific scenarios, does not yield significantly better results if just the fusion and reasoning steps are considered and possibly required IROs are neglected. If non-linear IROs are involved, however, the inability of traditional probability theory to explicitly model partial and complete ignorance leads to problems, since e.g. a uniform distribution used to represent ignorance is not maintained as a uniform distribution across non-linear transformations. In this case, probability theory does not allow to distinguish between ignorance represented through a uniform distribution and the knowledge that a random variable is uniformly distributed. Case study I has also revealed that our Dempster-Shafer-based solution, although not tailored to the RoboCup domain, can compete with state-of-the-art approaches in this domain and that with the concept of discounting we allow to consider sensor reliability and the outdatedness of sensor information in a straight-forward manner.

The evaluation of the performance of the prototype implementation used as basis for case study I has shown that the proposed solution is applicable for the RoboCup domain, where decisions have to be made in very short time intervals of a few milliseconds. However, just as expected it turned out that our solution approach does not scale with the number of information providers: The time required for the fusion step grows exponentially with the number of information providers in the worst case. Therefore, the proposed information fusion method is only applicable for about 12 robots providing a single basic hypothesis each, or for about 6 robots if they define belief functions with two basic hypotheses.

Furthermore, case study II has shown that activity recognition based on Hidden Markov Models is more robust against decreasing sensor performance if the observations are represented by probability vectors or Dempster-Shafer belief functions instead of integrating 'sharp' observations comprising only a single symbol. This also means that the classification benefits from explicit modelling of uncertainty and its consideration in the corresponding reasoning tasks. In contrast to Ramasso et al. in [114], we have not observed advantages of using HMM extended to DST instead of the pure probabilistic scheme for the classification tasks. Here, our analyses revealed that the results of Ramasso et al. can be explained by the application of *contextual discounting* [84], which in fact means the incorporation of additional knowledge, and by the usage of a modified classification criterion.

14.4 Outlook and Future Work

The work presented in this dissertation provides a viable and comprehensive solution to information exchange and fusion in dynamic and heterogeneous environments. Still there are a number of open questions which are subject of future work. In the following paragraphs we will discuss the most important ones with respect to the main building blocks of our solution approach.

Information model and IORL
The information model has already proven its viability as context model in the European research project MUSIC [90, 116], and the IORL has been inspired by the MUSIC Context Query Language (MUSIC CQL) [117]. However, so far we have presented only preliminary ideas for the matching of information offers and requests based on *semantic tableaus*. This approach has to be elaborated and its viability has to be tested in future work.

Inter-Representation Operations
In Chapter 8, we have elaborated on methods to maintain the measures for uncertainty and impreciseness across non-linear functions that are continuous and involve only continuous scales. Possibilities for realizing conversions from a continuous space to a discrete space or vice versa are still an open issue. In our future work, methods have to be found which enable to perform such transformations in an efficient manner as well. Furthermore, complex hypotheses involving the negation and intersection of basic hypotheses cannot be maintained across the transformation. The search for efficient solutions or appropriate approximations is also a subject of future work.

Information fusion
A major problem to be faced in our Dempster-Shafer-based information fusion method is that the distributive law does not hold for the Conjunctive and Disjunctive Rule of Combination. We have provided a set of rules that allow a simplification of complex hypotheses in a number of cases, but still we are not able to perform all necessary simplifications. In this context, appropriate approximations have to be found which allow to abstract the intersection of two basic hypotheses based on the primitives (uniform) value range and mean/covariance matrix. The scalability issues of our information fusion method constitute another important point to be addressed in future work. Here, appropriate methods have to be researched that allow a more compact representation of belief functions involving a smaller set of basic hypotheses and thus a smaller number of high-level focal elements.

Reasoning schemes
In Chapter 10, we have presented methods that allow the incorporation of the Dempster-Shafer Theory of Evidence in reasoning schemes based on probability theory and first-order logic. However, the proposed reasoning schemes rely on Dempster-Shafer belief assignments which are defined on a small frame of discernment. Therefore, methods are required that allow the abstraction of complex belief functions involving a number of complex and basic hypotheses in a suitable manner. Our case study on activity recognition has also shown that the classification results cannot be improved by extending the HMM theory to DST without applying the approach of contextual discounting and utilizing a modified classification criterion. An investigation, however, to which extent the classification results can be improved remains for future work.

Part IV

Appendices

A Hidden Markov Models for Activity Recognition

In this chapter, first we present the Hidden Markov Models which have been used in *Case Study I* for the generation of observation sequences, along with the confusion matrices which have been employed in order to obatain more realistic sensor outputs. Afterwards, the HMMs that have been estimated from a number of observation sequences and have been used in the classification process are shown.

A.1 Manually Designed Model for the 'Cinema' Activity

This section presents the manually designed HMM for the 'Cinema' activity in terms of the initial state distribution (Table A.1), the state transition matrix (Table A.2) and the observation matrices for the sensor outputs (Table A.3 to Table A.5).

$P_1(s)$						
s_1	s_2	s_3	s_4	s_5	s_6	s_7
1.0000	0.0000	0.0000	0.0000	0.0000	0.0000	0.0000

Table A.1: Initial State Distribution for the Manually Designed 'Cinema' Model

$P(s^{(t+1)}\|s^{(t)})$							
	s_1	s_2	s_3	s_4	s_5	s_6	s_7
s_1	0.9300	0.0000	0.0000	0.0000	0.0000	0.0000	0.0000
s_2	0.0700	0.9000	0.0000	0.0000	0.0000	0.0000	0.0000
s_3	0.0000	0.0500	0.9800	0.0000	0.0000	0.1500	0.0000
s_4	0.0000	0.0000	0.0200	0.9800	0.0000	0.0000	0.0000
s_5	0.0000	0.0000	0.0000	0.0200	0.9800	0.0000	0.0000
s_6	0.0000	0.0500	0.0000	0.0000	0.0000	0.8500	0.0000
s_7	0.0000	0.0000	0.0000	0.0000	0.0200	0.0000	1.0000

Table A.2: State Transition Matrix for the Manually Designed 'Cinema' Model

$P(o_L \mid s)$							
	s_1	s_2	s_3	s_4	s_5	s_6	s_7
AtHome	0.0000	0.0000	0.0000	0.0000	0.0000	0.0000	0.0000
AtOffice	0.0000	0.0000	0.0000	0.0000	0.0000	0.0000	0.0000
Street	0.7000	0.0000	0.0000	0.0000	0.0000	0.0000	0.4000
Shop	0.0000	0.0000	0.0000	0.0000	0.0000	0.0000	0.0000
Restaurant	0.0000	0.0000	0.0000	0.0000	0.0000	0.0000	0.0000
Cinema	0.0000	1.0000	1.0000	1.0000	1.0000	1.0000	0.0000
Car	0.1000	0.0000	0.0000	0.0000	0.0000	0.0000	0.3000
Subway	0.1000	0.0000	0.0000	0.0000	0.0000	0.0000	0.3000
unspecific	0.1000	0.0000	0.0000	0.0000	0.0000	0.0000	0.0000

Table A.3: Location Observation Matrix for the Manually Designed 'Cinema' Model

$P(o_A \mid s)$							
	s_1	s_2	s_3	s_4	s_5	s_6	s_7
NoSpeech	0.0250	0.1250	0.1000	0.1000	0.1000	0.4950	0.0250
MeSpeaking	0.1000	0.0500	0.0250	0.0250	0.0250	0.0500	0.2500
MeTalkingToOthers	0.1000	0.2500	0.0250	0.0250	0.0250	0.0500	0.3000
OthersSpeaking	0.1000	0.2500	0.2800	0.2800	0.2800	0.1000	0.1000
LoudCrowd	0.1500	0.1500	0.0250	0.0250	0.0250	0.0050	0.1000
DistantVoices	0.4250	0.1250	0.2500	0.2500	0.2500	0.1000	0.1250
unspecific	0.1000	0.0500	0.2950	0.2950	0.2950	0.2000	0.1000

Table A.4: Audio Observation Matrix for the Manually Designed 'Cinema' Model

$P(o_P \mid s)$							
	s_1	s_2	s_3	s_4	s_5	s_6	s_7
Lying	0.0000	0.0000	0.0000	0.0000	0.0000	0.0000	0.0000
Sitting	0.0500	0.0000	0.9750	0.9750	0.9750	0.3000	0.0500
Standing	0.1000	0.8000	0.0200	0.0200	0.0200	0.3000	0.1000
Walking	0.6500	0.1500	0.0000	0.0000	0.0000	0.3000	0.6500
Running	0.0500	0.0000	0.0000	0.0000	0.0000	0.0000	0.0500
Driving	0.2500	0.0000	0.0000	0.0000	0.0000	0.0000	0.2500
unspecific	0.0000	0.0500	0.0050	0.0050	0.0050	0.1000	0.0000

Table A.5: Posture Observation Matrix for the Manually Designed 'Cinema' Model

A.2 Manually Designed Model for the 'Dinner' Activity

This section presents the manually designed HMM for the 'Dinner' activity in terms of the initial state distribution (Table A.6), the state transition matrix (Table A.7) and the observation matrices for the sensor outputs (Table A.8 to Table A.10).

$P_1(s)$					
s_1	s_2	s_3	s_4	s_5	s_6
1.0000	0.0000	0.0000	0.0000	0.0000	0.0000

Table A.6: Initial State Distribution for the Manually Designed 'Dinner' Model

$P(s^{(t+1)}\|s^{(t)})$						
	s_1	s_2	s_3	s_4	s_5	s_6
s_1	0.9200	0.0000	0.0000	0.0000	0.0000	0.0000
s_2	0.0800	0.9400	0.0000	0.0000	0.0000	0.0000
s_3	0.0000	0.0350	0.9400	0.0000	0.1500	0.0000
s_4	0.0000	0.0000	0.0350	0.9400	0.1000	0.0000
s_5	0.0000	0.0250	0.0250	0.0250	0.7500	0.0000
s_6	0.0000	0.0000	0.0000	0.0350	0.0000	1.0000

Table A.7: State Transition Matrix for the Manually Designed 'Dinner' Model

$P(o_L\|s)$						
	s_1	s_2	s_3	s_4	s_5	s_6
AtHome	0.0000	0.0000	0.0000	0.0000	0.0000	0.0000
AtOffice	0.0000	0.0000	0.0000	0.0000	0.0000	0.0000
Street	0.7000	0.0000	0.0000	0.0000	0.0000	0.4000
Shop	0.0000	0.0000	0.0000	0.0000	0.0000	0.0000
Restaurant	0.0000	1.0000	1.0000	1.0000	1.0000	0.0000
Cinema	0.0000	0.0000	0.0000	0.0000	0.0000	0.0000
Car	0.1000	0.0000	0.0000	0.0000	0.0000	0.3000
Subway	0.1000	0.0000	0.0000	0.0000	0.0000	0.3000
unspecific	0.1000	0.0000	0.0000	0.0000	0.0000	0.0000

Table A.8: Location Observation Matrix for the Manually Designed 'Dinner' Model

$P(o_A\|s)$						
	s_1	s_2	s_3	s_4	s_5	s_6
NoSpeech	0.0250	0.1250	0.1250	0.1250	0.5000	0.0250
MeSpeaking	0.2500	0.3000	0.3000	0.3000	0.0500	0.2500
MeTalkingToOthers	0.3000	0.3000	0.3000	0.3000	0.0500	0.3000
OthersSpeaking	0.1000	0.1000	0.1000	0.1000	0.1000	0.1000
LoudCrowd	0.1000	0.0000	0.0000	0.0000	0.0000	0.1000
DistantVoices	0.1250	0.1250	0.1250	0.1250	0.1000	0.1250
unspecific	0.1000	0.0500	0.0500	0.0500	0.2000	0.1000

Table A.9: Audio Observation Matrix for the Manually Designed 'Dinner' Model

$P(o_P|s)$						
	s_1	s_2	s_3	s_4	s_5	s_6
Lying	0.0000	0.0000	0.0000	0.0000	0.0000	0.0000
Sitting	0.0000	0.9000	0.9000	0.9000	0.3000	0.0000
Standing	0.1000	0.0250	0.0250	0.0250	0.3000	0.1000
Walking	0.7000	0.0250	0.0250	0.0250	0.3000	0.7000
Running	0.0500	0.0000	0.0000	0.0000	0.0000	0.0500
Driving	0.2500	0.0000	0.0000	0.0000	0.0000	0.2500
unspecific	0.0000	0.0500	0.0500	0.0500	0.1000	0.0000

Table A.10: Posture Observation Matrix for the Manually Designed 'Dinner' Model

A.3 Manually Designed Model for the 'Presentation' Activity

This section presents the manually designed HMM for the 'Presentation' activity in terms of the initial state distribution (Table A.11), the state transition matrix (Table A.12) and the observation matrices for the sensor outputs (Table A.13 to Table A.15).

$P_1(s)$					
s_1	s_2	s_3	s_4	s_5	s_6
1.0000	0.0000	0.0000	0.0000	0.0000	0.0000

Table A.11: Initial State Distribution for the Manually Designed 'Presentation' Model

$P(s^{(t+1)}|s^{(t)})$						
	s_1	s_2	s_3	s_4	s_5	s_6
s_1	0.9400	0.3000	0.0000	0.0000	0.0000	0.0000
s_2	0.0200	0.6600	0.0000	0.0000	0.0000	0.0000
s_3	0.0400	0.0400	0.0500	0.0000	0.0000	0.0000
s_4	0.0000	0.0000	0.9500	0.9700	0.0000	0.0000
s_5	0.0000	0.0000	0.0000	0.0300	0.8000	0.0000
s_6	0.0000	0.0000	0.0000	0.0000	0.2000	1.0000

Table A.12: State Transition Matrix for the Manually Designed 'Presentation' Model

$P(o_L\|s)$						
	s_1	s_2	s_3	s_4	s_5	s_6
AtHome	0.0000	0.0000	0.0000	0.0000	0.0000	0.0000
AtOffice	1.0000	1.0000	1.0000	1.0000	1.0000	1.0000
Street	0.0000	0.0000	0.0000	0.0000	0.0000	0.0000
Shop	0.0000	0.0000	0.0000	0.0000	0.0000	0.0000
Restaurant	0.0000	0.0000	0.0000	0.0000	0.0000	0.0000
Cinema	0.0000	0.0000	0.0000	0.0000	0.0000	0.0000
Car	0.0000	0.0000	0.0000	0.0000	0.0000	0.0000
Subway	0.0000	0.0000	0.0000	0.0000	0.0000	0.0000
unspecific	0.0000	0.0000	0.0000	0.0000	0.0000	0.0000

Table A.13: Location Observation Matrix for the Manually Designed 'Presentation' Model

$P(o_A\|s)$						
	s_1	s_2	s_3	s_4	s_5	s_6
NoSpeech	0.7500	0.4950	0.0000	0.0000	0.0000	0.0000
MeSpeaking	0.1500	0.0500	0.0000	0.0000	0.0000	0.0000
MeTalkingToOthers	0.0250	0.0500	0.0000	0.9900	0.8000	0.3000
OthersSpeaking	0.0250	0.1000	0.9900	0.0100	0.2000	0.7000
LoudCrowd	0.0000	0.0050	0.0000	0.0000	0.0000	0.0000
DistantVoices	0.0500	0.1000	0.0000	0.0000	0.0000	0.0000
unspecific	0.0000	0.2000	0.0100	0.0000	0.0000	0.0000

Table A.14: Audio Observation Matrix for the Manually Designed 'Presentation' Model

$P(o_P\|s)$						
	s_1	s_2	s_3	s_4	s_5	s_6
Lying	0.0000	0.0000	0.0000	0.0000	0.0000	0.0000
Sitting	0.5000	0.3000	1.0000	0.0000	0.0000	1.0000
Standing	0.2000	0.3000	0.0000	0.9500	0.9000	0.0000
Walking	0.3000	0.3000	0.0000	0.0500	0.1000	0.0000
Running	0.0000	0.0000	0.0000	0.0000	0.0000	0.0000
Driving	0.0000	0.0000	0.0000	0.0000	0.0000	0.0000
unspecific	0.0000	0.1000	0.0000	0.0000	0.0000	0.0000

Table A.15: Posture Observation Matrix for the Manually Designed 'Presentation' Model

A.4 Manually Designed Model for the 'Shopping Alone' Activity

This section presents the manually designed HMM for the 'Shopping Alone' activity in terms of the initial state distribution (Table A.16), the state transition matrix (Table A.17) and the observation matrices for the sensor outputs (Table A.18 to Table A.20).

$P_1(s)$			
s_1	s_2	s_3	s_4
1.0000	0.0000	0.0000	0.0000

Table A.16: Initial State Distribution for the Manually Designed 'Shopping Alone' Model

$P(s^{(t+1)}\|s^{(t)})$				
	s_1	s_2	s_3	s_4
s_1	0.9000	0.0000	0.0000	0.0000
s_2	0.1000	0.9400	0.1500	0.0000
s_3	0.0000	0.0600	0.8000	0.0000
s_4	0.0000	0.0000	0.0500	1.0000

Table A.17: State Transition Matrix for the Manually Designed 'Shopping Alone' Model

$P(o_L\|s)$				
	s_1	s_2	s_3	s_4
AtHome	0.0000	0.0000	0.0000	0.0000
AtOffice	0.0000	0.0000	0.0000	0.0000
Street	0.7000	0.1000	1.0000	0.4000
Shop	0.0000	0.9000	0.0000	0.0000
Restaurant	0.0000	0.0000	0.0000	0.0000
Cinema	0.0000	0.0000	0.0000	0.0000
Car	0.1000	0.0000	0.0000	0.3000
Subway	0.1000	0.0000	0.0000	0.3000
unspecific	0.1000	0.0000	0.0000	0.0000

Table A.18: Location Observation Matrix for the Manually Designed 'Shopping Alone' Model

$P(o_A\|s)$				
	s_1	s_2	s_3	s_4
NoSpeech	0.0250	0.0000	0.0000	0.0250
MeSpeaking	0.2500	0.0100	0.0100	0.2500
MeTalkingToOthers	0.3000	0.0400	0.0300	0.3000
OthersSpeaking	0.1000	0.1000	0.0100	0.1000
LoudCrowd	0.1000	0.0000	0.0500	0.1000
DistantVoices	0.1250	0.8000	0.8000	0.1250
unspecific	0.1000	0.0500	0.1000	0.1000

Table A.19: Audio Observation Matrix for the Manually Designed 'Shopping Alone' Model

$P(o_P \mid s)$				
	s_1	s_2	s_3	s_4
Lying	0.0000	0.0000	0.0000	0.0000
Sitting	0.0500	0.0000	0.0000	0.0500
Standing	0.1000	0.3000	0.0500	0.1000
Walking	0.6500	0.7000	0.9000	0.6500
Running	0.0500	0.0000	0.0000	0.0500
Driving	0.2500	0.0000	0.0000	0.2500
unspecific	0.0000	0.0000	0.0500	0.0000

Table A.20: Posture Observation Matrix for the Manually Designed 'Shopping Alone' Model

A.5 Manually Designed Model for the 'Shopping with Friends' Activity

This section presents the manually designed HMM for the 'Shopping with Friends' activity in terms of the initial state distribution (Table A.21), the state transition matrix (Table A.22) and the observation matrices for the sensor outputs (Table A.23 to Table A.25).

$P_1(s)$					
s_1	s_2	s_3	s_4	s_5	s_6
1.0000	0.0000	0.0000	0.0000	0.0000	0.0000

Table A.21: Initial State Distribution for the Manually Designed 'Shopping with Friends' Model

$P(s^{(t+1)} \mid s^{(t)})$						
	s_1	s_2	s_3	s_4	s_5	s_6
s_1	0.9000	0.0000	0.0000	0.0000	0.0000	0.0000
s_2	0.0900	0.9300	0.1000	0.0000	0.0000	0.0000
s_3	0.0000	0.0600	0.8000	0.0000	0.0500	0.0000
s_4	0.0100	0.0100	0.0500	0.3000	0.0000	0.0000
s_5	0.0000	0.0000	0.0000	0.6500	0.9500	0.0000
s_6	0.0000	0.0000	0.0500	0.0500	0.0000	1.0000

Table A.22: State Transition Matrix for the Manually Designed 'Shopping with Friends' Model

$P(o_L\|s)$						
	s_1	s_2	s_3	s_4	s_5	s_6
AtHome	0.0000	0.0000	0.0000	0.0000	0.0000	0.0000
AtOffice	0.0000	0.0000	0.0000	0.0000	0.0000	0.0000
Street	0.7000	0.1000	1.0000	1.0000	0.0000	0.4000
Shop	0.0000	0.9000	0.0000	0.0000	0.0000	0.0000
Restaurant	0.0000	0.0000	0.0000	0.0000	1.0000	0.0000
Cinema	0.0000	0.0000	0.0000	0.0000	0.0000	0.0000
Car	0.1000	0.0000	0.0000	0.0000	0.0000	0.3000
Subway	0.1000	0.0000	0.0000	0.0000	0.0000	0.3000
unspecific	0.1000	0.0000	0.0000	0.0000	0.0000	0.0000

Table A.23: Location Observation Matrix for the Manually Designed 'Shopping with Friends' Model

$P(o_A\|s)$						
	s_1	s_2	s_3	s_4	s_5	s_6
NoSpeech	0.0250	0.0000	0.0000	0.0000	0.0000	0.0250
MeSpeaking	0.2500	0.0100	0.0100	0.0000	0.0000	0.2500
MeTalkingToOthers	0.3000	0.0400	0.0300	0.4900	0.4500	0.3000
OthersSpeaking	0.1000	0.1000	0.0100	0.4900	0.4500	0.1000
LoudCrowd	0.1000	0.0000	0.0500	0.0000	0.0000	0.1000
DistantVoices	0.1250	0.8000	0.8000	0.0200	0.0900	0.1250
unspecific	0.1000	0.0500	0.1000	0.0000	0.0100	0.1000

Table A.24: Audio Observation Matrix for the Manually Designed 'Shopping with Friends' Model

$P(o_P\|s)$						
	s_1	s_2	s_3	s_4	s_5	s_6
Lying	0.0000	0.0000	0.0000	0.0000	0.0000	0.0000
Sitting	0.0500	0.0000	0.0000	0.0000	1.0000	0.0500
Standing	0.1000	0.3000	0.0500	0.9500	0.0000	0.1000
Walking	0.6500	0.7000	0.9000	0.0500	0.0000	0.6500
Running	0.0500	0.0000	0.0000	0.0000	0.0000	0.0500
Driving	0.2500	0.0000	0.0000	0.0000	0.0000	0.2500
unspecific	0.0000	0.0000	0.0500	0.0000	0.0000	0.0000

Table A.25: Posture Observation Matrix for the Manually Designed 'Shopping with Friends' Model

A.6 Confusion Matrices

In this section, the confusion matrices are shown which have been employed in order to obtain more realistic sensor observations. For the location sensor, we distinguish between two confusion matrices corresponding to a normal sensor performance (Table A.26) and a decreased sensor performance (Table A.27). The confusion matrices for the audio and posture observations are shown for normal sensor performance in Table A.28 and Table A.29 respectively.

	AtHome	AtOffice	Street	Shop	Restaurant	Cinema	Car	Subway	unspecific
AtHome	737	1	19	1	1	1	1	1	1
AtOffice	1	1076	20	18	2	1	1	1	1
Street	9	9	218	17	16	16	2	2	1
Shop	1	1	115	343	47	3	1	3	1
Restaurant	1	2	98	48	482	14	1	1	1
Cinema	1	1	88	52	33	512	1	1	1
Car	1	3	298	20	2	4	412	15	1
Subway	2	2	302	22	2	5	12	509	1
unspecific	12	14	8	7	3	2	6	5	80

Table A.26: Confusion Matrix for Location Sensor (normal performance)

	AtHome	AtOffice	Street	Shop	Restaurant	Cinema	Car	Subway	unspecific
AtHome	491	1	19	1	1	1	1	1	1
AtOffice	1	717	20	18	2	1	1	1	1
Street	9	9	145	17	16	16	2	2	1
Shop	1	1	115	229	47	3	1	3	1
Restaurant	1	2	98	48	321	14	1	1	1
Cinema	1	1	88	52	33	341	1	1	1
Car	1	3	298	20	2	4	275	15	1
Subway	2	2	302	22	2	5	12	339	1
unspecific	12	14	8	7	3	2	6	5	53

Table A.27: Confusion Matrix for Location Sensor (decreased performance)

	NoSpeech	MeSpeaking	MeTalkingToOthers	OthersSpeaking	LoudCrowd	DistantVoices	unspecific
NoSpeech	785	4	4	21	3	8	1
MeSpeaking	7	104	50	15	9	1	2
MeTalkingToOthers	7	56	108	10	9	1	2
OthersSpeaking	26	6	6	493	21	10	1
LoudCrowd	1	1	2	16	46	1	2
DistantVoices	76	2	2	41	2	6	1
unspecific	11	12	12	12	10	10	80

Table A.28: Confusion Matrix for Audio Sensor

	Lying	Sitting	Standing	Walking	Running	Driving	unspecific
Lying	89	2	1	1	1	1	1
Sitting	3	6241	174	2	1	27	22
Standing	1	304	924	43	1	100	8
Walking	1	6	16	182	1	6	1
Running	1	1	1	1	22	1	1
Driving	1	6	17	2	1	547	1
unspecific	1	5	2	1	1	1	53

Table A.29: Confusion Matrix for Posture Sensor

A.7 Estimated Model for the 'Cinema' Activity

This section presents the estimated HMM for the 'Cinema' activity in terms of the initial state distribution (Table A.30), the state transition matrix (Table A.31) and the observation matrices for the sensor outputs (Table A.32 to Table A.34).

$P_1(s)$			
s_1	s_2	s_3	s_4
0.8953	0.0000	0.0000	0.1047

Table A.30: Initial State Distribution for the Estimated 'Cinema' Model

$P(s^{(t+1)}\|s^{(t)})$				
	s_1	s_2	s_3	s_4
s_1	0.8877	0.0104	0.0505	0.5455
s_2	0.0295	0.9529	0.7752	0.0327
s_3	0.0056	0.0361	0.1651	0.0109
s_4	0.0772	0.0005	0.0092	0.4109

Table A.31: State Transition Matrix for the Estimated 'Cinema' Model

$P(o_L\|s)$				
	s_1	s_2	s_3	s_4
AtHome	0.0089	0.0000	0.0000	0.0048
AtOffice	0.0107	0.0025	0.0000	0.0120
Street	0.8326	0.1198	0.0000	0.1781
Shop	0.0160	0.0555	0.0000	0.0911
Restaurant	0.0178	0.0185	0.0000	0.0683
Cinema	0.0157	0.8018	1.0000	0.0720
Car	0.0812	0.0000	0.0000	0.2453
Subway	0.0050	0.0019	0.0000	0.3017
unspecific	0.0121	0.0000	0.0000	0.0267

Table A.32: Location Observation Matrix for the Estimated 'Cinema' Model

$P(o_A\|s)$				
	s_1	s_2	s_3	s_4
NoSpeech	0.0000	0.0002	0.0000	0.0002
MeSpeaking	0.0235	0.9272	0.0057	0.0636
MeTalkingToOthers	0.0670	0.0000	0.9717	0.1322
OthersSpeaking	0.8005	0.0305	0.0170	0.3356
LoudCrowd	0.0249	0.0002	0.0000	0.0711
DistantVoices	0.0834	0.0049	0.0000	0.3158
unspecific	0.0007	0.0371	0.0057	0.0814

Table A.33: Audio Observation Matrix for the Estimated 'Cinema' Model

$P(o_P|s)$				
	s_1	s_2	s_3	s_4
Lying	0.1343	0.2366	0.3003	0.0783
Sitting	0.2472	0.2591	0.2040	0.1819
Standing	0.2711	0.2360	0.1926	0.2377
Walking	0.1856	0.1804	0.1700	0.2270
Running	0.0926	0.0361	0.0510	0.1319
Driving	0.0135	0.0142	0.0198	0.0289
unspecific	0.0556	0.0376	0.0623	0.1143

Table A.34: Posture Observation Matrix for the Estimated 'Cinema' Model

A.8 Estimated Model for the 'Dinner' Activity

This section presents the estimated HMM for the 'Dinner' activity in terms of the initial state distribution (Table A.35), the state transition matrix (Table A.36) and the observation matrices for the sensor outputs (Table A.37 to Table A.39).

$P_1(s)$			
s_1	s_2	s_3	s_4
0.8243	0.0000	0.0541	0.1216

Table A.35: Initial State Distribution for the Estimated 'Dinner' Model

$P(s^{(t+1)}|s^{(t)})$				
	s_1	s_2	s_3	s_4
s_1	0.9105	0.0021	0.0343	0.5369
s_2	0.0219	0.9669	0.3929	0.0369
s_3	0.0092	0.0307	0.5604	0.0123
s_4	0.0584	0.0004	0.0124	0.4139

Table A.36: State Transition Matrix for the Estimated 'Dinner' Model

$P(o_L \mid s)$				
	s_1	s_2	s_3	s_4
AtHome	0.0108	0.0000	0.0000	0.0047
AtOffice	0.0108	0.0010	0.0010	0.0076
Street	0.8373	0.1134	0.0298	0.1691
Shop	0.0240	0.0640	0.0123	0.0923
Restaurant	0.0172	0.7731	0.9435	0.0820
Cinema	0.0165	0.0470	0.0123	0.0703
Car	0.0710	0.0000	0.0010	0.2450
Subway	0.0014	0.0015	0.0000	0.2958
unspecific	0.0111	0.0000	0.0000	0.0332

Table A.37: Location Observation Matrix for the Estimated 'Dinner' Model

$P(o_A \mid s)$				
	s_1	s_2	s_3	s_4
NoSpeech	0.0000	0.0002	0.0021	0.0005
MeSpeaking	0.0505	0.9864	0.0031	0.0990
MeTalkingToOthers	0.0688	0.0002	0.8943	0.1501
OthersSpeaking	0.7663	0.0030	0.0585	0.3024
LoudCrowd	0.0258	0.0000	0.0000	0.0814
DistantVoices	0.0878	0.0050	0.0257	0.3235
unspecific	0.0007	0.0052	0.0164	0.0432

Table A.38: Audio Observation Matrix for the Estimated 'Dinner' Model

$P(o_P \mid s)$				
	s_1	s_2	s_3	s_4
Lying	0.1846	0.2808	0.2772	0.0845
Sitting	0.2072	0.0507	0.0873	0.1220
Standing	0.2007	0.0568	0.1253	0.1902
Walking	0.2215	0.3572	0.3039	0.2813
Running	0.1004	0.0526	0.1078	0.1303
Driving	0.0208	0.0404	0.0267	0.0438
unspecific	0.0649	0.1615	0.0719	0.1479

Table A.39: Posture Observation Matrix for the Estimated 'Dinner' Model

A.9 Estimated Model for the 'Presentation' Activity

This section presents the estimated HMM for the 'Presentation' activity in terms of the initial state distribution (Table A.40), the state transition matrix (Table A.41) and the observation matrices for the sensor outputs (Table A.42 to Table A.44).

$P_1(s)$		
s_1	s_2	s_3
0.5104	0.2500	0.2396

Table A.40: Initial State Distribution for the Estimated 'Presentation' Model

$P(s^{(t+1)}\|s^{(t)})$			
	s_1	s_2	s_3
s_1	0.7725	0.2958	0.4562
s_2	0.1594	0.6152	0.3076
s_3	0.0681	0.0890	0.2363

Table A.41: State Transition Matrix for the Estimated 'Presentation' Model

$P(o_L\|s)$			
	s_1	s_2	s_3
AtHome	0.0000	0.0000	0.0000
AtOffice	0.9814	0.9883	0.8328
Street	0.0091	0.0056	0.0721
Shop	0.0087	0.0040	0.0799
Restaurant	0.0009	0.0020	0.0152
Cinema	0.0000	0.0000	0.0000
Car	0.0000	0.0000	0.0000
Subway	0.0000	0.0000	0.0000
unspecific	0.0000	0.0000	0.0000

Table A.42: Location Observation Matrix for the Estimated 'Presentation' Model

$P(o_A\|s)$			
	s_1	s_2	s_3
NoSpeech	0.0004	0.0000	0.0000
MeSpeaking	0.9831	0.0044	0.0451
MeTalkingToOthers	0.0018	0.9553	0.0486
OthersSpeaking	0.0033	0.0032	0.6263
LoudCrowd	0.0000	0.0008	0.0014
DistantVoices	0.0093	0.0322	0.2454
unspecific	0.0020	0.0040	0.0332

Table A.43: Audio Observation Matrix for the Estimated 'Presentation' Model

$P(o_P|s)$			
	s_1	s_2	s_3
Lying	0.2103	0.1340	0.4235
Sitting	0.1220	0.2559	0.1579
Standing	0.1995	0.4628	0.2516
Walking	0.4151	0.0938	0.1122
Running	0.0382	0.0414	0.0398
Driving	0.0102	0.0004	0.0073
unspecific	0.0047	0.0117	0.0077

Table A.44: Posture Observation Matrix for the Estimated 'Presentation' Model

A.10 Estimated Model for the 'Shopping Alone' Activity

This section presents the estimated HMM for the 'Shopping Alone' activity in terms of the initial state distribution (Table A.45), the state transition matrix (Table A.46) and the observation matrices for the sensor outputs (Table A.47 to Table A.49).

$P_1(s)$		
s_1	s_2	s_3
0.7200	0.2267	0.0533

Table A.45: Initial State Distribution for the Estimated 'Shopping Alone' Model

$P(s^{(t+1)}|s^{(t)})$			
	s_1	s_2	s_3
s_1	0.8374	0.2316	0.4902
s_2	0.1090	0.7536	0.0588
s_3	0.0537	0.0149	0.4510

Table A.46: State Transition Matrix for the Estimated 'Shopping Alone' Model

$P(o_L|s)$			
	s_1	s_2	s_3
AtHome	0.0073	0.0037	0.0051
AtOffice	0.0092	0.0008	0.0118
Street	0.8791	0.0058	0.1821
Shop	0.0024	0.9390	0.1309
Restaurant	0.0226	0.0432	0.1075
Cinema	0.0204	0.0058	0.0583
Car	0.0559	0.0004	0.2255
Subway	0.0014	0.0008	0.2617
unspecific	0.0016	0.0004	0.0172

Table A.47: Location Observation Matrix for the Estimated 'Shopping Alone' Model

$P(o_A|s)$			
	s_1	s_2	s_3
NoSpeech	0.0004	0.0000	0.0001
MeSpeaking	0.0392	0.0419	0.0826
MeTalkingToOthers	0.0632	0.0979	0.2426
OthersSpeaking	0.8208	0.8465	0.2626
LoudCrowd	0.0143	0.0004	0.0659
DistantVoices	0.0571	0.0120	0.3278
unspecific	0.0051	0.0012	0.0185

Table A.48: Audio Observation Matrix for the Estimated 'Shopping Alone' Model

$P(o_P|s)$			
	s_1	s_2	s_3
Lying	0.2618	0.4548	0.0955
Sitting	0.1562	0.0635	0.1178
Standing	0.1686	0.0544	0.1819
Walking	0.2482	0.3270	0.3218
Running	0.0822	0.0253	0.1204
Driving	0.0271	0.0465	0.0495
unspecific	0.0559	0.0286	0.1132

Table A.49: Posture Observation Matrix for the Estimated 'Shopping Alone' Model

A.11 Estimated Model for the 'Shopping with Friends' Activity

This section presents the estimated HMM for the 'Shopping with Friends' activity in terms of the initial state distribution (Table A.50), the state transition matrix (Table A.51) and the observation matrices for the sensor outputs (Table A.52 to Table A.54).

$P_1(s)$			
s_1	s_2	s_3	s_4
0.8000	0.1500	0.0125	0.0375

Table A.50: Initial State Distribution for the Estimated 'Shopping with Friends' Model

$P(s^{(t+1)}|s^{(t)})$				
	s_1	s_2	s_3	s_4
s_1	0.8604	0.2202	0.0122	0.5512
s_2	0.0720	0.7528	0.0069	0.0540
s_3	0.0147	0.0114	0.9789	0.0019
s_4	0.0530	0.0156	0.0020	0.3929

Table A.51: State Transition Matrix for the Estimated 'Shopping with Friends' Model

$P(o_L|s)$				
	s_1	s_2	s_3	s_4
AtHome	0.0084	0.0013	0.0000	0.0062
AtOffice	0.0095	0.0026	0.0018	0.0127
Street	0.8670	0.0030	0.1341	0.1760
Shop	0.0012	0.9431	0.0620	0.1003
Restaurant	0.0197	0.0395	0.7865	0.0691
Cinema	0.0215	0.0065	0.0151	0.0541
Car	0.0698	0.0026	0.0000	0.2516
Subway	0.0013	0.0000	0.0005	0.3178
unspecific	0.0016	0.0013	0.0000	0.0122

Table A.52: Location Observation Matrix for the Estimated 'Shopping with Friends' Model

$P(o_A|s)$				
	s_1	s_2	s_3	s_4
NoSpeech	0.0003	0.0000	0.0000	0.0000
MeSpeaking	0.0406	0.0435	0.9781	0.0753
MeTalkingToOthers	0.0637	0.1078	0.0173	0.2167
OthersSpeaking	0.8002	0.8314	0.0000	0.2952
LoudCrowd	0.0199	0.0009	0.0000	0.0643
DistantVoices	0.0707	0.0161	0.0027	0.3361
unspecific	0.0045	0.0004	0.0018	0.0124

Table A.53: Audio Observation Matrix for the Estimated 'Shopping with Friends' Model

$P(o_P|s)$				
	s_1	s_2	s_3	s_4
Lying	0.2225	0.4576	0.0967	0.0874
Sitting	0.1857	0.0626	0.1341	0.1413
Standing	0.1927	0.0548	0.2605	0.2004
Walking	0.2301	0.3294	0.4443	0.3011
Running	0.0859	0.0309	0.0397	0.1314
Driving	0.0253	0.0343	0.0128	0.0369
unspecific	0.0578	0.0304	0.0119	0.1015

Table A.54: Posture Observation Matrix for the Estimated 'Shopping with Friends' Model

B Publications as (co-)author

[AHE06] M. Alia, G. Horn, F. Eliassen, M. U. Khan, R. Fricke, and R. Reichle.
A Component-Based Planning Framework for Adaptive Systems.
In *The 8th International Symposium on Distributed Objects and Applications (DOA '06)*, pages 1686–1704, Montpellier, France, 2006.

[BR07] Ph. A. Baer and R. Reichle.
Robotic Soccer, book chapter *Communication and Collaboration in Heterogeneous Teams of Soccer Robots*, pages 1–28.
I-Tech Education and Publishing, Vienna, Austria, December 2007,
ISBN 978-3-902613-21-9.

[BRB07] Ph. A. Baer, R. Reichle, K. Baumgart, T. Kleppe, C. Hoppe, S. Triller, D. Saur, J. Wollenhaupt, T. Amma, M. Blumenstein, F. Seute, A. Witsch, K. Geihs.
Carpe Noctem 2006 - Team description paper.
In *G. Lakemeyer, E. Sklar, D. Sorrenti, T. Takahashi (editors), RoboCup 2006: Robot Soccer World Cup X,* LNCS, Springer, 2006. (CD Supplement)

[BRG08] Ph. A. Baer, R. Reichle, and K. Geihs.
The Spica Development Framework – Model-Driven Software Development for Autonomous Mobile Robots.
In W. Burgard, R. Dillmann, C. Plagemann, and N. Vahrenkamp (editors), *Intelligent Autonomous Systems 10 – IAS-10*, pages 211–220, Baden Baden, Germany, 2008.

[BRZ07] Ph. A. Baer, R. Reichle, M. Zapf, T. Weise, and K. Geihs.
A Generative Approach to the Development of Autonomous Robot Software.
In *Fourth IEEE International Workshop on Engineering of Autonomic and Autonomous Systems (EASe 2007)*, pages 43–52, Tucson, USA, 2007.

[DRW09] A. Devlic, R. Reichle, M. Wagner, M. Kirsch Pinheiro, Y. Vanrompay, Y. Berbers, and M. Valla.
Context inference of users' social relationships and distributed policy management.
In *Proceedings of the 6th IEEE Workshop on Context Modeling and Reasoning (CoMoRea) at the 7th IEEE International Conference on Pervasive Computing and Communication (PerCom '09)*, pages 1–8, Galveston, USA, 2009.

[GBE09] K. Geihs, P. Barone, F. Eliassen, J. Floch, R. Fricke, E. Gjorven, S. Hallsteinsen, G. Horn, M. U. Khan, A. Mamelli, G. A. Papadopoulos, N. Paspallis, R. Reichle, and E. Stav.
A comprehensive solution for application-level adaptation.
Software Practice & Experiences, 39(4):385–422, 2009. John Wiley & Sons, Ltd,
ISSN 0038-0644.

[GBR08] K. Geihs, Ph. Baer, R. Reichle, and J. Wollenhaupt.
Ontology-based automatic model transformations.
In *SEFM '08: Proceedings of the 2008 Sixth IEEE International Conference on Software Engineering and Formal Methods*, pages 387–391, Cape Town, South Africa, 2008.

[GKR06] K. Geihs, M. U. Khan, R. Reichle, A. Solberg, S. Hallsteinsen, and S. Merral.
Modeling of component-based adaptive distributed applications.
In *SAC '06: Proceedings of the 2006 ACM symposium on Applied computing - Track on Dependable and adaptive distributed systems (DADS)*, pages 718–722, Dijon, France, 2006.

[GKR06a] K. Geihs, M. U. Khan, R. Reichle, and A. Solberg.
Modeling of Component-Based Self-Adapting Context-Aware Applications for Mobile Devices.
In *IFIP Working Conference on Software Engineering Techniques*, pages 85–96, Warsaw, Poland, 2006.

[GRK06] K. Geihs, R. Reichle, M. U. Khan, A. Solberg, and S. Hallsteinsen.
Model-driven development of self-adaptive applications for mobile devices (research summary).
In *SEAMS '06: Proceedings of the 2006 international workshop on Self-adaptation and self-managing systems*, page 95, Shanghai, China, 2006.

[GRW09a] K. Geihs, R. Reichle, M. Wagner, and M. U. Khan.
Modeling of context-aware self-adaptive applications in ubiquitous and service-oriented environments.
In *Software Engineering for Self-Adaptive Systems*, pages 146–163, Berlin, Heidelberg, 2009. Springer-Verlag,
ISBN: 978-3-642-02160-2.

[GRW09b] K. Geihs, R. Reichle, M. Wagner, and M. U. Khan.
Service-Oriented Adaptation in Ubiquitous Computing Environments.
In *Proceedings of the 12th IEEE International Conference on Computational Science and Engineering (CSE '09)*, pages 458-463, Vancouver, Canada, 2009.

[HKB09] J. Happe, H. Koziolek, U. Bellur, H. Giese, W. Hasselbring, R. Laddaga, T. Margaria, J. Martinez, C. Müller-Schloer, and R. Reichle.
The Role of Models in Self-adaptive and Self-healing Systems.
In *Artur Andrzejak, Kurt Geihs, Onn Shehory, John Wilkes: Dagstuhl Seminar Self-Healing and Self-Adaptive Systems 09201*, Dagstuhl, Germany, 2009.

[KMU03] G. K. Kraetzschmar, G. Mayer, H. Utz, Ph. A. Baer, M. Clauss, U. Kaufmann, M. Lauer, S. Natterer, S. Przewoznik, R. Reichle, A. Reisser, A. Roth, M. Schmidt, C. Sitter, F. Sterk, and G. Palm.
The Ulm Sparrows 2003 - Team description paper.
In *RoboCup 2003: Robot Soccer World Cup VII*, Lecture Notes in Artificial Intelligence, volume 3020, 2003.

[KMU04] G. K. Kraetzschmar, G. Mayer, H. Utz, Ph. A. Baer, M. Clauss, U. Kaufmann, M. Lauer, S. Natterer, S. Przewoznik, R. Reichle, C. Sitter, F. Sterk, and G. Palm.
The Ulm Sparrows 2004 - Team description paper.
In *RoboCup 2004: Robot Soccer World Cup VIII*, Lecture Notes in Artificial Intelligence, volume 3276, 2004.

[KRG07a] M. U. Khan, R. Reichle, and K. Geihs.
Architectural constraints in the model-driven development of self-adaptive applications.
IEEE Distributed Systems Online, 9(7),
ISSN:1541-4922.

[KRG07b] M. U. Khan, R. Reichle, and K. Geihs.
Applying architectural constraints in the modeling of self-adaptive component-based applications.
In *Model-Driven Software Adaptation (M-ADAPT '07) at The European Conference on Object-Oriented Programming (ECOOP '07)*, volume 1, pages 13–22, Berlin, Germany, 2007.

[KRH07] U. Kaufmann, R. Reichle, C. Hoppe, and Ph. A. Baer.
An unsupervised approach for adaptive color segmentation.
In *Proceedings of the 1st International Workshop, in conjunction with VISAPP 2007*, pages 3–12, Barcelona, Spain, 2007.

[KRW09] M. U. Khan, R. Reichle, M. Wagner, K. Geihs, U. Scholz, C. Kakousis, and G. A. Papadopoulos.
An adaptation reasoning approach for large scale component-based applications.
In *Communications of the EASST*, volume 19, Proceedings of the Second International DisCoTec Workshop on Context-Aware Adaptation Mechanisms for Pervasive and Ubiquitous Services(CAMPUS 2009), pages 75–86, Lisbon, Portugal, 2009.

[Re09] R. Reichle.
Context modelling and reasoning for adaptive applications in ubiquitous computing environments.
In *Artur Andrzejak, Kurt Geihs, Onn Shehory, John Wilkes: Dagstuhl Seminar Self-Healing and Self-Adaptive Systems 09201 (Abstracts Collection)*, Dagstuhl, Germany, 2009.

[RBN04] R. Reichle, P. Bayerl, and H. Neumann.
A self-organized neural network for pattern recognition.
In *Proceedings of the 5th Workshop Dynamische Perzeption*, pages 183–188, Tübingen, Germany, 2004.

[RKG08] R. Reichle, M. U. Khan, and K. Geihs.
How to combine parameter and compositional adaptation in the modeling of self-adaptive applications.
Praxis der Informationsverarbeitung und Kommunikation (PIK), 31 (2008)(1):34–38, March 2008, K.G. Saur, Munich.

[RWK08] R. Reichle, M. Wagner, M. U. Khan, K. Geihs, J. Lorenzo, M. Valla, C. Fra, N. Paspallis, and G. A. Papadopoulos.
A comprehensive context modeling framework for pervasive computing systems.
In *Proceedings of the 8th IFIP WG 6.1 International Conference on Distributed Applications and Interoperable Systems (DAIS)*, volume 5022 of *LNCS*, pages 281-295, Oslo, Norway, June 2008. Springer Verlag.

[RWK08a] R. Reichle, M. Wagner, M. U. Khan, K. Geihs, M. Valla, C. Fra, N. Paspallis, and G. A. Papadopoulos.
A context query language for pervasive computing environments.
In *Proceedings of the 5th IEEE Workshop on Context Modeling and Reasoning (CoMoRea) at the 6th IEEE International Conference on Pervasive Computing and Communication (PerCom '08)*, pages 434–440, Hong Kong, 2008.

[SWR09] H. Skubch, M. Wagner, and R. Reichle.
A language for interactive cooperative agents.
Technical report, University of Kassel, 2009.

[SWR10] H. Skubch, M. Wagner, R. Reichle, S. Triller, and K. Geihs.
Towards a Comprehensive Teamwork Model for Highly Dynamic Domains.
In *Proceedings of the 2nd International Conference on Agents and Artificial Intelligence*, volume 2, pages 121–127, Valencia, Spain, 2010.

[WNS08] T. Weise, S. Niemczyk, H. Skubch, R. Reichle, and K. Geihs.
A tunable model for multi-objective, epistatic, rugged, and neutral fitness landscapes.
In *Proceedings of Genetic and Evolutionary Computation Conference, GECCO 2008*, pages 795–802. Atlanta, USA, 2008.

Bibliography

[1] ANTLR Parser Generator.
`http://antlr.org/` (accessed 2010-05-12).

[2] B. Babcock, S. Babu, M. Datar, R. Motwani, and J. Widom.
Models and issues in data stream systems.
In *PODS '02: Proceedings of the 21st ACM SIGMOD-SIGACT-SIGART symposium on Principles of database systems*, pages 1–16, Madison, Wisconsin, USA, 2002.

[3] Ph. A. Baer.
Platform-Independent Development of Robot Communication Software.
PhD thesis, Distributed Systems Group, University of Kassel, Germany, 2008.

[4] Ph. A. Baer and R. Reichle.
Communication and collaboration in heterogeneous teams of soccer robots.
In *Robotic Soccer*, pages 1–28, Vienna, Austria, 2007. I-Tech Education and Publishing.
ISBN 978-3-902613-21-9.

[5] Ph. A. Baer, R. Reichle, M. Zapf, T. Weise, and K. Geihs.
A Generative Approach to the Development of Autonomous Robot Software.
In *Fourth IEEE International Workshop on Engineering of Autonomic and Autonomous Systems (EASe 2007)*, pages 43–52, Tucson, USA, 2007.

[6] Ph. A. Baer, R. Reichle, and K. Geihs.
The Spica Development Framework – Model-Driven Software Development for Autonomous Mobile Robots.
In *Intelligent Autonomous Systems 10 – IAS-10*, pages 211–220, Baden Baden, Germany, 2008.

[7] Ph. A. Baer, T. Weise, and K. Geihs.
Geminga: Service Discovery for Mobile Robotics.
In *The Third International Conference on Systems and Networks Communications*, pages 167–172, Sliema, Malta, 2008.

[8] Y. Bar-Shalom and T. E. Fortmann.
Tracking and data association.
Mathematics in Science and Engineering. Academic Press Professional, Inc., San Diego, CA, USA, 1987.
ISBN 0120797607.

[9] H. S. M. Beigi, S. H. Maes, and J. S. Sorensen.
A distance measure between collections of distributions and its application to speaker recognition.
In *Proceedings of International Conference on Acoustics, Speech and Signal Processing*, volume 2, pages 753–756, Seattle, Washington, USA, 1998.

[10] E. W. Beth.
Semantic entailment and formal derivability.
Koninklijke Nederlandse Akademie van Wentenschappen, Proceedings of the Section of Sciences, 18:309–342, 1955.

[11] M. L. Blum.
Real-time Context Recognition.
Master's thesis, Swiss Federal Institute of Technology Zurich (ETH), 2005.

[12] M. H. Bowling, B. Browning, and M. M. Veloso.
Plays as Effective Multiagent Plans Enabling Opponent-Adaptive Play Selection.
In *Proceedings of the Fourteenth International Conference on Automated Planning and Scheduling (ICAPS 2004)*, pages 376–383, Whistler, British Columbia, Canada, 2004.

[13] I. Bratko.
Prolog (3rd ed.): Programming for Artificial Intelligence.
Addison-Wesley Longman Publishing Co., Inc., Boston, MA, USA, 2001.
ISBN 0-201-40375-7.

[14] D. Brickley and R. V. Guha.
RDF Vocabulary Description Language 1.0: RDF Schema, W3C Recommendation 10 February 2004.
`http://www.w3.org/TR/rdf-schema/` (accessed 2010-06-30).

[15] R. R. Brooks and S. S. Iyengar.
Multi-sensor fusion: fundamentals and applications with software.
Prentice-Hall, Inc., Upper Saddle River, NJ, USA, 1998.
ISBN 0-13-901653-8.

[16] M. Broxvall, B.S. Seo, and W.Y. Kwon.
The PEIS kernel: A middleware for ubiquitous robotics.
In *Proceedings of the IROS-07 Workshop on Ubiquitous Robotic Space Design and Applications*, San Diego, CA, USA, 2007.

[17] H. Bruyninckx.
Open Robot Control Software: the OROCOS project.
In *Proceedings of the IEEE International Conference on Robotics and Automation (ICRA 2001)*, volume 3, pages 2523–2528, Seoul, Korea, 2001.

[18] Carpe Noctem RoboCup Team.
`http://carpenoctem.das-lab.net/` (accessed 2010-07-02).

[19] G. Chen.
Solar: Building A Context Fusion Network for Pervasive Computing.
PhD thesis, Deptartment of Computer Science, Dartmouth College, Hanover, NH, USA, 2004.

[20] G. Chen and D. Kotz.
Context aggregation and dissemination in ubiquitous computing systems.
In *Proceedings of the Fourth IEEE Workshop on Mobile Computing Systems and Applications (WMCSA)*, pages 105–114, Callicoon, New York, USA, 2002.

[21] G. Chen, M. Li, and D. Kotz.

Design and implementation of a large-scale context fusion network.
In *First Annual International Conference on Mobile and Ubiquitous Systems: Networking and Services (Mobiquitous)*, pages 246–255, Boston, Massachusetts, USA, 2004.

[22] H. Chen and T. Finin.
An ontology for a context aware pervasive computing environment.
In *IJCAI workshop on ontologies and distributed systems*, Acapulco, Mexico, 2003.

[23] H. Chen, T. Finin, and A. Joshi.
The SOUPA Ontology for Pervasive Computing.
In *Ontologies for Agents: Theory and Experiences*, pages 233–258. Birkhäuser Verlag, Basel, 2005.
ISBN 978-3-7643-7237-8.

[24] B. R. Cobb and P. P. Shenoy.
On the plausibility transformation method for translating belief function models to probability models.
International Journal of Approximate Reasoning, 41(3):314–330, 2006.

[25] S. Coradeschi and A. Saffiotti.
An Introduction to the Anchoring Problem.
Robotics and Autonomous Systems, 43(2-3):85–96, 2003.

[26] Ó. Corcho and A. Gómez-Pérez.
A roadmap to ontology specification languages.
In *EKAW '00: Proceedings of the 12th European Workshop on Knowledge Acquisition, Modeling and Management*, pages 80–96, London, UK, 2000.

[27] A. P. Dempster.
A generalization of Bayesian inference.
Journal of the Royal Statistical Society, 30(B):205–247, 1968.

[28] A. P. Dempster, N. M. Laird, and D. B. Rubin.
Maximum likelihood from incomplete data via the EM algorithm.
Journal of the Royal Statistical Society, Series B, 39(1):1–38, 1977.

[29] A. K. Dey.
Providing architectural support for building context-aware applications.
PhD thesis, Georgia Institute of Technology, Atlanta, GA, USA, 2000.

[30] A. K. Dey.
Understanding and using context.
Personal Ubiquitous Computing, 5(1):4–7, 2001.
ISSN 1617-4909.

[31] M. B. Dias.
TraderBots: A New Paradigm for Robust and Efficient Multirobot Coordination in Dynamic Environments.
PhD thesis, Robotics Institute, Carnegie Mellon University, Pittsburgh, PA, USA, 2004.

[32] D. Dubois and H. Prade.
On the unicity of Dempster's rule of combination.
International Journal of Intelligent Systems, 1:133–142, 1986.

[33] D. Dubois and H. Prade.
A Set-Theoretic View of Belief Functions.
In *Classic Works of the Dempster-Shafer Theory of Belief Functions*, pages 375–410. 2008.

[34] Eigen - C++ template library for linear algebra.
http://eigen.tuxfamily.org/index.php?title=Main_Page (accessed 2010-07-02).

[35] R. T. Fielding.
Architectural Styles and the Design of Network-based Software Architectures.
PhD thesis, University of California, Irvine, USA, 2000.

[36] Foundation for Intelligent Physical Agents.
http://www.fipa.org/ (accessed 2010-07-02).

[37] Foundation for Intelligent Physical Agents.
FIPA Abstract Architecture Specification. 2002.
http://www.fipa.org/specs/fipa00001 (accessed 2010-07-02).

[38] Foundation for Intelligent Physical Agents.
FIPA ACL Message Structure Specification. 2001.
http://www.fipa.org/specs/fipa00061 (accessed 2010-07-02).

[39] Foundation for Intelligent Physical Agents.
FIPA ACL Message Representation in Bit-Efficient Specification. 2002.
http://www.fipa.org/specs/fipa00069 (accessed 2010-07-02).

[40] Foundation for Intelligent Physical Agents.
FIPA ACL Message Representation in XML Specification. 2002.
http://www.fipa.org/specs/fipa00071 (accessed 2010-07-02).

[41] Foundation for Intelligent Physical Agents.
FIPA Agent Message Transport Protocol for HTTP Specification. 2002.
http://www.fipa.org/specs/fipa00084 (accessed 2010-07-02).

[42] Foundation for Intelligent Physical Agents.
FIPA Agent Message Transport Service Specification. 2002.
http://www.fipa.org/specs/fipa00067 (accessed 2010-07-02).

[43] Foundation for Intelligent Physical Agents.
FIPA SL Content Language Specification. 2002.
http://www.fipa.org/specs/fipa00008 (accessed 2010-07-02).

[44] E. Freeman, K. Arnold, and S. Hupfer.
JavaSpaces Principles, Patterns, and Practice.
Addison-Wesley Longman Ltd., Essex, UK, UK, 1999.
ISBN 0201309556.

[45] P. Gärdenfors.
Conceptual Spaces: the Geometry of Thought.
MIT Press, London, 2000.

[46] K. Geihs.
Middleware challenges ahead.

IEEE Computer, 34(6):24–31, 2001.
ISSN 0018-9162.

[47] K. Geihs and U. Hollberg.
Retrospective on DACNOS.
Communications of the ACM, 33(4):439–448, 1990.
ISSN 0001-0782.

[48] K. Geihs, R. Reichle, M. Wagner, and M. U. Khan.
Modeling of context-aware self-adaptive applications in ubiquitous and service-oriented environments.
In *Software Engineering for Self-Adaptive Systems*, pages 146–163, Berlin, Heidelberg, 2009. Springer-Verlag.
ISBN 978-3-642-02160-2.

[49] M. R. Genesereth and N. J. Nilsson.
Logical Foundations of Artificial Intelligence.
Morgan Kaufman, 1987.

[50] K. Gödel.
Zum intuitionistischen Aussagenkalkül.
Anzeiger Akademie der Wissenschaften Wien, 69:65–66, 1932.

[51] Y. Y. Goland, T. Cai, P. Leach, and Y. Gu.
Simple Service Discovery Protocol/1.0, Operating without an Arbiter.
IETF (Internet Draft), 1999.

[52] GPS Monitor.
`http://www.kowoma.de/gps/gpsmonitor/gpsmonitor.php` (accessed 2010-07-02).

[53] T. R. Gruber.
A translation approach to portable ontology specifications.
Knowledge Acquisition, 5(2):199–220, 1993.
ISSN 1042-8143.

[54] E. Guttman, C. E. Perkins, J. Veizades, and M. Day.
Service Location Protocol, Version 2.
RFC 2608 (Proposed Standard), 1999, `http://www.ietf.org/rfc/rfc2608.txt` (accessed 2010-07-02).

[55] D. L. Hall and J. Llinas.
An introduction to multisensor data fusion.
Proceedings of the IEEE, 85(1):6–23, 2002.

[56] K. Henricksen and J. Indulska.
A software engineering framework for context-aware pervasive computing.
In *PERCOM '04: Proceedings of the Second IEEE International Conference on Pervasive Computing and Communications (PerCom'04)*, pages 77–86, Orlando, Florida, USA, 2004.

[57] K. Henricksen and J. Indulska.
Developing context-aware pervasive computing applications: Models and approach.
Pervasive and Mobile Computing, 2(1):37–64, 2006.

[58] K. Henricksen, J. Indulska, T. McFadden, and S. Balasubramaniam.
Middleware for distributed context-aware systems.
In *Proceedings of the 7th International Conference On Distributed Objects And Applications (DOA '05)*, pages 846–863, Agia Napa, Cyprus, 2005.

[59] K. Henricksen, J. Indulska, and A. Rakotonirainy.
Using context and preferences to implement self-adapting pervasive computing applications.
Software, Practice and Experience, 36(11-12):1307–1330, 2006.

[60] M. R. Hestenes and E. Stiefel.
Methods of Conjugate Gradients for Solving Linear Systems.
Journal of Research of the National Bureau of Standards, 49:409–436, 1952.

[61] M. Horridge, N. Drummond, J. Goodwin, A. Rector, R. Stevens, and H. Wang.
The manchester owl syntax.
In *Second Workshop on OWL Experiences and Directions (OWLED2006)*, Athens, GA, USA, 2006.

[62] I. Horrocks.
DAML+OIL: A Description Logic for the Semantic Web.
IEEE Bulletin Technical Committee on Data Engineering, 25(1):4–9, 2002.

[63] IKVM.NET - Java for Mono and the Microsoft .NET Framework.
`http://www.ikvm.net/` (accessed 2010-07-02).

[64] International Organization for Standardization.
SQL/Foundation - ISO/IEC 9075-2:2008.
`http://www.iso.org/iso/iso_catalogue/catalogue_tc/catalogue_detail.htm?csnumber=38640` (accessed 2010-07-02).

[65] International Telecommunication Union.
Abstract Syntax Notation One (ASN.1): Specification of basic notation, 2002.
ITU-T Recommendation X.680.

[66] JADE - Java Agent Development Framework.
`http://jade.tilab.com/` (accessed 2010-07-02).

[67] JIAC - Java-based Intelligent Agent Componentware.
`http://jiac.de/` (accessed 2010-07-02).

[68] E. Jones, B. Browning, M. B. Dias, B. Argall, M. M. Veloso, and A. Stentz.
Dynamically formed heterogeneous robot teams performing tightly-coordinated tasks.
In *IEEE International Conference on Robotics and Automation (ICRA 2006)*, pages 570 – 575, Orlando, Florida, USA, 2006.

[69] JSON - JavaScript Object Notation. Language specification.
`http://www.json.org` (accessed 2010-07-02).

[70] S. J. Julier and J. K. Uhlmann.
A New Extension of the Kalman Filter to Nonlinear Systems.
In *The 11th International Symposium on Aerospace/Defence Sensing, Simulation and Controls*, pages 182–193, Orlando, FL, USA, 1997.

[71] R. E. Kalman.

A new approach to linear filtering and prediction problems.
Transactions of the ASME-Journal of Basic Engineering 82, (Series D):35–45, 1960.

[72] M. Kifer and G. Lausen.
F-logic: a higher-order language for reasoning about objects, inheritance, and scheme.
ACM SIGMOD Record, 18(2):134–146, 1989.
ISSN 0163-5808.

[73] M. Kifer, G. Lausen, and J. Wu.
Logical foundations of object-oriented and frame-based languages.
Journal of the ACM, 42(4):741–843, 1995.
ISSN 0004-5411.

[74] J. H. Kim and J. Pearl.
A computational model for combined causal and diagnostic reasoning in inference systems.
In *Proceedings of the IJCAI-83*, pages 190–193, Karlsruhe, Germany, 1983.

[75] L. A. Klein.
Sensor and Data Fusion: A Tool for Information Assessment and Decision Making (SPIE Press Monograph Vol. PM138).
SPIE - International Society for Optical Engineering, 2004.
ISBN 0819454354.

[76] K. LeBlanc and A. Saffiotti.
Cooperative anchoring in heterogeneous multi-robot systems.
In *Proceedings of the IEEE International Conference on Robotics and Automation (ICRA)*, pages 3308–3314, Pasadena, CA, USA, 2008.

[77] L. Liu, C. Shenoy, and P. P. Shenoy.
Knowledge representation and integration for portfolio evaluation using linear belief functions.
IEEE Transactions on Systems, Man, and Cybernetics, Part A, 36(4):774–785, 2006.

[78] J. D. Lowrance, T. D. Garvey, and T. M. Strat.
A framework for evidential-reasoning systems.
In *Readings in Uncertain Reasoning*, pages 611–618, San Francisco, CA, USA, 1990. Morgan Kaufmann Publishers Inc.
ISBN 1-55860-125-2.

[79] R. M. MacGregor.
Inside the LOOM description classifier.
ACM SIGART Bulletin, 2(3):88–92, 1991.
ISSN 0163-5719.

[80] P. C. Mahalanobis.
On the generalised distance in statistics.
In *Proceedings of the National Institute of Science, India*, volume 2, pages 49–55, 1936.

[81] F. Manola and E. Miller.
RDF Primer, W3C Recommendation 10 February 2004.
`http://www.w3.org/TR/rdf-primer/` (accessed 2010-07-02).

[82] S. McKeever, J. Ye, L. Coyle, and S. Dobson.

Using Dempster-Shafer Theory of Evidence for Situation Inference.
In *4th European Conference on Smart Sensing and Context (EuroSSC)*, pages 149–162, Guildford, UK, 2009.

[83] S. McKeever, J. Ye, L. Coyle, and S. Dobson.
A context quality model to support transparent reasoning with uncertain context.
In *1st International Workshop on Quality of Context (QuaCon)*, pages 65–75, Stuttgart, Germany, 2009.

[84] D. Mercier, B. Quost, and T. Denux.
Refined modeling of sensor reliability in the belief function framework using contextual discounting.
Information Fusion, 9(2):246–258, 2008.
ISSN 1566-2535.

[85] Mono - Cross platform, open source .NET development framework.
`http://www.mono-project.com/Main_Page` (accessed 2010-07-02).

[86] E. Motta.
An Overview of the OCML Modelling Language.
In *Proceedings KEML'98: 8th Workshop on Knowledge Engineering Methods and Languages*, pages 21–22, Karlsruhe, Germany, 1998.

[87] C. K. Murphy.
Combining belief functions when evidence conflicts.
Decision Support Systems, 29(1):1–9, 2000.
ISSN 0167-9236.

[88] K. Murphy.
Pearl's algorithm and multiplexer nodes.
Technical report, U.C. Berkeley, Department of Computer Science, 1999.

[89] R. R. Murphy.
Dempster-Shafer Theory for Sensor Fusion in Autonomous Mobile Robots.
IEEE Transactions on Robotics and Automation, 14(2):197–206, 1998.

[90] MUSIC - Self-Adapting Applications for Mobile Users in Ubiquitous Computing Environments.
`http://www.ist-music.eu/` (accessed 2010-07-02).

[91] D. Nardi, G. Adorni, A. Bonarini, A. Chella, G. Clemente, E. Pagello, and M. Piaggio.
Art-99: Azzurra robot team.
In *RoboCup-99: Robot Soccer World Cup III*, volume 1856 of *Lecture Notes in Computer Science*, pages 695–698. Springer, 2000.
ISBN 3-540-41043-0.

[92] D. Nau, Y. Cao, A. Lotem, and H. Munoz-Avila.
SHOP: simple hierarchical ordered planner.
In *IJCAI'99: Proceedings of the 16th international joint conference on Artificial intelligence*, pages 968–973, Stockholm, Sweden, 1999.

[93] A. Newberger and A. K. Dey.
Designer support for context monitoring and control.
Technical report, Intel Research Berkeley, 2003.

[94] G. Niemeyer.
Kybernetische System- und Modelltheorie.
Systemstudium Wirtschaftsinformatik. Vahlen, München, 1977.
ISBN 3800605813.

[95] NTP - The Network Time Protocol.
`http://www.ntp.org/` (accessed 2010-07-02).

[96] Object Management Group.
Common Object Request Broker Architecture (CORBA) Specification, Version 3.1. 2008.
`http://www.omg.org/spec/CORBA/3.1/` (accessed 2010-07-02).

[97] K. Ogata.
State Space Analysis of Control Systems.
Prentice-Hall, Englewood Cliffs, NJ, 1967.

[98] N. Oliver, E. Horvitz, and A. Garg.
Layered representations for human activity recognition.
In *Fourth IEEE International Conference on Multimodal Interfaces*, pages 3–8, Pittsburgh, PA, USA, 2002.

[99] OSGi Alliance.
OSGi Service Platform Core Specification, Release 4.1. 2007.
`http://www.osgi.org/Specifications` (accessed 2010-07-02).

[100] OWL API.
`http://owlapi.sourceforge.net/` (accessed 2010-07-02).

[101] A. Padovitz, S. W. Loke, A. B. Zaslavsky, B. Burg, and C. Bartolini.
An approach to data fusion for context awareness.
In *Fifth International Conference on Modelling and Using Context - CONTEXT 2005*, volume 3554 of *Lecture Notes in Computer Science*, pages 353–367, Paris, France, 2005. Springer.

[102] A. Padovitz, A. B. Zaslavsky, and S. W. Loke.
A unifying model for representing and reasoning about context under uncertainty.
In *Proceedings of the Conference on Information Processing and Management of Uncertainty in Knowledge-Based Systems*, pages 1983–1989, Paris, France, 2006.

[103] A. Padovitz, S. W. Loke, A. B. Zaslavsky, and B. Burg.
Verification of uncertain context based on a theory of context spaces.
International Journal of Pervasive Computing and Communication, 3(1):30–56, 2007.

[104] A. Padovitz, S. W. Loke, and A. B. Zaslavsky.
The ECORA framework: A hybrid architecture for context-oriented pervasive computing.
Pervasive and Mobile Computing, 4(2):182–215, 2008.

[105] T. J. Parr.
A Functional Language for Generating Structured Text.
Technical report, University of San Francisco, 2006.

[106] N. Paspallis.
Middleware-based development of context-aware applications with reusable components.

PhD thesis, Department of Computer Science, University of Cyprus, Nicosia, Cyprus, 2009.

[107] N. Paspallis, R. Rouvoy, P. Barone, G. A. Papadopoulos, F. Eliassen, and A. Mamelli. A Pluggable and Reconfigurable Architecture for a Context-aware Enabling Middleware System. In *10th International Symposium on Distributed Objects, Middleware, and Applications (DOA'08)*, pages 553–570, Monterrey, Mexico, 2008.

[108] J. Pearl. Reverend Bayes on inference engines: A distributed hierarchical approach. In *Proceedings of the American Association of Artificial Intelligence National Conference on AI*, pages 133–136, Pittsburgh, PA, 1982.

[109] J. Pearl. *Probabilistic reasoning in intelligent systems: networks of plausible inference*. Morgan Kaufmann Publishers Inc., San Francisco, CA, USA, 1988. ISBN 0-934613-73-7.

[110] PEIS Ecology - Ecology of Physically Embedded Intelligent Systems. `http://www.aass.oru.se/~peis/frameset_page.html` (accessed 2010-07-02).

[111] E. Prud'hommeaux and A. Seaborne. SPARQL Query Language for RDF (Working Draft). Technical report, W3C, 2007.

[112] D. V. Pynadath and M. Tambe. An Automated Teamwork Infrastructure for Heterogeneous Software Agents and Humans. *Autonomous Agents and Multi-Agent Systems*, 7(1-2):71–100, 2003. ISSN 1387-2532.

[113] L. R. Rabiner. A Tutorial on Hidden Markov Models and Selected Applications in Speech Recognition. *Proceedings of the IEEE*, 77(2):257–285, 1989.

[114] E. Ramasso, M. Rombaut, and D. Pellerin. Forward-Backward-Viterbi Procedures in the Transferable Belief Model for State Sequence Analysis Using Belief Functions. In *ECSQARU '07: Proceedings of the 9th European Conference on Symbolic and Quantitative Approaches to Reasoning with Uncertainty*, pages 405–417, Hammamet, Tunisia, 2007.

[115] A. Ranganathan, J. Al-Muhtadi, and R. H. Campbell. Reasoning about uncertain contexts in pervasive computing environments. *IEEE Pervasive Computing*, 3:62–70, 2004. ISSN 1536-1268.

[116] R. Reichle, M. Wagner, M. U. Khan, K. Geihs, J. Lorenzo, M. Valla, C. Fra, N. Paspallis, and G. A. Papadopoulos. A comprehensive context modeling framework for pervasive computing systems. In *8th IFIP WG 6.1 International Conference on Distributed Applications and Interoperable Systems (DAIS)*, volume 5022 of *LNCS*, pages 281–295, Oslo, Norway, 2008.

[117] R. Reichle, M. Wagner, M. U. Khan, K. Geihs, M. Valla, C. Fra, N. Paspallis, and G. A. Papadopoulos.
A context query language for pervasive computing environments.
In *Proceedings of the 5th IEEE Workshop on Context Modeling and Reasoning (CoMoRea) at the 6th IEEE International Conference on Pervasive Computing and Communication (PerCom '08)*, pages 434–440, Hong Kong, 2008.

[118] D. B. Reid.
An algorithm for tracking multiple targets.
IEEE Transactions on Automatic Control, 24:843–854, 1979.

[119] M. Riedmiller and H. Braun.
A Direct Adaptive Method for Faster Backpropagation Learning: The RPROP Algorithm.
In *IEEE International Conference On Neural Networks*, pages 586–591, San Francisco, California, USA, 1993.

[120] J. A. Robinson.
A machine-oriented logic based on the resolution principle.
Journal of the ACM, 12(1):23–41, 1965.

[121] RoboCup Foundation.
`http://www.robocup.org/` (accessed 2010-07-02).

[122] S. J. Russell and P. Norvig.
Artificial Intelligence: a modern approach.
Prentice Hall, 2nd international edition edition, 2003.
ISBN 978-0130803023.

[123] D. Sanchez, M. Tentori, and J. Favela.
Hidden Markov Models for Activity Recognition in Ambient Intelligence Environments.
8th Mexican International Conference on Current Trends in Computer Science (ECN 2007), pages 33–40, 2007.

[124] J. Santos and P. Lima.
Multi-robot cooperative object localization.
In *RoboCup 2009: Robot Soccer World Cup XIII - 13th annual RoboCup International Symposium*, pages 332–343, Graz, Austria, 2009.

[125] T. Schmitt.
Vision-based Probabilistic State Estimation for Cooperating autonomous Robots.
PhD thesis, Department of Informatics, Technische Universität München, 2004.

[126] G. Shafer.
A Mathematical Theory of Evidence.
Princeton University Press, Princeton, 1976.

[127] C. E. Shannon.
A mathematical theory of communication.
Bell System Technical Journal, 27:379–423 and 623–656, 1948.

[128] E. Sirin, B. Parsia, B. C. Grau, A. Kalyanpur, and Y. Katz.
Pellet: A practical OWL-DL reasoner.
Journal of Web Semantics, 5(2):51–53, 2007.

[129] Ph. Smets.
Constructing the pignistic probability function in a context of uncertainty.
In *UAI '89: Proceedings of the Fifth Annual Conference on Uncertainty in Artificial Intelligence*, pages 29–40, Amsterdam, The Netherlands, 1989.

[130] Ph. Smets.
The transferable belief model and other interpretations of Dempster-Shafer's model.
In *UAI '90: Proceedings of the Sixth Annual Conference on Uncertainty in Artificial Intelligence*, pages 375–384, Cambridge, MA, USA, 1990.

[131] Ph. Smets.
The nature of the unnormalized beliefs encountered in the transferable belief model.
In *UAI '92: Proceedings of the Eighth Annual Conference on Uncertainty in Artificial Intelligence*, pages 292–297, Stanford, USA, 1992.

[132] Ph. Smets.
Belief Functions: The Disjunctive Rule of Combination and the Generalized Bayesian Theorem.
International Journal of Approximate Reasoning, 9:1–35, 1993.

[133] Ph. Smets and R. Kennes.
The transferable belief model.
Artificial Intelligence, 66:191–234, 1994.

[134] T. Strang and C. Linnhoff-Popien.
A context modeling survey.
In *UbiComp 1st International Workshop on Advanced Context Modelling, Reasoning and Management*, pages 31–41, Nottingham, UK, 2004.

[135] T. Strang, C. Linnhoff-Popien, and K. Frank.
Cool: A context ontology language to enable contextual interoperability.
In *Proceedings of 4th IFIP WG 6.1 International Conference on Distributed Applications and Interoperable Systems (DAIS2003)*, pages 236–247, Paris, France, 2003.

[136] T. M. Strat.
Decision analysis using belief functions.
International Journal of Approximate Reasoning, 4(5-6):391–417, 1990.

[137] SWEET - Semantic Web for Earth and Environmental Terminology.
`http://sweet.jpl.nasa.gov/index.html` (accessed 2010-07-02).

[138] The Web Service Modeling Language WSML.
`http://www.wsmo.org/wsml/wsml-syntax` (accessed 2010-07-02).

[139] S. Triller.
Plan Recognition and Tracking for Cooperative Autonomous Robots in Dynamic Environments.
Master's thesis, Distributed Systems Group, University of Kassel, Germany, 2010.

[140] D. Tsarkov and I. Horrocks.
Fact++ description logic reasoner: System description.
In *Proceedings of the International Joint Conference on Automated Reasoning (IJCAR 2006)*, volume 4130 of *Lecture Notes in Artificial Intelligence*, pages 292–297, Seattle, WA, USA, 2006.

[141] H. Utz.
Advanced Software Concepts and Technologies for Autonomous Mobile Robotics.
PhD thesis, University of Ulm, Germany, 2005.

[142] H. Utz, S. Sablatnög, S. Enderle, and G. K. Kraetzschmar.
Miro – Middleware for Mobile Robot Applications.
IEEE Transactions on Robotics and Automation, Special Issue on Object-Oriented Distributed Control Architectures, 18(4):493–497, 2002.

[143] H. Utz, F. Stulp, and A. Mühlenfeld.
Sharing Belief in Teams of Heterogeneous Robots.
In *RoboCup 2004: Robot Soccer World Cup VIII*, volume 3276/2005 of *Lecture Notes in Computer Science*, pages 508–515. Springer Berlin/Heidelberg, 2004.
ISBN 3-540-25046-8.

[144] M. Völter, T. Stahl, J. Bettin, A. Haase, S. Helsen, K. Czarnecki, and B. von Stockfleth.
Model-Driven Software Development: Technology, Engineering, Management.
Wiley, 2006.
ISBN 978-0-470-02570-3.

[145] X. H. Wang, T. Gu, D. Q. Zhang, and H. K. Pung.
Ontology based context modeling and reasoning using owl.
In *IEEE International Conference on Pervasive Computing and Communication (PerCom'04)*, pages 18–22, Orlando, Florida, USA, 2004.

[146] Web Dictionary of Cybernetics and Systems.
`http://pcp.vub.ac.be/ASC/indexASC.html` (accessed 2010-07-02).

[147] M. Weiser.
Some computer science issues in ubiquitous computing.
Communications of the ACM, 36(7):75–84, 1993.
ISSN 0001-0782.

[148] D. Werner and U. Schneider.
Taschenbuch der Informatik.
Fachbuchverlag Leipzig, 2004.
ISBN 3-446-21753-3.

[149] D. Winer.
XML/RPC specification. 1999.
`http://www.xmlrpc.com/spec` (accessed 2010-07-02).

[150] World Wide Web Consortium (W3C).
`http://www.w3.org/` (accessed 2010-07-02).

[151] World Wide Web Consortium (W3C).
Extensible Markup Language (XML) 1.0 (Fifth Edition). 2008.
`http://www.w3.org/TR/xml/` (accessed 2010-07-02).

[152] World Wide Web Consortium (W3C).
OWL 1.1 Web Ontology Language - Overview, W3C Member Submission 19 December 2006.
`http://www.w3.org/Submission/owl11-overview/` (accessed 2010-07-02).

[153] World Wide Web Consortium (W3C).
OWL 2 Web Ontology Language Document Overview, W3C Recommendation 27 October 2009.
http://www.w3.org/TR/2009/REC-owl2-overview-20091027/ (accessed 2010-07-02).

[154] World Wide Web Consortium (W3C).
OWL 2 Web Ontology Language - Profiles, W3C Recommendation 27 October 2009.
http://www.w3.org/TR/2009/REC-owl2-profiles-20091027/ (accessed 2010-07-02).

[155] World Wide Web Consortium (W3C).
OWL Web Ontology Language - Semantics and Abstract Syntax. 2004.
http://www.w3.org/TR/owl-semantics/ (accessed 2010-07-02).

[156] World Wide Web Consortium (W3C).
OWL Web Ontology Language - Use Cases and Requirements, W3C Recommendation 10 February 2004.
http://www.w3.org/TR/webont-req/ (accessed 2010-07-02).

[157] World Wide Web Consortium (W3C).
SOAP Version 1.2 Part 0: Primer (Second Edition). 2007.
http://www.w3.org/TR/soap12-part0/ (accessed 2010-07-02).

[158] World Wide Web Consortium (W3C).
Web Services Description Language (WSDL) 1.1, 2001.
http://www.w3.org/TR/wsdl (accessed 2010-07-02).

[159] H. Wu.
Sensor data fusion for context-aware computing using Dempster-Shafer theory.
PhD thesis, Carnegie Mellon University, Pittsburgh, PA, USA, 2004.

[160] T. G. Xiao, X. H. Wang, H. K. Pung, and D. Q. Zhang.
An ontology-based context model in intelligent environments.
In *Proceedings of Communication Networks and Distributed Systems Modeling and Simulation Conference*, pages 270–275, San Diego, California, USA, 2004.

[161] H. Xu and Ph. Smets.
Evidential reasoning with conditional belief functions.
In *UAI '94: Proceedings of the 10th Annual Conference on Uncertainty in Artificial Intelligence*, pages 598–606, Seattle, Washington, USA, 1994.

[162] R. R. Yager.
Decision Making Under Dempster-Shafer Uncertainties.
In *Classic Works of the Dempster-Shafer Theory of Belief Functions*, volume 219 of *Studies in Fuzziness and Soft Computing*, pages 619–632. Springer, 2008.
ISBN 978-3540253815.

[163] B. B. Yaghlane, Ph. Smets, and K. Mellouli.
Directed evidential networks with conditional belief functions.
In *Proceedings of the 7th European Conference on Symbolic and Quantitative Approaches to Reasoning with Uncertainty (ECSQARU 2003)*, pages 291–305, Aalborg, Denmark, 2003.

[164] J. Ye, S. McKeever, L. Coyle, S. Neely, and S. Dobson.
Resolving uncertainty in context integration and abstraction.
In *ICPS '08: Proceedings of the 5th international conference on Pervasive services*, pages 131–140, Sorrento, Italy, 2008.

[165] L. A. Zadeh.
Fuzzy sets.
Information and Control, 8:338–353, 1965.

[166] L. A. Zadeh.
Fuzzy Sets, Fuzzy Logic, and Fuzzy Systems: Selected Papers by Lotfi A. Zadeh.
World Scientific Publishing Co., Inc., River Edge, NJ, USA, 1996.
ISBN 9810224214.

[167] H. Zhang.
Exploring conditions for the optimality of naïve bayes.
International Journal of Pattern Recognition and Artificial Intelligence, 19(2):183–198, 2005.

[168] W. Zhang, F. Chen, W. Xu, and Z. Cao.
Decomposition in Hidden Markov Models for activity recognition.
In *MCAM'07: Proceedings of the 2007 international conference on Multimedia content analysis and mining*, pages 232–241, 2007.

Erklärung

Hiermit versichere ich, dass ich die vorliegende Dissertation selbständig und ohne unerlaubte Hilfe angefertigt und andere als die in der Dissertation angegebenen Hilfsmittel nicht benutzt habe. Alle Stellen, die wörtlich oder sinngemäß aus veröffentlichten oder unveröffentlichten Schriften entnommen sind, habe ich als solche kenntlich gemacht. Kein Teil dieser Arbeit ist in einem anderen Promotions- oder Habilitationsverfahren verwendet worden.

Kassel, im Dezember 2010

Dipl.-Inf. Roland Reichle