Linköping Studies in Science and Technology

Dissertation No. 1204

A Representation Scheme for Description and Reconstruction of Object Configurations Based on Qualitative Relations

by

H. Joe Steinhauer

Department of Computer and Information Science
Linköpings universitet
SE-581 83 Linköping, Sweden

Linköping 2008

ISBN 978-91-7393-823-5
ISSN 0345-7524

Printed by LiU-Tryck, Linköping 2008

Abstract

One reason Qualitative Spatial Reasoning (QSR) is becoming increasingly important to Artificial Intelligence (AI) is the need for a smooth 'human-like' communication between autonomous agents and people. The selected, yet general, task motivating the work presented here is the scenario of an object configuration that has to be described by an observer on the ground using only relational object positions. The description provided should enable a second agent to create a map-like picture of the described configuration in order to recognize the configuration on a representation from the survey perspective, for instance on a geographic map or in the landscape itself while observing it from an aerial vehicle. Either agent might be an autonomous system or a person. Therefore, the particular focus of this work lies on the necessity to develop description and reconstruction methods that are cognitively easy to apply for a person.

This thesis presents the representation scheme QuaDRO (Qualitative Description and Reconstruction of Object configurations). Its main contributions are a specification and qualitative classification of information available from different local viewpoints into nine qualitative equivalence classes. This classification allows the preservation of information needed for reconstruction into a global frame of reference. The reconstruction takes place in an underlying qualitative grid with adjustable granularity. A novel approach for representing objects of eight different orientations by two different frames of reference is used. A substantial contribution to alleviate the reconstruction process is that new objects can be inserted anywhere within the reconstruction without the need for backtracking or re-reconstructing. In addition, an approach to reconstruct configurations from underspecified descriptions using conceptual neighbourhood-based reasoning and coarse object relations is presented.

Acknowledgements

Many people have contributed to this thesis in many different ways and I am grateful to all of them. First and foremost I wish to thank my main supervisor Erik Sandewall who has supported me with advice, discussions and inspiration. I'm deeply grateful for the support of my co-supervisor Christian Freksa, for his encouragement, guidance and valuable feedback on my work. Furthermore, I thank Patrick Doherty for providing an inspiring working environment and moral support. I am grateful for the fruitful discussions with the members of the CoSy group in Bremen. Special thanks go to Thomas Barkowsky, Diedrich Wolter, and Frank Dylla. I thank Eilert Viking and Tom Wiese for careful proofreading, inspiring discussions and encouragement.

I am grateful for the support from my parents Heinz and Elke and my grandmother Annita. I'm very lucky to have such wonderful supportive friends and I would especially like to thank Kirsten, Jens, Petra, Lars, Holger, Bettina, Vaida, Aleksandra, Tom, Dirk, Gabriella, Gert and Ann-Sofi. Once again, I would like to express my gratitude to my boyfriend, Eilert, for keeping things off my back and thus allowing me to focus clearly on my Ph.D. Last but not least I'd like to thank Lillemor, Inger, Lisbeth, Carita, Lise-Lott and Anne for their help concerning the administration around my Ph.D. studies and I thank all my colleagues at AIICS, especially my lab partner Karolina.

This work has been supported financially by CUGS (the National Graduate School of Computer Science) and the Knut and Alice Wallenberg Foundation.

H. Joe Steinhauer
Linköping, September 2008

List of Publications

Publications included in the thesis

- H. Joe Steinhauer. Object configuration reconstruction from incomplete binary object relation descriptions. In A. R. Dengel, K. Berns, and T. M. Breul, editors, *KI 2008: Advances in Artificial Intelligence.* pp. 348-355. Springer, Kaiserslautern, Germany, September 2008.
- H. Joe Steinhauer. Object configuration reconstruction for descriptions using relative and intrinsic reference frames. In *Proceedings of the 18th European Conference on Artificial Intelligence ECAI-08.* pp. 821-822 IOS Press, Patras, Greece, July 2008.
- H. Joe Steinhauer. Qualitative Reconstruction and Update of an Object Constellation. In H. W. Güsgen, G. Ligozat, J. Renz, and R. V. Rodriguez, editors, In *ECAI-06: Workshop on Spatial and Temporal Reasoning*, pp. 11-19. Riva del Garda, Italy, 28 August - 1 September 2006.
- H. Joe Steinhauer. Qualitative Communication about Object Scenes. In *KI-04: Spatial Cognition Doctoral Colloquium.* Bremen, Germany, 14-19 June 2006.

Further publications by the author

- H. Joe Steinhauer. Towards a Qualitative Model for Natural Language Communication about Vehicle Traffic. In H. W. Güsgen, C. Freksa, and G. Ligozat, editors, *IJCAI-05 Workshop on Spatial and Temporal Reasoning*, pp. 45-51, Edinburgh, Scotland, 30 July - 5 March August 2005.
- H. Joe Steinhauer. A Qualitative Model for Natural Language Communication about Vehicle Traffic. In T. Barkowsky, C. Freksa, H. Hegarty, and R. Lowe, editors, *AAAI Spring Symposium 2005: Reasoning with Mental and External Diagrams: Computational Modeling and Spatial Assistance*, pp. 52-57. AAAI Press, Stanford, California, 21-23 March 2005.

- H. Joe Steinhauer. A Qualitative Description of Traffic Maneuvers. In T. Barkowsky, C.Freksa, M. Knauff, B. Krieg-Brückner, and B. Nebel, editors, *Spatial Cognition 2004: Poster Presentations*, Report Series of the Transregional Collaborative Research Center SFB/TR 8 Spatial Cognition, pp. 39-41. SFB/TR 8 Spatial Cognition, Frauenchiemsee, Germany, October 2004.

- H. Joe Steinhauer. The Qualitative Description of Traffic Maneuvers. In H. W. Güsgen, F. D. Anger, C. Freksa, G. Ligosat, and R. V. Rodriguez, editors, *ECAI-04: Proceedings of Workshop on Spatial and Temporal Reasoning*, pp. 141-148. Valencia, Spain, 23-27 August 2004.

Contents

Chapter 1

Introduction

Hello, hello? Can anybody hear me? I seem to be all on my own. I don't know where I am, nor how I got here. My compass is broken and the GPS is malfunctioning.

Please remain calm; your rescue assistant is on its way. For the moment stay right where you are. So that he can find you quickly please describe the positions of the objects around you in relation to your current position and orientation.

1.1 Motivation

In the rescue scenario presented above a configuration of the objects that the lost agent (person or autonomous artificially intelligent agent) sees around him has to be described. The agent is asked to describe the objects' positions in relations to his own position, using his orientation as origin of a thought underlying coordinate system. The description provided ought to enable a listening agent (person or device) to derive an 'imagination' about how the situation looks. This 'imagination' can be regarded as a mental image or an internal diagram of the situation, that becomes externalized when manifested in a sketch or any other form of external representation. The external representation is supposed to be used as a 'qualitative map' of the object configuration that surrounds the lost agent. Therefore, the choice of information to be passed on must cover everything relevant to the task and should leave out unnecessary detailed descriptions that might be more disrupting than helpful. Furthermore, the important parts must be presented in a way that makes it easy for the listener to take them in and develop his own representation of the situation.

Qualitative Spatial Reasoning (QSR) is a subfield of Artificial Intelligence (AI) creating representations for one-, two-, or higher dimensional space by classifying the aspects of space in suitable ways for the particular tasks to be solved. One reason QSR becomes more and more relevant to AI is the need for a smooth and 'human-like' natural language communication between an autonomous agent and a person. People usually use qualitative information rather than quantitative information when they describe scenes, scenarios and configurations of objects. They abstract from details, aggregate objects into groups, and classify objects into categories. Therefore, a human agent can be assumed to describe a situation qualitatively while to a certain degree focusing on the information relevant for the task. Computer-controlled systems, on the other hand, rely on input sensor data and use quantitative information to describe coordinates of object position or distance.

The smooth communication between the autonomous system and a person calls for novel solutions using qualitative knowledge to enable the system to use qualitative information that seems to be preferred by people. The challenge underlying this thesis is to develop a representation scheme with associated methods for description and reconstruction of object configurations.

1.2 Project description

The research project addressed within this thesis aims to develop a representation scheme for the description and reconstruction of object configurations. The configuration description is given by an agent on site. No technical device (compass, GPS, etc.) is used, nor are measuring tools for distances between objects or sizes of objects. The configuration description has to be given from the observer's local perspective. In order to provide a broader description, including more objects and object relations, the agent (human or machine) might move, whereby he changes his location and orientation and continues the description based on these new parameters.

The description should enable a second agent, the listener (human or machine) to reconstruct the object configuration into a global frame of reference. The resulting reconstruction is supposed to be used as an abstracted map of the described environment. In order to be cognitively easy for a human observer and easy to understand for a human listener, the description should be purely qualitative. Furthermore, it should be naturally obtainable and understood by people.

1.3 Research contributions

The main research contributions achieved during the project and presented in this thesis are:

(1) The representation scheme QuaDRO (Qualitative Description and Reconstruction of Object configurations).

(2) The analysis of fourteen qualitative spatial calculi for suitability to this project.

QuaDRO does not use any of the presented calculi in their original version, but merges many aspects of different approaches into one novel representation scheme, which includes:

1. A specification and qualitative classification of information available from different local view points providing an object configuration description that allows for the configuration's reconstruction into a global frame of reference.
2. Methods that reconstruct an object configuration from a given qualitative description into a global frame of reference, which includes:
    - A technique that decides the quantitative position within the reconstruction a qualitatively described object has to be inserted.
    - A technique that allows the representation of objects at uncertain (coarse) positions, a problem that occurs when the object configuration description is underspecified, which means that not all object relations are provided within the description.
    - A novel approach which represents eight different orientations of objects using different reference frames for objects with orientations aligned to the underlying coordinate system, and objects with orientations at angles to the underlying coordinate system. This approach simplifies the reconstruction process enormously.

As a proof of concept for many of the presented methods and techniques, a technical prototype has been built.

1.4 Thesis outline

Firstly chapter 2 introduces the research areas of qualitative reasoning, qualitative spatial reasoning, and diagrammatic reasoning, which are the main areas that the research provided within this thesis falls into. It also gives some background information on the WITAS project that was the umbrella project for the research presented in this thesis. Chapter 3 introduces fifteen qualitative spatial calculi that have been developed over the years to fulfill different qualitative spatial reasoning tasks, or to provide a general approach on qualitative spatial representations. Chapter 4 analyzes fourteen of these calculi with regard to their usability for object configuration description from local viewpoints and their reconstruction into a global frame of reference from just the established qualitative description. The chapter analyzes whether a calculus can express all relations necessary while using as little information as possible to be cognitively easy to use for a person. Chapter 5 follows with an introduction to describe the general aspects of a qualitative object configuration description. It considers how people in psychological experiments choose perspectives while describing object configurations, how people divide space into qualitative classes, and it provides a description strategy based on the results of these experiments. Furthermore, it introduces some results from mental model theory that help the understanding of typical problems and human preferences concerning the object configuration reconstruction process, and provides a list of requirements for the reconstruction process.

The representation scheme QuaDRO, originated during this research project is presented in the following four chapters. QuaDRO uses nine qualitative equivalence classes to describe positional relations between objects, which result in cognitively easy to produce qualitative descriptions of object configurations. In addition, QuaDRO provides the ability to reconstruct the configuration into a global frame of reference. QuaDRO is introduced in chapter 6. This chapter first describes the principal ideas and techniques using a simple example of a situation for orientationless objects. Chapter 7 follows expanding the description for the situation of objects with eight different orientations. In chapter 8 the QuaDRO technical prototype is presented and the scenario, introduced in chapter 5, is described using QuaDRO. For all scenarios presented so far a complete object configuration description is necessary. Therefore chapter 9 follows with the introduction of several techniques used to reconstruct object configurations from underspecified descriptions. Finally, yet importantly, chapter 10 summarizes the previously presented work followed by a discussion of the research results and finishes with the outlook for continued research in the future.

Chapter 2

Background

The work presented here is strongly inspired by the WITAS project [Doh00; Wzo06b] and its follow ups [Wzo06c; Dur07b]. The WITAS project, which is further described in section 2.1, was a long-term project at the computer science department of Linköping's university. The aim of the project was to develop a high-level control system for an autonomous helicopter that would be able to fulfill several tasks on demand and communicate with the human operator or mission leader through natural language.

The WITAS Dialogue Technology Project [San03a; Lem01b; Lem01a; Lem02], further described in section 2.2, focused on developing a dialogue system for spoken and written natural language dialogue between a human operator and the helicopter. Natural language dialog between an autonomous system and a person might be needed when autonomous agents and humans work together, for instance while executing a rescue mission, where the autonomous agents are not necessarily helicopters. The overriding scenario which this thesis is concerned with comes from the following:

Imagine a disaster area, for instance a rapidly spreading forest fire, a flood or an area after an earthquake. Rescue experts are busy organizing rescue teams and equipment. To do their job as effectively as possible it would be advantageous if they knew what the disaster area looked like, for instance, where people are trapped, where the water level is rising, or in what direction the fire is spreading. In contaminated areas where it is extremely difficult for humans to investigate the disastrous terrain, autonomous agents like all-terrain vehicles and helicopters can be deployed for the job of exploring the area and distributing survival kits to trapped people.

Normally human rescue experts are only experienced in their specific rescue field. It would be time consuming and a source of error if the experts needed to learn to interpret all the data collected by the autonomous systems. It would also be a disadvantage if additional specialists were needed to translate the data into natural language. Instead, the autonomous system itself should be able to interpret the collected data and translate its findings into human natural language.

Nowadays the deployment of expert systems is also growing in many application areas where experts are rare or expensive. Therefore even human rescue team leaders in the field might have to communicate with an expert system instead of a person. The team leaders' surroundings might be very stressful which demands high levels of concentration and it would be disadvantageous to put an additional load onto them by requiring communication to the expert system in a particularly unfamiliar way. In these cases, we want the expert system to be capable of a calm dialogue, requesting required information, and of supporting and alleviating the leaders' work.

To achieve the realization of theses scenarios, the autonomous- or expert system must not only be capable of a natural language dialogue, it must also be able to use human concepts, for structuring space, classifying objects, giving directions and describe object configurations. The latter has been chosen for this research project. A representation scheme, presented in the chapters 6, 7, 8, and 9 had to be built, that without much cognitive effort is usable by people and at the same time is clearly structured to be implemented into a system. This undertaking mainly touches the research areas of qualitative reasoning, qualitative spatial reasoning, mental images, and diagrammatic reasoning, which are briefly described in the sections 2.3, 2.4, and 2.5 below. Object's relationships to each other can be described differently, depending on the frame of reference used within the description. The term frame of reference and the different types of frames of reference are described in section 2.6.

2.1 The WITAS project

As mentioned before, the WITAS project [Doh00] had the goal to develop a high-level control system to a fully autonomous UAV (unmanned aerial vehicle), that could fulfill several tasks on demand. Its main goal was to develop an integrated hardware/software vertical take-off and landing (VTOL) platform. The platform used in the project was a 21hp two-stroke engine powered Yamaha RMAX helicopter already equipped by Yamaha Motor Company with an attitude sensor (YAS) and an attitude control system (YACS). The radio controlled helicopter, shown in figure 2.1 is commercially available in Japan. Including the main rotor, the helicopter has a total length of 3.6m and a maximum take-off weight of 95kg. Three PC104 which carry the primary control system, the image processing system and the deliberative/reactive system have been added to the helicopter. A wireless modem, a GPS, a barometric altitude sensor, sonar and infrared altimeter and a compass are part of the primary control system. The image processing system uses a color CCD camera that is mounted on a pan-tilt unit, a video trans-

Figure 2.1: *The Yamaha Rmax helicopter used in the WITAS project.*

mitter and a mini DVD recorder. The deliberative/reactive system is connected via Ethernet using CORBA (Common Object Request Broker Architecture) to the other PCs [Doh00; Hei04b; San03a]. The WITAS project that started 1997 and successfully ended in 2006 was headed by Patrick Doherty and Erik Sandewall. The project was financed by the Wallenberg Foundation, which gave it its name (Wallenberg laboratory for Information Technology and Autonomous Systems). Several follow up projects are currently building on the developed architecture.

These and the WITAS project itself host several multi disciplinary research areas such as robot architectures and UAV control [Doh04; Con04; Wzo06a; Mer06], fuzzy control [Kad04a] and fuzzy gain-scheduling [Kad04b], helicopter dynamics [Dur07a], vision [Nor02a], image processing [Nor02b], visual navigation [Mer04], knowledge processing [Hei04b; Hei04c; Hei04a; Hei05], sensor and symbol integration [Hei07a; Hei07b], planning [Pet04; Pet06; Wzo06a], dialogue systems [Lem01b; Lem01a; Lem02; San03a; San03b; San05] and case-based reasoning [Eli05a; Eli05b; Eli05c; Eli06; Eli07]. Further more, the project included research in qualitative spatial reasoning, whose results are presented in this thesis and can be read in [Ste04a; Ste04b; Ste05a; Ste05b; Ste06a; Ste06b; Ste08a; Ste08b].

The long-term motivation for the WITAS project is that a UAV with high-level autonomy is likely to be useful in some of the following ways.

2.1.1 Mission tasks

One of the tasks the helicopter should be able to fulfill is traffic monitoring and surveillance. If necessary it should also be able to interact with the traffic, for instance to prevent accidents or to guide an ambulance quickly and safely to a desired place. The helicopter has to 'see' the traffic situation and to 'understand' what is going on on the ground. From this 'understanding' it draws conclusions as to how the situation will develop in time. According to its conclusions and to the given task it has to fulfill, it deliberates a plan for its own actions. Then it executes this plan while frequently updating the information of the situation on the ground and perhaps changing its plans according to new situations.

The helicopter can also be used for tasks such as photogrammetry, surveying, checking the quality of a retaining wall in order to detect cracks or to carry out patrol flights over mountains to recognize surface changes in order to predict avalanches. Observing an ongoing catastrophe like a rising flood or a volcano eruption are other possible tasks. The helicopter then uses the information it gets as visual input and along with other information, for instance how the ground is formed in this area, to predict how the catastrophe might develop.

Furthermore, the helicopter can serve as an emergency service assistant in catastrophic situations, for instance in a contaminated environment where people are trapped, to support them with necessary aid. It could also be used in catastrophic areas to find the right places where a help team is needed in order for all available help personnel to act efficiently. A lot of valuable rescue time can be saved by being able to guide rescue teams directly to victims without having the teams searching by themselves.

2.1.2 Natural language communication

Natural language communication is getting more and more important for artificially intelligent systems. Imagine for example the autonomous helicopter in the WITAS project patrolling over the rush-hour traffic and reporting to the police headquarters. What's more, you can think of a driver support system in your car that not only gives you advice where to drive but also interprets the traffic around you in order to warn you of dangerous situations. It can even help you to avoid them by predicting the potential tactics of other road users.

What we would like to achieve is that the system talks to us in natural language, using our vocabulary and concepts that we are used to when we talk about traffic. The helicopter should say for instance ‘The red Porsche is driving along Main Street, now passing the church and will soon turn right into Park Avenue.’ And you might expect your driver support system to express something like: ‘Be careful, you should not overtake here, the car in front might turn left soon, its indicator is probably broken.’ To be able to speak like this the system does not only have to have the right interpretation of the information that it gets from its sensors, it must also be able to express it in a way that precisely states the information that is important to us.

Another application is a group of agents that work together in a disaster area. The group consists of human experts that are needed in the specific type of catastrophe. This could be experts for earthquake rescue or fire specialists in a quickly spreading forest fire. The experts work together with autonomous robots such as all-terrain vehicles and helicopters that explore the area and report what kind of help is needed and where. They distribute the correct equipment to rescue parties or survival kits to people trapped.

In the optimal scenario, the communication between the different kinds of agents takes place in natural language. This is advantageous because the human experts are often only experienced in their specific rescue field; they don’t need to be trained how to communicate with the autonomous agents. Thus the experts can concentrate solely on their work instead of on communication; and every available expert can be deployed on the scene and not just those experts with the relevant training in communication. Therefore, the problem with how the communication is carried out has to be solved by the autonomous agents; they have to learn how to communicate with the experts, using everyday language with the terminology of the rescue field, and not the other way around.

> *‘Natural-language and multimedia dialogue with an autonomous robot is a challenging research problem which introduces several important issues that are not present in, for example, dialogue with a database or a service provider such as an automated travel agency.’* [San03a]

2.2 The WITAS dialogue systems

The WITAS Dialogue Technology Project focuses on developing a dialogue system for natural language communication, both spoken and written with the UAV. Several dialogue systems have been developed through the years of the project. The first was the WITAS Dialogue System [Lem01b; Lem01a; Lem02] that was a system for multi-threaded robot dialogue using spoken I/O. The system was run against the simple UAV environment simulator DOSAR (Dialogue Oriented Simulation and Reasoning) [San03a].

The following research dealt with developing operator assistants. An operator assistant is a system that communicates with the operator on the one side and with one or several autonomous systems on the other side. This means that communication becomes more convenient for the operator as he can talk to the operator assistant, which then translates the orders of the operator to the autonomous systems.

The second system used in the WITAS Dialogue Technology Project is the Robotic Dialogue Environment (RDE), which is an environment providing the possibility for a natural language dialogue in restricted English, both spoken and written. In addition, it provides video, which is meant to be the video taped by the autonomous agent to whom the operator communicates. The operator has the possibility to point into the video to define points ('fly here'), areas ('circle in this area') or trajectories ('fly this way').

RDE consist of three subsystems: The Autonomous Operator Assistant (AOA), which contains the Dialogue Manager and a Speech and Graphical User Interface (SGUI). Second, the Robotic Agent that consists of a Robotic World, which can be either the actual UAV system and the world it is flying in or a simulator called Dialogue-enabled Operator Support Agent for Robots (also abbreviated to DOSAR). The third system is a Development Infrastructure, which is needed to provide services for demonstration, validation and development of the dialogue system [San05].

A third and latest dialogue system used is called CEDERIC, which is an abbreviation for Case-base Enabled Dialogue Extension for Robotic Interaction Control [Eli05a; Eli05b; Eli05c; Eli06; Eli07] and is based on the dialogue manager in the RDE system. CEDERIC focuses on discourse handling and learning using Case-Based Reasoning (CBR). CEDERIC is able to deal with phrases that have not been stored in the system before. A discourse model keeps track of the dialogue history and can handle references to previously mentioned phrases and even handle sub dialogues. CEDRIC learns from explanations by putting questions to the user about words it does not understand. Therefore, CEDERIC's performance improves over time [Eli06; Eli07].

2.3 Qualitative reasoning

> *'Qualitative reasoning is the area of AI which creates representations for continuous aspects of the world, such as space, time, and quantity, which support reasoning with very little information.'* [For97]

People frequently use qualitative reasoning in their everyday life. They understand relations in the world; they draw conclusions and make predictions of what is likely to happen using incomplete and imprecise information on the basis of estimations and rules of thumb. This section briefly describes the most important terms in qualitative reasoning that are frequently used within this thesis.

Qualitative knowledge

Often differences in values are given in comparison to each other instead of in exact measures. Tom is taller than Mary, Stockholm is further away from Rome than from Oslo and Sally gets a much higher salary than Peter, are fragments of qualitative knowledge.

Quantitative knowledge

The counterpart to qualitative knowledge considers the exact and measurable values. Tom is 10cm taller than Mary, Stockholm is 2030,24km further away from Rome than from Oslo, and Sally earns $25000 more per year than Peter.

Qualitative classification

It is human to categorize and to classify. We speak of short versus tall people or of rich versus poor people. We talk about the temperature in the room as cold or hot. If we want to give information that is more specific, we divide the object space into further categories and classify the temperature as too cold, cold, normal, hot or too hot. Qualitative classification divides the space in classes at the points where the differences suddenly get relevant. This would be that we classify the temperature as too cold when we start freezing and consequently put on a sweater or that we classify the temperature as too hot when we start sweating and want to get rid of the sweater again. Therefore, the borders separating the different classes might often be values that result in a change of consequences for us and so qualitative knowledge can be viewed as aspects of knowledge which critically influence decisions [Fre93].

One of the challenges for qualitative reasoning is to find the points where a change from one class to another occurs. Certainly, those borders are often individual but usually people do not have any problem dealing with them and even prefer qualitative information to quantitative information in many cases. It is just easier for us to say that it is pretty cold today instead of trying to guess the current temperature.

Sensor equipped computers on the other hand usually collect quantitative data. We can use this data directly or let the computer or even technically much simpler devices, classify them into qualitative classes that are convenient for our purpose. A simple relay in the thermostat of a heater ‘knows’ when it gets too cold and starts the radiator and the computer of a nuclear power plant flashes on a warning light, when the temperature in the reactor exceeds the normal temperature bounds.

Qualitative spatial reasoning

Qualitative spatial reasoning (QSR) is the subfield of qualitative reasoning that is concerned with spatial information.

Temporal reasoning

Temporal Reasoning [All83] considers reasoning about time. Temporal reasoning can be regarded as a special case of spatial reasoning in one dimension, whereby the direction of time is a crucial restriction.

Spatio-temporal reasoning

The term spatial reasoning often appears in connection with temporal reasoning and is shortened to spatio-temporal reasoning which considers reasoning in space and time.

2.4 Mental images

The term *mental images* is frequently used in literature about natural language systems, dialogue systems, and vision systems. According to Barkowsky [Bar02] mental images are spatio-analogical representations in working memory which are constructed from knowledge in long-term memory. Ideally, a vision system should describe the scene it sees in a way that the listener's mental image of the scene becomes identical with what it would be if the listener were to watch the scene himself.

Arens and Nagel [Are03] want their system to impose the correct mental image on a human listener. Whereas Herzog and Wazinski [Her94b] produce the mental image within the listener model that the system maintains to keep track of what the listener already knows. Sproat [Spr01] wants his system WordsEye to consider even common sense knowledge and to apply additional context knowledge to given textual description in order that the resulting mental image is as close as possible to a human one. Mental images are to some degree individual as Johansson [Joh05] realized during the work with the system CarSim.

All this work implies that the mental image of the described scene is meant to be a representation of the scene that includes visual and spatial information of objects and their relations.

2.4.1 Visual mental imagery

According to Kosslyn [Kos05] visual mental imagery is, unlike visual perception, a set of representations that lead to the experience of a stimulus, without the presence of any appropriate sensory input. Visual perception on the other hand, occurs only when an appropriate stimulus is presented. As the Stanford Encyclopedia of Philosophy, states, mental imagery is defined as a quasi-perceptual experience that occurs in the absence of the appropriate external stimuli but resembles perceptual experience [Nig05].

> *'The belief that such mental representations are real is justified in the same sort of way that belief in the reality of electrons, or natural selection, or gravitational fields (or other scientifically sanctioned 'unobservables') is justified.'* [Nig05]

Psychologists, cognitive scientists and philosophers have had complex and fractious debates on the different interpretations of mental imagery. Cognitive scientists today believe that mental images play an essential role in human mental economy [Nig05]. In the area of Artificial Intelligence, mental imagery can be seen as a problem-solving paradigm, thus becoming more and more important. The concept of computational imagery has been proposed by Glasgow and Papadias [Gla92] as a potential application to problems that involve mental imagery for people.

2.5 Diagrammatic reasoning

The area of diagrammatic reasoning is concerned with the question of how humans and machines can represent information using diagrams and then use this representation for further reasoning.

Diagram

A diagram is an abstract pictorial representation of information e.g. a sketch, a bar chart, or a map, whereas a photograph or a video is not a diagram. Therefore, a diagram contains less detailed information than a photograph but still contains spatio-visual information [Nar97].

Internal diagrams, external diagrams

Diagrammatic reasoning as well as mental imagery gains more and more interest in Artificial Intelligence. Human-machine interaction tasks would be supported by a representation that suits the person as well as the computer. People use mental images when dealing with spatial reasoning tasks [JL91; JL01]. These mental images can be seen as internal diagrams whereas diagrams drawn on paper are external diagrams.

Barkowsky et al. [Bar05] explain that diagrammatic reasoning has been investigated over the last three decades from three distinct perspectives. These are computational modelling of cognitive processes, spatial assistance, which would be human spatial reasoning, spatial interaction, and communication about spatial issues, and actions in space and the interplay of mental representations and external diagrams. It is not yet clear to what extent the external diagrams drawn by persons match their internal diagrams. However, diagrams normally concentrate on just the information that is important for the reasoning task, leaving out the details unnecessary for the task. For instance, a flowchart of a computer programme completely abstracts from the programming language and memory management.

Bertel [Ber05] points out that collaborative human-computer reasoning creates asymmetric reasoning situations due to dissimilar and sometimes even incomparable reasoning faculties. He goes on, that diagrams and sketches provide unique opportunities to get an idea of mental reasoning. Actions performed on an external diagram reflect actions carried out mentally on the internal diagram. We might be able to extrapolate from collected data of perceptual and manipulative actions that a human reasoner performs on an external diagram, the corresponding mental mechanisms carried out on the mental representations. As diagrams are a natural means for human-human interaction, they have great potential for human-computer interaction even though much modelling effort is required to synchronize the partners' reasoning.

2.6 Frame of reference

The origin of the term *frame of reference* dates back to the 1920s and the Gestalt theories of perception [Wer12; Wer22]. Many different research areas, e.g. philosophy, brain sciences, linguistics, psychology, vision theory, or visual perception, have been and are still concerned with frames of reference. Unfortunately, each of them has defined its own terminology. Levinson [Lev96] gives an overview of the different terminologies, their exchangeability and their contradictions.

In general, a frame of reference is used to describe a spatial relation between the *figure* (*referent/target object*) and the *ground* (*relatum/reference object*). The position of the figure is described in a binary or ternary relation to the ground. In a binary relation, the origin of the coordinate system within which the figure is located is established at the ground object, whereas in a ternary relation the origin of the coordinate system is situated at a point (*viewpoint*) different from figure and ground.

Levinson [Lev03] emphasizes that it is not the type of object that is talked about, but the underlying coordinate system, that people of certain cultures are used to, which invokes distinctions between frames of reference. He distinguishes three main frames of reference, which are *intrinsic*, *relative*, and *absolute*. It seems that these three are the only frames of reference used in human natural language. Nevertheless, not all three are used in every language. Whereas European languages mostly use relative frames of reference (viewpoint centered) many other languages prefer absolute frames of references even for indoor environments and

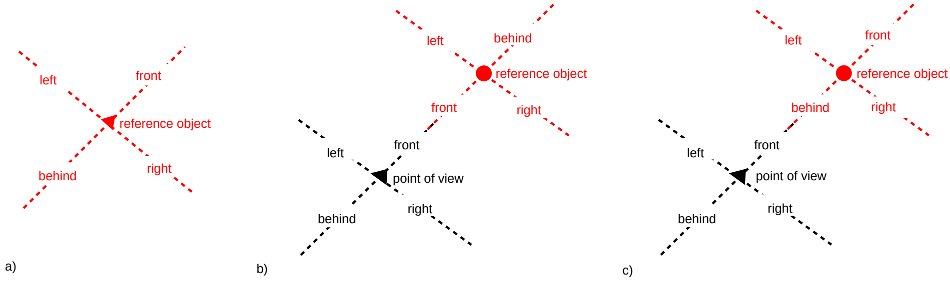


Figure 2.2: *Frames of references. a) intrinsic frame of reference b) relative frame of reference used in English c) relative frame of reference used in Hausa.*

some Australian languages use topological terms like 'under' or 'in' to describe spatial object relations [Lev03].

Intrinsic frame of reference

For an intrinsic frame of reference, shown in figure 2.2a), the coordinate system is object centered and oriented by the features of that object. Objects often have a prominent front, for instance a person's or animal's face, a car's headlights or a house's main entrance. According to this natural front, the coordinate system is established, and everything that is on the side of the object defined as its front, is *in front of* the object. Everything on the opposite side is therefore *behind* the object and the regions *to the left* and *to the right* are established accordingly. Thereby the intrinsic frame of reference imposes a binary spatial relation between the figure and the ground.

Motion can be expressed using an intrinsic frame of reference. *The car is moving backwards* or *the crab is walking sideways* in relation to its own (intrinsic) orientation. For objects not having an intrinsic front, back and sides, for instance a ball, an intrinsic frame of reference is often attached to them by the direction of their movement. *The goalkeeper jumped in front of the ball* in relation to the ball's direction of movement. Intrinsic relations are generally not transitive. If object B is left of object A and object C is left of object B, nothing can be said about object C's relationship to object A, because that depends on object A's orientation.

Relative frame of reference

A relative frame of reference imposes a ternary spatial relation between figure, ground, and a viewer's point. A primary coordinate system is established at the viewer and a secondary coordinate system at the ground, which normally inherits at least some of the primary system's coordinates.

In English, the sentence '*The ball is in front of the tree*' means that the ball is between the viewer (speaker) and the tree. As shown in figure 2.2b) are front and back 'mirrored' at the ground (tree) and the tree can be seen to be facing the viewer, whereas left and right are not exchanged and denote the viewer's left and right. This system however, does not appear to be entirely natural, as children naturally use a system where left and right are also exchanged much earlier in their development but do not learn the mixed system before the age of five or six [Lev03]. On the contrary, Levinson [Lev03] considers the relative frame of reference where the viewer's coordinate system is shifted without any rotation or reflection onto the ground object, as shown in figure 2.2c), as natural. He points out that this system is used in many languages, for instance in the chadic language Hausa [Hil82], an Afro-Asiatic language that is spoken by 24 million people as first language and a further 15 million people as second language [wik].

As long as the viewpoint does not change, the relations in a relative frame of reference are transitive and conversable. If object B is left of object A and object C is left of object B the conclusions that object C is left of object A, object B is right of object C, object A is right of object B, and object A is right of object C can be drawn.

Absolute frame of reference

An absolute frame of reference uses fixed bearings, for instance the cardinal directions *north*, *south*, *east*, and *west*. However, several other absolute frames of reference are used in natural language. Brown [Bro00] reports that the people living in Tenejapan Tzetal use an absolute frame of reference that is given by the slope of the landscape and manifests into the terms *uphill* and *downhill*. An absolute frame of reference is fixed. It depends neither on the viewer's position nor on the object's orientation. The binary relations between figure and ground are transitive and conversable.

2.6.1 Projection-based frame of reference

A projection-based frame of reference is in a sense similar to the representation of position in latitude and longitude [Fra91]. A line, for instance from north to south, shown in figure 2.3a), divides the plane into the two half planes that represent the directions *east* and *west*. A line from east to west establishes the half planes representing the directions *north* and *south* shown in figure 2.3b). If both lines are applied at the same time the half planes overlap and result in the four directions *north-west*, *south-west*, *north-east* and *south-east* presented in figure 2.3c).

Hernandez [Her94a] describes a relative frame of reference for the directions right, left, front and back. He draws a straight line between the viewpoint and the reference object. A target object can be left, right or colinear^{lr} to this line, as shown in figure 2.3d). The index lr indicates that the relation is collinear on the line that separates left from right. Establishing a perpendicular line at the

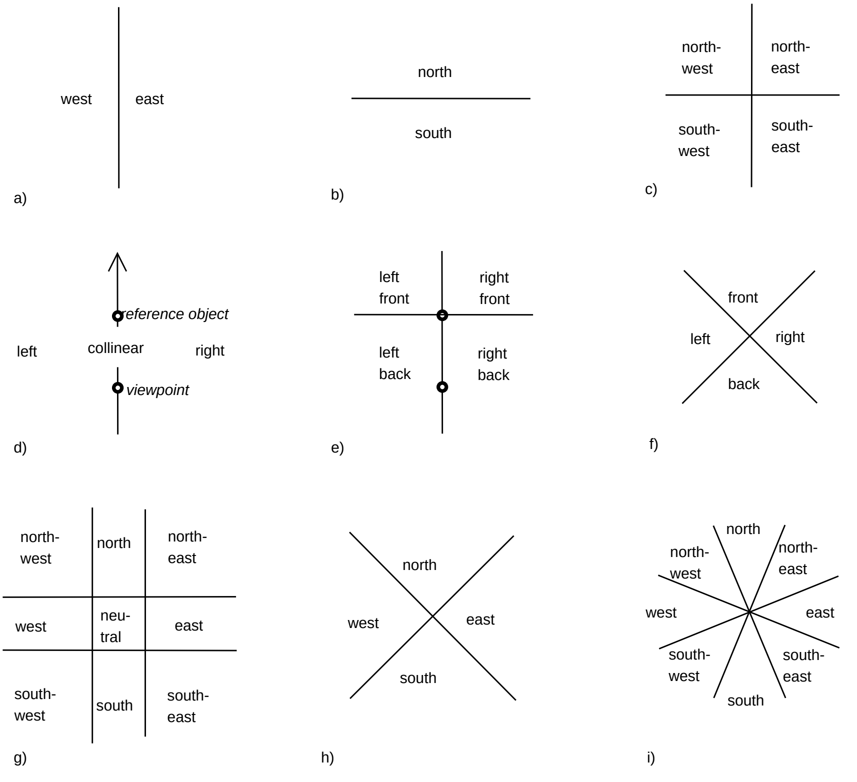


Figure 2.3: *Frames of references. a), b), and c) projection-based d) and e) relative frame of reference by Hernandez [Her94a] f) rotated frame of reference [Her94a] g) projection-based with neutral zone h) and i) cone-based.*

reference object the space is divided into the areas front, back, and colinear^{fb}. Using both lines together the four regions leftfront, rightfront, rightback, and leftback, shown in figure 2.3e) appear.

Projection-based frame of reference with neutral zone

The frame of reference shown in figure 2.3g) is projection based with neutral zone [Fra91]. It can be seen as a representation of the reference object by its two-dimensional bounding box or a buffer zone around it. The separation into the areas *east*, *west*, *north*, and *south* takes place at the east, west, north, and south edges of the reference object, and the object itself is in between all areas in a neutral position. Using four directions the plane around the reference object divides into nine areas. Four are non-overlapping one-directional (*east*, *west*, *north*, *south*) and four are overlapping in two directions (*north-east*, *north-west*, *south-east*, *south-west*), and the *neutral* area.

Cone-based frame of reference

A cone-based frame of reference, like the two presented in figure 2.3h) and 2.3i), represents the areas for each direction as sections of an infinite circle originating at the reference object [Fra91]. The further away from the reference object, the wider the area of a direction becomes. Frank [Fra91] points out that this approach captures the human intuitive concept of moving in a direction.

Hernandez [Her94a] establishes the cone-based frame of reference, shown in figure 2.3f) by rotation of the previously established projection-based frame of reference described in section 2.6.1 and shown in figure 2.3e), by 45 degrees and renaming the areas as front, back, left and right.

This chapter introduced some background information to the thesis project and defined some frequently used terms. Chapter 3 introduces fifteen spatial calculi, which are subsequently analyzed in chapter 4 with regards to their ability to describe object configurations suitable for the task introduced in chapter 1.

Chapter 3

Qualitative spatial calculi

Over the years, several qualitative spatial calculi have been developed to reason with qualitative spatial information. Most of them concentrate on reasoning about one particular aspect of space, for instance topology (reasoning about regions) [Ege91; Ran92; Pap95], or position (reasoning about orientation and direction) [Güs89; Fra91; Fre92b; Lig93; Sch95; Bal98; Mor00; Ren04; Mor05; Dyl05]. Reasoning about position can be classified further to reasoning about configurations of point objects [Fra91; Fre92b; Lig93; Ren04; Mor05], lines or line segments [Sch95; Mor00; Dyl05], rectangles [Güs89; Bal98] or objects' positions within in a grid [Rag03]. For an overview about qualitative spatial calculi, see for example [Coh01; Fre93].

Usually a system of qualitative relationships between spatial entities is developed, which covers the particular spatial aspect that is of interest to a degree that appears useful from an applicational or a cognitive point of view [Ren02]. Qualitative spatial calculi are usually designed to infer implicit knowledge from given knowledge. This means, given the relationship of object A with regard to object B, *R(A,B)*, and the relationship of object B with regard to object C, *R(B,C)*, the relationship of object A with regard to object C, *R(A,C)*, is in question. Often a composition table (which Allen [All83] calls a transitivity table) is provided. In such a table the relationships *R(A,B)* and *R(B,C)* are given in row and column and the intersection entry of the two reveals all possibilities for the relationship *R(A,C)* that might hold under certain given premisses.

In many cases, the derived information is ambiguous as it is possible to infer several alternatives from the underlying knowledge. Consider the following example in one-dimensional space (time) as an illustration: Imagine a group of people, all arriving separately at a friend's house. If one knows that Mary arrived before Tom, *before(Mary,Tom)*, and that Annie arrived before Mary, *before(Annie,Mary)*, one can draw the unambiguous conclusion that Annie arrived before Tom, *before(Annie,Tom)*. If we know that Chris also arrived before Mary, *before(Chris,Mary)* we can be certain that Chris arrived before Tom, *before(Chris,Tom)*. However, we do not know if Chris arrived before, at the same time as, or after Annie. In this case the relationship for Chris and Annie is the disjunction of all possibilities, *before(Chris,Annie)* $\vee$ *same(Chris,Annie)* $\vee$ *after(Chris,Annie)*.

To several qualitative spatial calculi, for instance [All83; Fre92b; Mor00; Sch95; Ran92; Rag03] standard constraint-based reasoning techniques can be applied. Historically constraint-based reasoning techniques were first developed for temporal reasoning [All83] and later for spatial reasoning [Ren98; Ren01; Ren07].

A typical constraint-based reasoning problem (constraint satisfaction problem [Mac77]) is the consistency problem CSPSAT($\mathcal{S}$), of deciding whether a given set of spatial constraints over the set $\mathcal{S}$ of basic relations is consistent. An instance of CSPSAT($\mathcal{S}$) is a given set $\mathcal{V}$ of n variables over the domain $\mathcal{D}$, further a given set Θ of binary constraints in the form $x\mathcal{R}y$ where $\mathcal{R} \in \mathcal{S}$ and x, y $\in \mathcal{V}$. The question is if there exists an instantiation of all variables in Θ with values from $\mathcal{D}$ that possibly satisfies all constraints in Θ.

The remainder of this chapter provides an overview of several qualitative spatial calculi for orientation (reasoning about point configurations and about points in relation to a line or line segment). Furthermore, some calculi for rectangle relations and also for topological reasoning about rectangle relations and cardinal directions in two-dimensional space are introduced. As several of them are two-dimensional extensions of Allen's Interval Calculus [All83], the Interval Calculus itself is presented first.

3.1 Interval Calculus

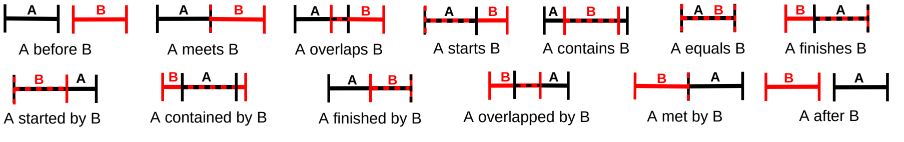


Figure 3.1: *The 13 relations of the Interval Calculus.*

Allen [All83] introduced the interval-based temporal logic together with a computationally efficient reasoning algorithm based on constraint propagation. The calculus describes all possible relations between two temporal intervals, A and B, with thirteen jointly exhaustive and pairwise disjoint (JEPD) terms. The thirteen relations: *before*, *meets*, *overlaps*, *starts*, *contains*, *equals*, *finishes*, *started by*, *contained by*, *finished by*, *overlapped by*, *met by*, and *after* are illustrated in figure 3.1. A typical reasoning task for this calculus is to determine the relationship of interval A to interval C, given the relationships of A to B, and B to C.

The composition table (transitivity table) for the Interval Calculus, revealing all possible relationships for A to C, is presented in Allen [All83].

3.2 Line Segment Relations

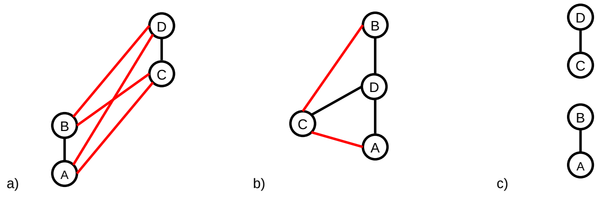


Figure 3.2: *Line segment relations.*

An interval can be seen as a line segment. In Allen's Interval Calculus, all line segments are collinear. Schlieder [Sch95] adds 50 relations for line segments in the plane. Fourteen are called *line segment relations* and describe relations of line segments with starting point and ending point at general positions. 36 further relations describe all cases where three of the points are collinear. All 63 relations together are referred to as *line segment relations in the broad sense.*

The approach is based on the idea that given two line segments AB and CD, four triangles, ABC, ABD, ACD, and BCD, that are each given in lexicographic order, can be established between the four points A, B, C and D. The orientation of a triangle is given as + for counter-clockwise order of the points, - for clockwise order, or *0* in the case where all three points are collinear. If all four points are in general positions, the line segment relationship is given by the orientation tuple of the four triangles, presented in lexicographic order. The orientation information is then encoded into a natural number. Each number from 0 to 15 describes a unique relation. The relations 5 and 10 are not geometrically realizable and 14 relations for points in general positions remain. Figure 3.2a) shows the relation with number 3, that is derived from the four triangle orientations $[ABC]$ = -; $[ABD]$ = -; $[ACD] = +$; $[BCD] = +$. The relation tuple - - + +, coded as a binary number where + converts to 1 and - converts to 0, gives 0011_{bin} which equals 3_{dec}.

When three points are collinear, 36 further relations are possible. Figure 3.2b) shows one example of this case: $[ABC] = +$; $[ABD] = 0$; $[ACD]$ = -; $[BCD] = +$. If all four points are collinear, no triangles can be established. An ordering between the points can be achieved by classifying each pair of points by the terms of *before* (-), *at the same position* (*0*), and *after* (+). This is the case in figure 3.2c) where the ordering of the points is $[AB] = +$; $[AC] = +$; $[AD] = +$; $[BC] = +$; $[BD] = +$; $[CD] = +$. All possible combinations for collinear points result in 13 different cases, which are similar to Allen's temporal relations [All83].

Schlieder is mostly concerned with the subset of 14 relations for line segments whose endpoints lie in general positions. In [Sch95] he gives the conceptual neighbourhood graph for those relations. The neighbourhood graph considers movement of one point of one line segment at a time. The neighbourhood graph can be used for motion planning, where an object has to be moved around another object, both presented as line segments. The object's start and end positions are given as line segment relations. The task is to find the shortest path in the conceptual neighbourhood graph not containing a collision (touching or crossing) of the two line segments.

3.3 Dipole Relation Algebra

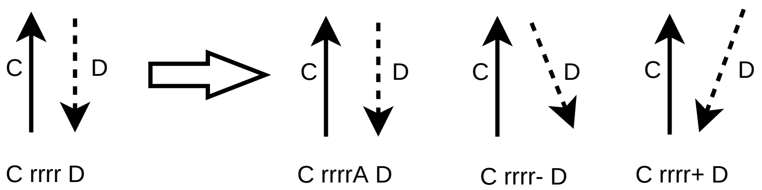


Figure 3.3: *Dipole relations. left: a $\mathcal{DRA}_{\text{c}}$ relation, right: the corresponding relations in $\mathcal{DRA}_{\text{fp}}$.*

A directed line segment with start point s and end point e is what Moratz [Mor00] calls a *dipole*. Dipoles are used to represent spatial objects with intrinsic orientations [Dyl05]. The relationship of two dipoles in two-dimensional continuous space, $\mathbb{R}^2$, is given by the quadruple of relationships of each point of each dipole to the other dipole. The relationship of a point to a dipole can be *left of* (l), *right of* (r), or *on the straight line through* that dipole (o). Yet the latter relation (o) is neglected and only the relations left (l) and right (r) are of further interest. The relationship of the dipoles C (s_C, e_C) and D (s_D, e_D) in figure 3.3 to the left side of the arrow is $C\ rrrr\ D$. This is the abbreviation of the case where s_D and e_D are to the right of C, and s_C and e_C are to the right of D ($C\ r\ s_D \wedge C\ r\ e_D \wedge D\ r\ s_C \wedge D\ r\ e_C$).

Using this notation, 14 different dipole relations are possible which are similar to the 14 line segment relations with end points at general positions developed by Schlieder [Sch95]. The reference frame for the dipole relation algebra, $\mathcal{DRA}_{24}$, by Moratz, Renz and Wolter [Mor00] is shown in figure 3.4a). It contains ten further relations that capture the cases where two dipoles share a common point. The start point of one dipole can be equivalent to the other dipole's start point (s), or to the other dipoles end point (e). This means that s_D can be equivalent to s_C or to e_C. Also respectively s_C can be equivalent to s_D or to e_D.

Taken together, the dipole relation algebra consists of 24 jointly exhaustive and

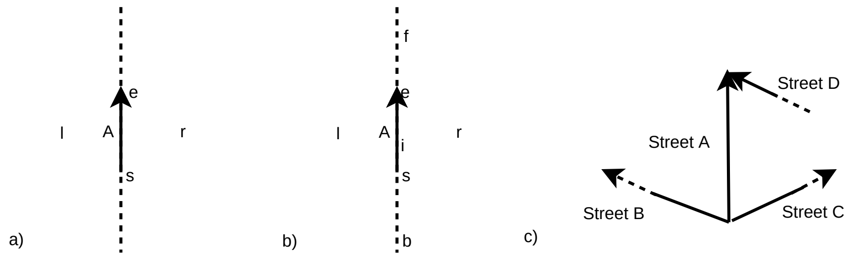


Figure 3.4: *Frame of reference for a) DRA_{24} (DRA_c) b) DRA_f. c) Spatial navigation example [Mor00].*

pairwise disjoint (JEPD) basic relations, a set which is referred to as D_{24}, where DRA_{24} refers to the powerset of $\mathcal{D}_{24}$, which contains 2^{24} possible unions of basic relations. In [Dyl05] $\mathcal{DRA}_{24}$ is called $\mathcal{DRA}_c$ due to the coarse distinction between different orientations within this algebra. $\mathcal{DRA}_{24}$ forms a relation algebra that allows applying constraint-based reasoning techniques.

A further development of the dipole relation algebra adds the relations *b*, where a point is straight behind the dipole; *i*, where a point is in the interior of the dipole, and *f*, where a point is straight in front of the dipole. This reference frame is shown in figure 3.4b). Together with these new relations, a total of 72 basic relations is obtained which also contains Allen's 13 one-dimensional relations as special cases. This dipole relation algebra was originally called $\mathcal{DRA}_{69}$ in [Mor00] and later referred to as fine grained dipole relation algebra ($\mathcal{DRA}_f$) for instance in [Dyl05].

The application domain for DRA_f is spatial navigation. One example, given by [Mor00], is a car navigating its way through the network of one-way streets shown in figure 3.4c). The car starts from the intersection point of the three streets A, B and C and attempts to reach a goal on *Street D*. Due to one-way traffic, it cannot turn into *Street D* directly from *Street A* and has to make the decision whether to turn to *Street B* or *Street C* in the hope that the chosen one will lead to *Street D*. The initial knowledge is expressed as $A\{slsr\}B$, $A\{srsl\}C$, $A\{rele\}D$. The questions are if $B\{ells, errs\}D$ or $C\{elles, errs\}D$ hold. This leads to two sets of constraints; Θ_1 contains the initial knowledge and $B\{ells, errs\}D$, whereas Θ_2 contains the initial knowledge and $C\{elles, errs\}D$. The path-consistency method [Mac77] is used and leads to the result, that only Θ_2 is path-consistent whereas Θ_1 contains a contradiction. That means, that *Street B* cannot turn into *Street D* whereas there is the possibility that *Street C* might turn into *Street D*.

To facilitate application for robot navigation, Dylla and Moratz [Dyl05] extended $\mathcal{DRA}_f$ to $\mathcal{DRA}_{fp}$, which contains information about the angle between the two dipoles. The four relations P (parallelism), A (antiparallelism), + and - for a positive respectively negative angle between the dipoles are added. This leads to refinements of several relations of $\mathcal{DRA}_f$. One example is given in figure 3.3, which shows the $\mathcal{DRA}_f$ relation *rrrr* on the left side of the arrow and its refined relations, *rrrA*, *rrr-*, and *rrr+* in $\mathcal{DRA}_{fp}$ to the right.

For robot navigation the conceptual neighbourhood graph between two dipoles is used. One dipole represents an object, and the other represents the robot. When the robot moves, its relation to the object changes only between conceptually neighboring states.

3.4 Bipartite Arrangements

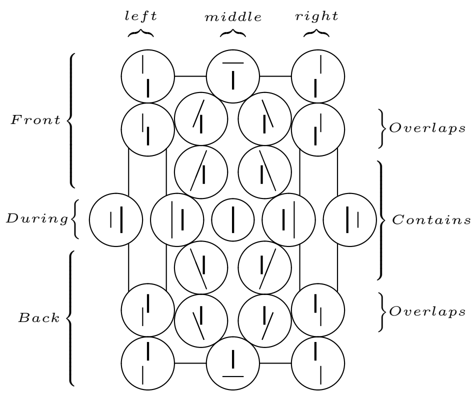


Figure 3.5: *Pictorial representation of the 23 Bipartite Arrangements [Got04].*

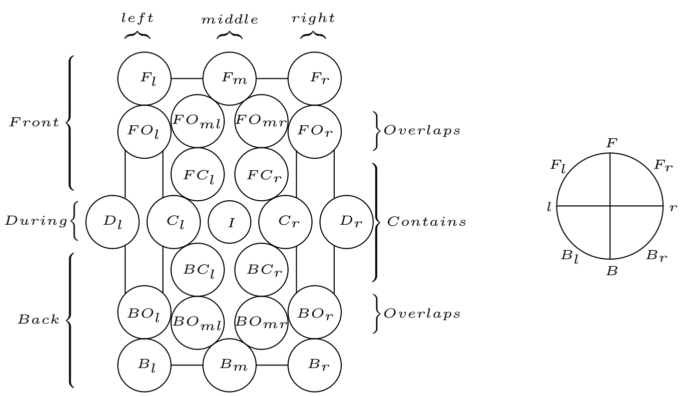


Figure 3.6: *Bipartite Arrangements. Left: Abbreviations for the 23 Bipartite Arrangements. Right: The eight possible orientations for each relation [Got04].*

Gottfried [Got04] provides a set of qualitative interval relations in the plane, the so-called Bipartite Arrangements ($\mathcal{BA}$). From originally 225 possible Bipartite Arrangements, he chooses 23 JEPD relations that are sufficient for a coarse representation of rigid objects that do not naturally intersect. The relation x_y describes the relative position of the primary interval y to the reference interval x. Furthermore, the orientation of the primary interval can be taken into account. Gottfried distinguishes eight different orientations. 125 interval relations are obtained, which he refers to as $\mathcal{BA}_{23}^{8}$. These 23 basic relations are pictorially presented in figure 3.5 and their mnemonic description is given in figure 3.6 to the left. To the right in figure 3.6, the eight possible orientations of the primary interval are given: $0°$ (F), $]0°; 90°[$ (F_l), $90°$ (r), $]90°; 180°[$ (F_r), $180°$ (B), $]180°; 270°[$ (B_l), $270°$ (l), $]270°; 360°[$ (F_l). Note that for instance the relation/orientation combinations for the arrangements D_l and C_r equal each other if the consideration of primary and secondary interval is lost. The same is the case for D_r and C_l, F_l and B_r, as well as B_l and F_r which leads in total to 125 relations.

3.5 Qualitative Triangulation

Schlieder's line segment relations [Sch95] mentioned above are based on triangle orientations of the four possible triangles that can be established between the end points of two line segments. If just one triangle (ABC) is considered, its orientation is clockwise (-) if point C lies to the right of the straight line through A and B, and counterclockwise $(+)$ if C lies to its left. If the coordinates of A and B are given, and both angles α and β are known, C's coordinates and its distance from the straight line between A and B can be determined using *triangulation*. Triangulation, illustrated in figure 3.7, is a geometric technique used for many purposes within metrology, astrometry, binocular vision, gun direction of weapons, surveillance, and navigation.

Qualitative Triangulation [Lig93], shown in figure 3.8, is the qualitative version thereof, where the angles α and ν (with $\nu = 180° - \beta$) are not quantitatively measured but qualitatively estimated and assigned to a certain class of angles whose members are not differentiated. In the case where only angles of 45° can be distinguished, these sixteen angle classes are to be used: 0°,]0°;45°[, 45°,]45°;90°[, 90°,]90°;135°[, 135°,]135°;180°[, 180°,]180°;225°[, 225°,]225°;270°[, 270°,]270°;315°[, 315°,]315°;360°[. An angle of 183° then falls into the interval $]180°; 225°[$.

Therefore, the result for C's position will be a region instead of an exact point, as in the quantitative counterpart. When the observer at point A estimates α to point C as between 0° and 45°, and the observer at point B estimates ν as between 90° and 135°, point C can be anywhere in the intersection of these two regions shown in figure 3.8. The accuracy of the resulting position description of C depends on the estimation skills of the two observers. The more angle classes

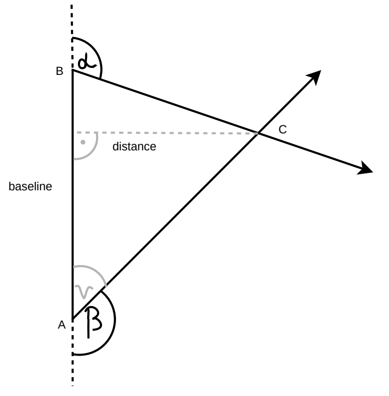


Figure 3.7: *Triangulation.*

that are distinguished, the more accurate the result becomes. In any case, both observers are using the same classification. Qualitative triangulation provides a family of calculi where each calculus is specified by its specific set of angle classes [Lig93].

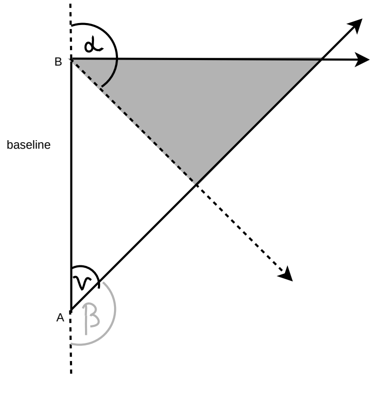
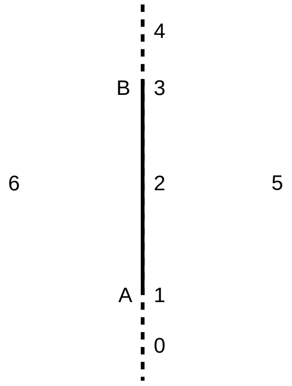


Figure 3.8: *Qualitative Triangulation. Angle ν at point A is estimated to be between $0°$ and $45°$ and angle α at point B is estimated between $90°$ and $135°$. Point C must be in the intersection region of this two angle classes.*

3.6 Flip-Flop Calculus

One of the qualitative triangulation calculi is the Flip-Flop Calculus by Ligozat [Lig93] that uses four angle classes. It is assumed that the two observers are only able to distinguish between something being *straight front* ($0°$), *to the right*



alpha	nu	name of region
180	180	0
180	undefined	1
180	0	2
undefined	0	3
0	0	4
<180	<180	5
>180	>180	6

Figure 3.9: *The Flip-Flop Calculus [Lig93].*

($]0°; 180°[$), *straight behind* (180°), or *to the left* ($]180°; 360°[$). From all combinations of angles for α and ν, seven different regions for the position of C can be distinguished, as illustrated in figure 3.9. Region *0* is on the straight line behind point A, region *1* is at point A, and region *2* is on the straight line between A and B. Region *3* is at point B, region *4* is on the straight line in front of B, region *5* is to the right of the straight line through A and B and region *6* is to the left of that line. The region that C lies in can be looked up in the composition table shown in figure 3.9 for any combination of angles.

LR Calculus

The *LR* Calculus [Sci04] refines the Flip-Flop Calculus by adding two more relations *equal* and *e12*. *equal* describes the situation in that all three points A, B, and C are at the same location and the relation *e12* stands for the case where only A and B are at the same location. The resulting *LR* Calculus, therefore, has nine different relations.

3.7 Single and Double Cross Calculi

A neighbourhood-oriented representation is used in Freksa's Single and Double Cross Calculi for oriented objects, which are both used for reasoning about spatial direction information [Fre92b; Zim96]. The overall motivation for the Single Cross Calculus can be thought of as follows: Given two points, a and b, we would like to know where a third point, c, is in relation to a directed line from a to b. Therefore, a front back dichotomy is established in point b by introducing an orthogonal line with regard to the vector ab through point b, see figure 3.10a).

This approach yields the eight qualitative regions *straight front* (*0*), *right front* (*1*), *right neutral* (*2*), *right back* (*3*), *straight back* (*4*), *left back* (*5*), *left neutral*

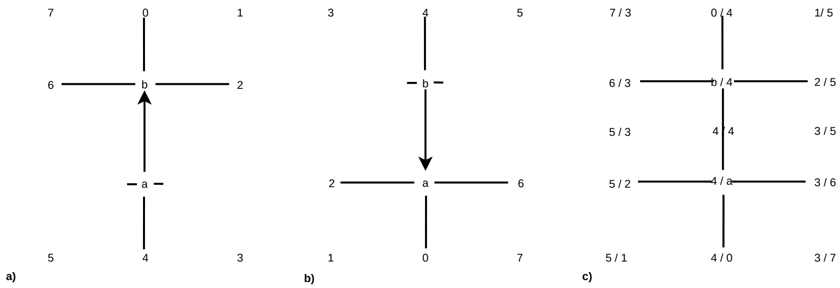


Figure 3.10: *Frames of reference for Single and Double Cross Calculi. a) Single Cross established at point b b) Single Cross established at point a c) Double Cross Calculus [Fre92b].*

(*6*) and *left front* (*7*) as shown in figure 3.10a). Note that the regions *straight front*, *right neutral*, *straight back* and *left neutral* are one-dimensional, whereas the regions *right front*, *right back*, *left front* and *left back* are two-dimensional. In figure 3.10b) the front back dichotomy is established in point a and the position of point c can be given in relation to the directed line from b to a. Combining the two single cross reference frames of figure 3.10a) and 3.10b), the Double Cross Calculus with 15 different qualitative regions, shown in figure 3.10c), is obtained. Six regions are areas, seven are lines, and two are points. Freksa [Fre91b] points out that only distinguishable features count for qualitative reasoning and that the type differences of the regions have no importance for it. The double cross presented here can also be seen as a special case of a qualitative triangulation calculus, which uses the eight angle classes $0°$, $]0°;90°[$, $90°$, $]90°;180°[$, $180°$, $]180°;270°[$, $270°$, and $]270°;360°[$ [Lig93].

A typical reasoning task for the Double Cross Calculus is given in figure 3.11a) where it is known that c is in region *1/5* of vector ab and d lies in region *3/5* of vector bc. The question is what d's position is in relation to vector ab. As no distance information is included in the model and therefore nothing is known about the length of ab and bc, no definite answer for the position of d in relation to ab can be given. The relation can either be *1/5*, *2/5*, *3/5*, *3/6* or *3/7*.

Reasoning in the Double Cross Calculus is typically done by composition tables where the entries are the position of c in relation to vector ab, here shown in figure 3.11b) and the position of d in relation to vector bc, shown in figure 3.11c). The entries in the intersection, given in figure 3.11d) are then all possible positions of d in relation to ab. For the complete composition table of the Double Cross Calculus, see [Fre92b]. Further examples of reasoning and inferences can be found in [Fre92b] and [Zim96].

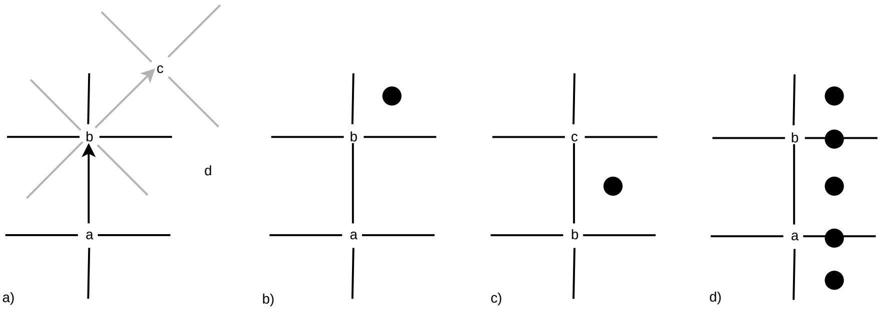
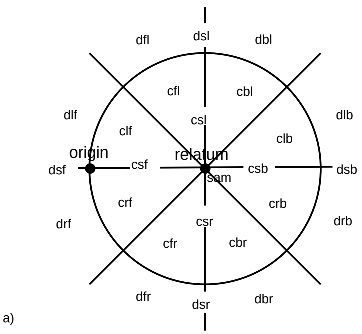


Figure 3.11: *Reasoning within the Double Cross Calculus. a) c is right front of the vector ab and d is right neutral of the vector bc. The question is where is d in relations to the vector ab. b) Pictorial presentation of c's qualitative position to vector ab. c) Pictorial presentation of d's position to vector bc. d) Pictorial presentation of the inferred relations that d can have to vector ab.*

3.8 Ternary Point Configuration Calculus



csf	close straight front	clb	close left back
dsf	distant straight front	dlb	distant left back
crf	close right front	cbl	close back left
drf	distant right front	dbl	distant back left
cfr	close front right	csl	close straight left
dfr	distant front right	dsl	distant straight left
csr	close straight right	cfl	close front left
dsr	distant straight right	dfl	distant front left
cbr	close back right	clf	close left front
dbr	distant back right	dlf	distant left front
crb	close right back	dou	origin = relatum ≠ referent
drb	distant right back	tri	origin = relatum = refernt
csb	close staight back	sam	origin ≠ relatum = referent
dsb	distant straight back		

b)

Figure 3.12: *TPCC. a) The TPCC reference frame [Mor03a]. b) Abbreviations for the qualitative regions in the TPCC reference frame.*

The Ternary Point Configuration Calculus (TPCC) [Mor03a] has been developed for robot navigation. It can be used to integrate an agent's local and survey knowledge about a spatial environment. The calculus is derived from the Single Cross Calculus. By adding another and by 45° rotated cross at the relatum finer distinctions can be made where all planar regions are divided into two parts. Furthermore, a circle whose radius is the distance between origin and relatum is added around the relatum. Regions inside the circle are *close* regions whereas regions outside the circle are *distant* regions. One further difference to the single and double cross frame of reference is that the relatum is facing the origin whereas in the other two calculi, the relatum is facing the same direction as the origin. The development of the reference frame, shown in figure 3.12a), is based on psycholinguistic research results on reference frames [Ten02; Mor03b]. Thus, the TPCC consists of 27 atomic relations that are listed in figure 3.12b).

A possible application is that an autonomous agent derives a map from the information it gathers by its sensors while exploring an environment given as a route graph. The agent follows the route sequences and makes observations at certain time points. Its task is to distinguish between the landmarks that it reaches during its exploration and to know when two landmarks are identical. Due to its separation of close and distant regions, the TPCC allows the agent to distinguish between landmarks even if more than just orientation knowledge is needed for that distinction. For a complete example and details on the inference process see [Mor03a].

3.9 Rectangle Algebra

The rectangle algebra [Bal98; Güs89; Muk90] is based on Allen's Interval Calculus [All83] and represents axis parallel rectangles in two-dimensional Euclidian space. The rectangle's projections on each of the axes are intervals as shown in figure 3.13. Each dimension is regarded separately and the relationship between two rectangles is given as a tuple of the relationships of each dimension. 13 x 13 pairs of atomic relations are available. The relationship of rectangle A to rectangle B in figure 3.13 is therefore

$$\mathbf{A/B} = \begin{pmatrix} x_{before} \\ y_{overlaps} \end{pmatrix}$$

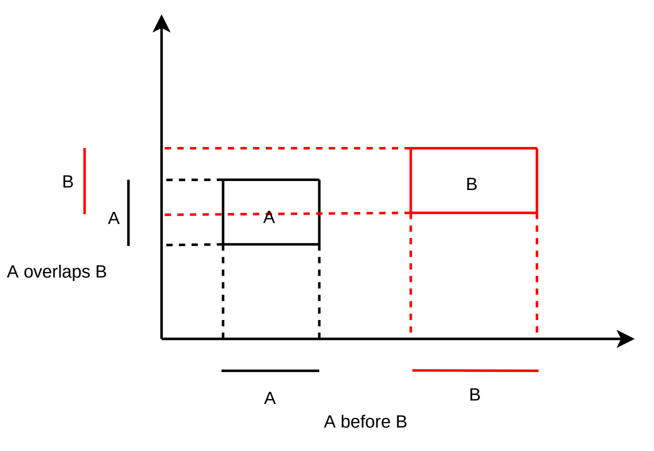


Figure 3.13: *Rectangle algebra.*

Like the previously presented calculi, the rectangle algebra is designed to infer new object relations from given ones. If, for instance, the relationships A/B and B/C are given, the relationship A/C can be read from a composition table, where the relationship for each dimension is inferred separately and the results are combined to the relationship tuple.

3.10 Relations of Minimum Bounding Rectangles

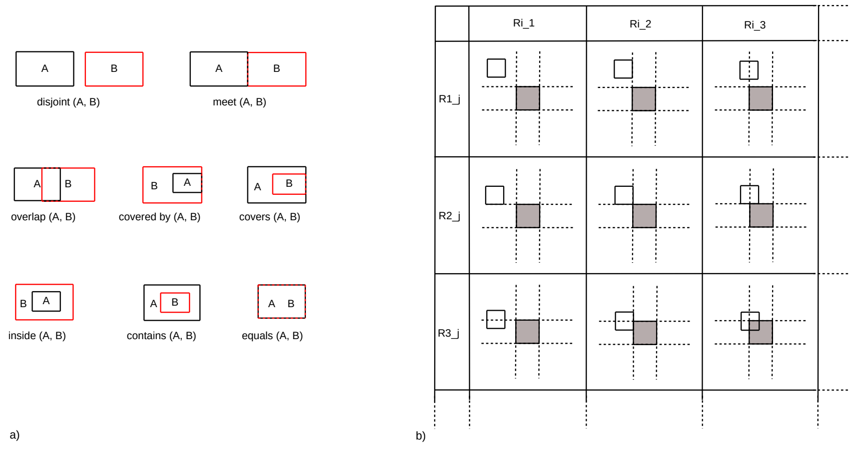


Figure 3.14: *Relations of MBRs. a) Rectangle Relations b) MBR Matrix [Pap95].*

Papadias et al. [Pap95] describe how topological information can be retrieved from Minimum Bounding Rectangle (MBR) relations. MBRs are often used to approximate a topological geographic object for instance in a geographic information system (GIS), where a topological object O is included in its minimum bounding rectangle O'.

The relationships between two rectangles are based on Egenhofer's 9-intersection model [Ege91]. A rectangle has an interior, a boundary, and an exterior. The 9-intersection model covers 512 relations. However, only eight pairwise disjoint and jointly exhaustive relations *disjoint*, *meet*, *equal*, *overlap*, *contains*, *inside*, *covers*, and *covered by*, shown in figure 3.14a), are sufficient to describe the relationships of two rectangles.

The projection of any two rectangles onto an axes of the coordinate system provides two intervals that can have one out of Allen's [All83] relationships. Projected onto both axis 13 x 13 = 169 relations are available. R_{i_j} denotes the relationship between two rectangles, where i and j indicate the corresponding interval relationship on the x an y axes respectively. The relationships for $1 \leq i \leq 3$ and $1 \leq j \leq 3$ are depicted in figure 3.14b).

This approach is used as an initial filtering step for retrieving topological information from a GIS to a given query. If the query is to find all objects *A* that are *equal* to *B*, all MBRs *A'* that are *equal* to the MBR *B'* have to be found, because only these MBRs can hold objects that might be *equal* to *B* but also could *overlap* *B*, are *covered-by* *B*, *cover* *B*, or *meet* *B*. Additional filter steps are necessary to separate the actual *equal* objects. The MBRs are *equal* in 2-D space when their projected interval relationships onto the x-axis and y-axis are both *equal* in 1-D space.

3.11 Mukerjee and Joe's approach

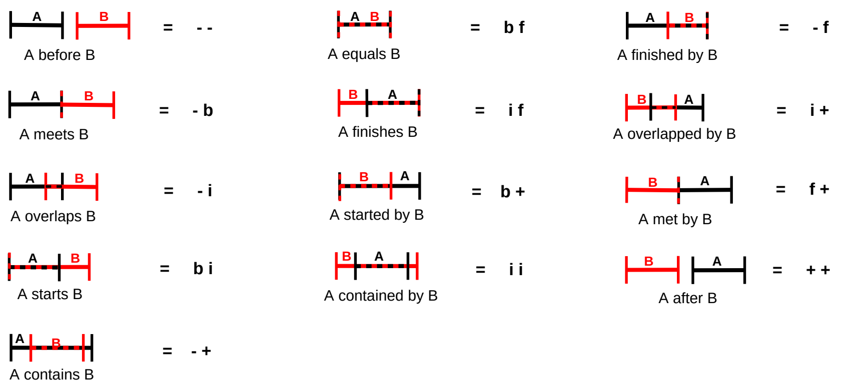


Figure 3.15: *Mukerjee and Joe's relations in each dimension matches Allen's interval relations.*

Mukerjee and Joe [Muk90] propose a model for qualitative relations between objects in one-, two-, or three-dimensional space. For one dimension, they base their object relations on Allen's Interval Calculus [All83], but redefine the interval relations by specifying the relations of the two end points of one interval to the second interval. An end point can be in one of the relationships: *ahead* (+), *front* (f), *interior* (i), *back* (b) or *posterior* (-) to the other interval. Figure 3.15 shows how this description matches Allen's relations.

For two- or three-dimensional space, Mukerjee and Joe handle objects whose enclosing boxes or cuboids are aligned with the orthogonal reference system, with an approach similar to rectangle algebra [Bal98; Güs89]. The objects are projected onto the x and y (respectively x, y, and z) axes and their relationship is given as the tuple of interval relationships in all dimensions.

For objects at arbitrary angles in two-dimensional space, an enclosing box around a two-dimensional object is used. The object must have an identified front, so that the plane around the box can be divided by extending the boxes' lines into eight two-dimensional regions, shown in figure 3.16a). The regions are named *1* to *8*. Only the *lines of travel* are used, which are the lines in the forward and backward direction, shown in figure 3.16b).

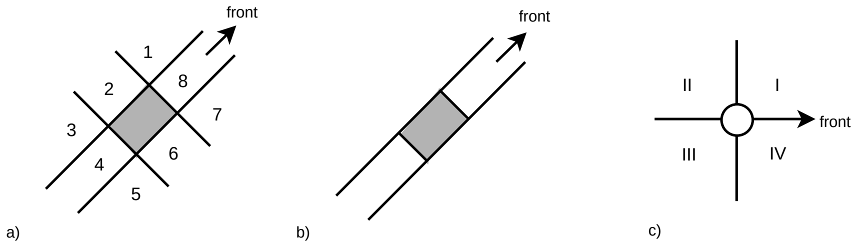


Figure 3.16: *Mukerjee and Joe's approach. a) The eight regions around the reference object. b) the lines of travel. c) Four quadrants that are obtained by establishing a left/right and a front/back dichotomy in a point.*

In order to reason about direction information, the object is seen as a point. The directions *left/right* and *front/back* form four quadrants around the object. Figure 3.16c) illustrates that quadrant *I* lies between the directions *front* and *left*, quadrant *II* between *left* and *back*, quadrant *III* between *back* and *right* and quadrant *IV* between *right* and *front*. Another object's direction is given in terms of which quadrant of the reference object it faces. In the example in figure 3.17a) object B is facing A's first quadrant whereas A is facing B's fourth quadrant.

Where two objects' lines of travel intersect, they form a *collision parallelogram* (*CP*). While the objects move, both of them will eventually pass the *CP* or already have passed it. An object, as well as an interval, has two end lines: its front line and its back line. To describe an object's position its end lines' relationships to the *CP* are regarded. The relationship between an end line and the *CP* can be one out of the five relationships described above and illustrated in figure 3.17b)

To fully describe the relationship of two objects we need the quadrant information, $dir(A/B)$, the location of object A with respect to B, $pos(A/B)$, and the location of object B with respect to A, $pos(B/A)$. The example constellation of figure 3.17a) is described by $dir(A/B) = IV$ which means that A is facing object B's fourth quadrant, $pos(A/B) =$ - -, which means that both end lines of B have not reached the collision parallelogram yet, and $pos(B/A) =$ - - which means that both end lines of A have not reached the collision parallelogram either.

Mukerjee and Joe developed their approach without any particular application in mind and are mostly concerned with precisely the spatial relations [Muk90]. However, given two relationships that have one object in common, the relationship between the other two objects can be composed. If the relationships between (A,B) and (B,C) are known, the relationship (A,C) is inferable. In figure 3.17c), the relationship (A,B) is given by $pos(A/B) =$ - i (A is before the *CP* with B), $pos(B/A) = + +$ (B is after the *CP* with A) and $dir(B/A) = I$ and the

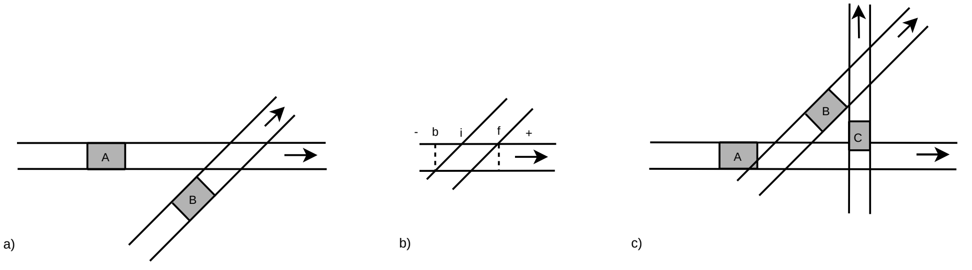


Figure 3.17: *Reasoning in Mukerjee and Joe's approach. a) The collision parallelogram. b) The five different relations that an endline of the object can have to the collision parallelogram. c) A reasoning situation with three objects.*

relationship (B,C) by $pos(B/C)$ = - b (B meets the CP with C), $pos(C/B)$ = - - (C is before the CP with B) and $dir(B/C) = IV$. The inferences that can be drawn for the relationship (A,C) are: $pos(A/C)$ = - -, $dir(A/C) = III$ *or* IV. However, the relationship $pos(C/A)$ (which in this example is A is before the CP with C), is not inferable. Also in this approach composition tables are used to infer the composite information. For more detail on the approach see Mukerjee and Joe [Muk90].

3.12 AC Calculus

Ragni [Rag03] presents the AC calculus for spatial reasoning on a grid structure. The grid $G_n(\mathbb{N}^2) \in \mathbb{N}^2$ is sized n x n. Spatial objects are given as ordered pairs (x, y) denoting an x-y-coordinate in the grid. An object is represented in the grid by obtaining a grid cell between the four coordinates (x, y), (x+1, y), (x, y+1), (x+1, y+1). The AC Calculus uses thirteen binary position relations to the given coordinate (x, y). The relation of object z_1 at position (x_1, y_1) to object z_2 at position (x_2, y_2) is defined as:

$$
\begin{array}{ll}
\mathrm{EQ} z_1 z_2 := (x_1 = x_2) \wedge (y_1 = y_2) & \text{(Equal)} \\
\mathrm{AR} z_1 z_2 := (x_1 + 1 = x_2) \wedge (y_1 = y_2) & \text{(AdjacentRight)} \\
\mathrm{DR} z_1 z_2 := (x_1 + 1 < x_2) \wedge (y_1 = y_2) & \text{(DistantRight)} \\
\mathrm{AL} z_1 z_2 := (x_1 = x_2 + 1) \wedge (y_1 = y_2) & \text{(AdjacentLeft)} \\
\mathrm{DL} z_1 z_2 := (x_1 > x_2 + 1) \wedge (y_1 = y_2) & \text{(DistantLeft)} \\
\mathrm{AF} z_1 z_2 := (x_1 = x_2) \wedge (y_1 + 1 = y_2) & \text{(AdjacentFront)} \\
\mathrm{DF} z_1 z_2 := (x_1 = x_2) \wedge (y_1 + 1 < y_2) & \text{(DistantFront)} \\
\mathrm{AB} z_1 z_2 := (x_1 = x_2) \wedge (y_1 = y_2 + 1) & \text{(AdjacentBehind)} \\
\mathrm{DB} z_1 z_2 := (x_1 = x_2) \wedge (y_1 > y_2 + 1) & \text{(DistantBehind)} \\
\mathrm{FR} z_1 z_2 := (x_1 < x_2) \wedge (y_1 < y_2) & \text{(FrontRight)} \\
\mathrm{FL} z_1 z_2 := (x_1 > x_2) \wedge (y_1 < y_2) & \text{(FrontLeft)} \\
\mathrm{BR} z_1 z_2 := (x_1 < x_2) \wedge (y_1 > y_2) & \text{(BehindRight)} \\
\mathrm{BL} z_1 z_2 := (x_1 > x_2) \wedge (y_1 > y_2) & \text{(BehindLeft)}
\end{array}
$$

The AC calculus has two levels of granularity. For example, an object to the right can directly neighbour the reference object, which is given by the relation AR (AdjacentRight), or lie further away to its right, given by the relation DR (DistantRight). AC is a representation algebra, which is a finite concrete relational algebra without the compose operator. For more details see [Rag03].

The motivation behind AC is to analyze psychological experimental data about human spatial reasoning from the computational perspective. It is assumed that people use a set of formal rules which they apply to the information given at the premisses while establishing a mental representation of the described scenario and while concurrently reasoning about objects' positions within the described object configuration. The problems to be modelled are given by a set of premisses followed by a question such as:

The hammer is to the right of the pliers.
The screwdriver is to the left of the hammer.
The wrench is in front of the screwdriver.
The saw is in front of the hammer.
What is the relationship of the saw and the wrench?

ACSAT

Ragni [Rag03] defines the constraint satisfaction problem ACSAT, which is to decide consistency of a described spatial configuration. The description consist of the set θ of spatial formulas. Ragni states that the problem of satisfiability of an ACSAT problem θ with n spatial variables can be answered in a grid of size $(2n)^2$ but is NP-complete. However, he provides a complete, nondeterministic polynomial time algorithm for the smaller problem ACSAT for Base Relations. The algorithm presented in [Rag03] delivers the coordinates of the objects in $G_n(\mathbb{N}^2)$.

3.13 Star Calculi

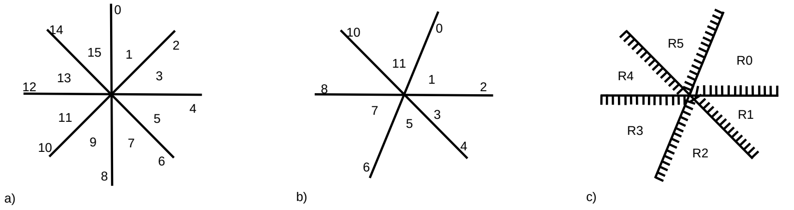


Figure 3.18: *Star Calculi. a)* $\mathcal{STAR}_4(0)$ *b)* $\mathcal{STAR}_3[23, 90, 135](23)$ *c)* $\mathcal{STAR}^r_3$*[23, 90, 135](23).*

Star Calculi [Ren04] are a class of qualitative direction calculi for point objects. They can be specified with arbitrary fixed granularity. A Star Calculus of the class $\mathcal{STAR}_m$ is established at a point, p, it uses a global reference direction, $\mathcal{P}$ for instance north, and consists of m lines that intersect at p. Thereby, the plane around p is partitioned into $4m$+1 star relations, whereof $2m$ are half lines, $2m$ are two-dimensional sectors and one is the point p itself. The zones are numbered clockwise from 0, to $4m$, starting with the first half line clockwise from the reference direction $\mathcal{P}$. The relations {1,3, 5,...,4m-1} are called odd relations and the relations {2,4,6,..., 4m} are called even relations.

The angles between the intersecting lines may vary, and the reference direction $\mathcal{P}$ does not have to be one of the relations. Figure 3.18a) shows the reference frame of $\mathcal{STAR}_4(0)$, with 4 intersection lines at equal angles. The first relation (*0*) is identical to the reference direction (differing from the reference direction by 0°). Figure 3.18b) shows the reference frame of $\mathcal{STAR}_3$[23, 90, 135](23), with three intersecting lines at the angles 23°, 90°, 135° in clockwise direction from $\mathcal{P}$. The first relation (*0*) differs by 23° from the reference direction.

A Star Calculus $\mathcal{A}$ over the $4m + 1$ basic relations ($bas(\mathcal{A})$) consists of 2^{4m+1} different relations. The consistency problem CSPSAT($bas(\mathcal{A})$) can be solved with quantitative methods but is not decidable with only qualitative reasoning methods [Ren04]. In order to change this Renz, and Mitra [Ren04] revised the Star Calculus to decide consistency of a set of constraints over $bas(\mathcal{A})$ by the path-consistency method [Mac77].

Revised Star Calculus

In the revised Star Calculus, the one-dimensional relations are removed so that each of the 4m lines is subsumed by its preceding two-dimensional relation in clockwise order. Figure 3.18c) shows the revised Star Calculus $\mathcal{STAR}^r_3$[23, 90, 135](23). The consistency for a set of constraints, Θ over $bas(\mathcal{A})$, for a revised Star Calculus $\mathcal{A} \in \mathcal{STAR}^r_m$, with m $\leq$ 3, can be decided by the path consistency method [Ren04].

A suggested application areas for Star Calculi and revised Star Calculi is location of positions in relation to the direction of given landmarks, for instance reasoning about positions relative to a cell phone transmitter. The transmission cells of different angle sizes can be represented exactly by the Star Calculus. Another application area is direction description for navigation tasks. A direction can be described at the finest granularity at which the user or user's navigation tool is capable [Ren04].

3.14 Oriented Point Relation Algebra

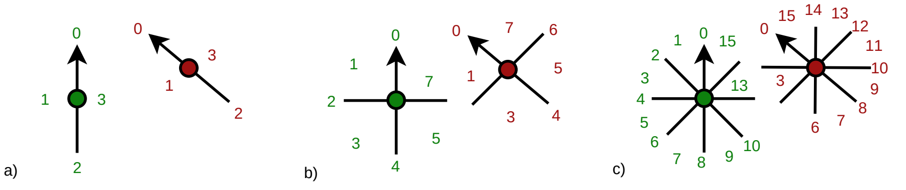


Figure 3.19: $\mathcal{OPRA}_m$. *a)* $\mathcal{OPRA}_1$ *b)* $\mathcal{OPRA}_2$ *c)* $\mathcal{OPRA}_4$

$\mathcal{OPRA}_m$ [Mor05; Mor06; Dyl06] is designed to describe the relative orientation of two oriented points, so called $\mathcal{O}$-points. An $\mathcal{O}$-point is a point pair and a direction in the 2D-plane, which can also be described as a dipole [Mor00; Sch95] with infinitely small length [Mor06]. An $\mathcal{O}$-point is therefore the simplest spatial entity with an intrinsic orientation [Mor05]. It is suited to model objects with intrinsic fronts or intrinsic direction of movement that are abstracted to points.

$\mathcal{OPRA}_m$ with granularity parameter m $\in N$ has scalable granularity that determines the number of relations distinguished. An angular reference frame with the resolution $\frac{2\pi}{4m}$ is attached to each of the two $\mathcal{O}$-points. The 2m planar and 2m linear regions are numbered 0 to (4m-1), where region 0 always coincides with the $\mathcal{O}$-point's orientation. The qualitative spatial relation, rel_{OPRA_m}, between two $\mathcal{O}$-points, A and B, is given as the pair of relations (rel_i, rel_j) where i is the region of A in which B falls and j the region of B in which A falls, usually written as $A\ m\angle_i^j\ B$ [Dyl06].

$\mathcal{OPRA}_1$ induces for instance, as shown in figure 3.19a), the qualitative regions *left* (1) and *right* (3) which are planar, and the regions *front* (0) and *back* (2) which are linear together with a fifth relation *same* which denotes the position of the $\mathcal{O}$-point itself. In $\mathcal{OPRA}_1$ the two objects in figure 3.19a) have the relation $A\ 1\angle_3^1\ B$, whereas in $\mathcal{OPRA}_2$, shown in figure 3.19b), where two lines intersect at an $\mathcal{O}$-point which results in four planar and four linear qualitative regions, the relation can be given more precisely as $A\ 2\angle_7^1\ B$. Using $\mathcal{OPRA}_4$ with four intersecting lines, at the $\mathcal{O}$-points, resulting in 16 regions, shown in figure 3.19c), the relation between the objects will be $A\ 4\angle_{13}^3\ B$.

The motivation for a scalable qualitative calculus was given from cognitive robotics where fine distinctions are useful in many tasks but the calculus nevertheless provides provably minimal composition tables. Using $\mathcal{OPRA}_m$ the granularity can be adjusted to the robot's perception capabilities during task execution. The highest possible resolution is set by the maximal possible resolution of the vision system. However, the maximum resolution might not be needed in simple tasks, for instance when to decide if something is left or right. A lower resolution can be used to save computation time [Mor05].

OPSAT

$\mathcal{OPRA}_1$ results in 20 jointly exhaustive and pairwise disjoint atomic relations that build the set $\mathcal{OP}_1$. $\mathcal{OPRA}_1$ refers to the 2^{20} possible unions of atomic relations, the powerset of $\mathcal{OP}_1$. Constraint-based reasoning techniques should be used for reasoning in $\mathcal{OPRA}_1$. A typical constraint satisfaction problem is OPSAT. Given Θ, a set of constraints in the form xRy, with x, y as variables and R is an $\mathcal{OPRA}_1$ relation, it needs to be decided whether Θ is consistent. OPSAT can be solved using standard CSP methods [Mor05].

3.15 Direction Relation Matrices

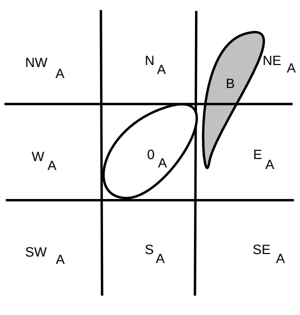


Figure 3.20: *Direction Relation Matrix.*

Direction Relation Matrices [Goy00a; Goy00b; Goy01] are used to retrieve geographical information about objects from a geographical information system (GIS). The matrices use a global frame of reference where the relation of the target object to the reference object is given in cardinal directions. A user's query to a GIS could for instance be: Find all lakes northeast of a city. Any city in the GIS' database is taken as a reference object and thereafter, if there are any lakes in the region northeast of the object is checked. The GIS will try to find all city/lake pairs that match the description. Furthermore, it might grade them (for instance by considering additional distance information) and display lakes that lie close to cities and exactly to the northeast of them first, and lakes that are further away and only vaguely to the northeast later.

The reference frame for the *Coarse Direction Relation Matrix* [Goy00a], shown in figure 3.20, partitions the space around the reference object into nine mutually exclusive regions, the *direction tiles*. The boundaries between the tiles have no extent and the union of the nine tiles forms a complete partition of the space. The tile in the middle is the minimum bounding box around the reference object, A, and is referred to as *same*, whereas each of the remaining tiles represents one of the eight cardinal directions *north*, *northeast*, *east*, *southeast*, *south*, *southwest*, *west*, *northwest*.

$$\mathrm{dir}_{RR}(A,B) := \begin{bmatrix} NW_A \cap B & N_A \cap B & NE_A \cap B \\ W_A \cap B & 0_A \cap B & E_A \cap B \\ SW_A \cap B & S_A \cap B & SE_A \cap B \end{bmatrix} \quad (3.1)$$

$$\mathrm{dir}_{RR}(A,B) := \begin{bmatrix} \emptyset & \emptyset & \neg\emptyset \\ \emptyset & \emptyset & \neg\emptyset \\ \emptyset & \emptyset & \emptyset \end{bmatrix} \quad (3.2)$$

The Direction Relation Matrix (equation (3.1)) that gives the cardinal direction of the target object, B, consists of 3x3 entries, each of which stands for the intersection of B with one of the nine regions. $2^9 = 512$ distinct configurations are theoretically possible to describe with the nine elements in the Direction Relation Matrix, but not all configurations are possible direction relations. For instance, the matrix with only empty values would describe a configuration with no target object and is therefore excluded. Furthermore, all configurations where the target object would have to be disconnected are excluded. The geographical configuration depicted in figure 3.20 leads to the Direction Relation Matrix shown in equation (3.2).

More detailed Direction Relation Matrices have been developed to capture object relations more precisely. The *Detailed Direction Relation Matrix* also considers the arial distribution of the target object, that is, how big a percentage of it falls into a certain tile of the matrix. When a tile contains more than one component of the target object, it is meant to compute the area of each component.

To answer a user's GIS query, the Coarse Direction Relation Matrix could be applied as a first filter for the retrieved information and the Detailed Direction Relation Matrix could be used afterwards to prioritize certain candidates provided by the coarse Direction Relation Matrix. Goyal [Goy00a] discusses a 5x5 Direction Relation Matrix in order to take not only the nine partitions but also all 16 boundaries into account. However, he considers this approach to be cognitively overwhelming due to its 25 different relations, and does not explore this solution further.

Skiadopoulos and Koubarakis provide a consistency-based composition algorithm for cardinal direction relations based on the relations used in the Direction

Relation Matrix [Ski04] and they introduce an algorithm for consistency checking of a set of cardinal constraints [Ski05].

The matrices considered so far have been only applicable for region objects. It can however be important to also handle point and line objects as the representation of an object can change across different scales within the GIS. The *Deep Direction Relation Matrix* [Goy00a] is designed for any combination of point, line, and region reference and target object and takes the appearance of the target object within a tile and on the boundaries between the tiles into account. For further details on the different Direction Relation Matrices see [Goy00a].

This chapter presented besides Allen's Interval Calculus [All83] fourteen qualitative spatial reasoning calculi developed for different tasks in two-dimensional space. The next chapter analyzes how applicable these calculi are for describing an object configuration from several local viewpoints and for reconstructing it into a global frame of reference.

Chapter 4

Applicabilities of the described calculi for map-like object configuration reconstruction

The qualitative spatial calculi introduced in chapter 3 are all designed to reason about spatial relationships between objects, mostly in the way that given relationship B/A and relationship C/B, relationship C/A can be inferred. However, any of these calculi can be used to describe an object configuration. In order to be applicable for the situation described in chapter 1, four requirements must be fulfilled.

- The observer on site might not have any knowledge of the underlying global frame of reference and should therefore be allowed to describe the object positions in relation to each other using only their intrinsic frames of reference.
- In order to alleviate the configuration description for the observer and the reconstruction for the listener only binary relation descriptions are of interest.
- The nine positional relation classes need to be distinguished and shall be called: *STRAIGHT FRONT* (*SF*), *RIGHT FRONT* (*RF*), *RIGHT NEUTRAL* (*RN*), *RIGHT BACK* (*RB*), *STRAIGHT BACK* (*SB*), *LEFT BACK* (*LB*), *LEFT NEUTRAL* (*LN*) and *LEFT FRONT* (*LF*), and *SAME* (*S*). Figure 4.1 shows the frames of references that divide the plane exactly into these nine position classes. It is not necessary that a calculus uses this exact classification. However, it is necessary that the members of different classes can be distinguished within the calculus.

- In addition eight object orientations that will accordingly be called *straight front* (*sf*), *right front* (*rf*), *right neutral* (*rn*), *right back* (*rb*), *straight back* (*sb*), *left back* (*lb*), *left neutral* (*ln*), and *left front* (*lf*) have to be expressible. This means that a calculus must be able to distinguish between 72 position-orientation combinations independently of the object representation used.

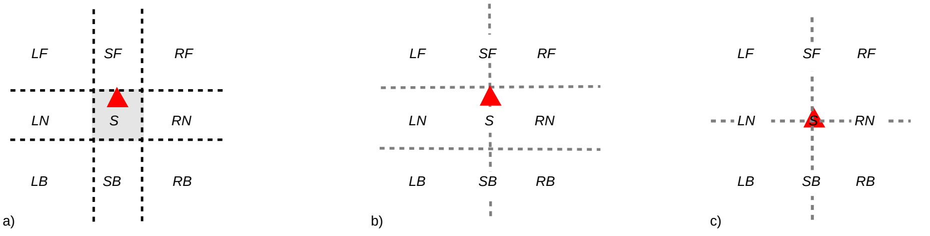


Figure 4.1: *Frames of reference using nine position classes. a) rectangle representation b) line representation c) point representation.*

Not all presented calculi use an intrinsic frame of reference or binary object relation description in their original version. However, any of them can be deployed after small adjustments. An absolute frame of reference can be changed into an intrinsic frame of reference, and a ternary point calculus can be changed into a binary point calculus for the special case where origin and relatum are at the same position. Alternatively, the objects can be represented as lines with two end points that can be interpreted as origin and relatum. The following sections show how the calculi introduced in chapter 3 can be applied for the description and reconstruction of an object configuration into a global frame of reference.

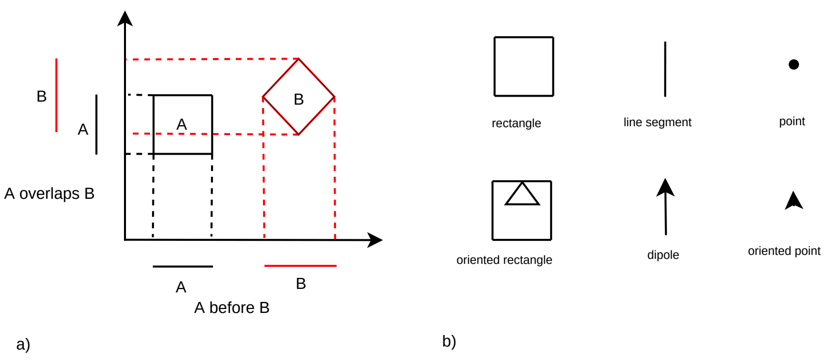


Figure 4.2: *Object representation. a) Rectangle projections onto the axis. b) Oriented object representations.*

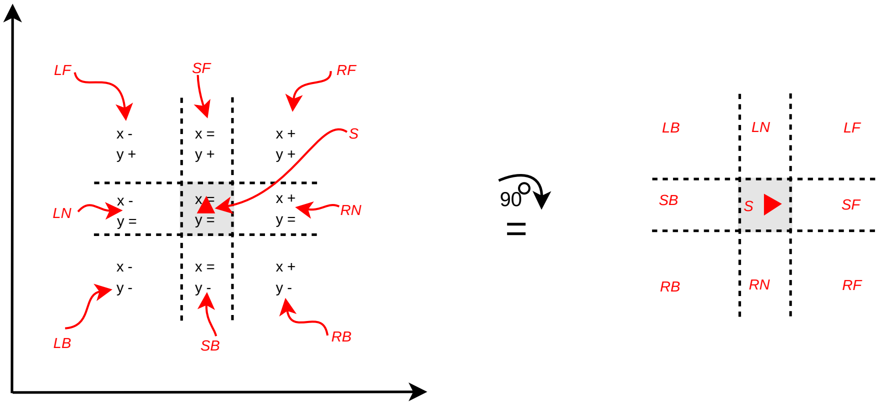


Figure 4.3: *The change of an absolute frame of reference into an intrinsic frame of reference.*

4.1 Rectangle representations

Any object can be represented as a rectangle by their minimum bounding box, either aligned with the coordinate system or at any free angle. An object's orientation can be attached to the rectangle by an arrow pointing into its facing direction. Objects without known orientation can be represented just by its minimum bounding boxes. Rectangle based representations include Minimum Bounding Rectangle Relations [Pap95], Rectangle Algebra [Bal98; Güs89; Muk90], the Direction Relation Matrix [Goy00a; Goy00b; Goy01] and the part of the Mukerjee and Joe's approach that deals with aligned objects [Muk90]. These use an absolute projection based frame of reference with neutral zone, and are of particular interest for the task at hand as opposed to calculi that use a projection based frame of reference without neutral zone. The neutral zone specifies the position of the reference object. The space around the reference object is divided into eight two-dimensional areas that each represent object positions for objects with a specific alignment to the reference object.

Within the projection-based frame of reference without neutral zone, parts of the reference object itself might overlap with the areas and an intersection between the reference object and a target object cannot be expressed.

As mentioned before, an absolute frame of reference can be changed into an intrinsic one, which is demonstrated in figure 4.3. In the example, the object faces increasing y-dimension. The areas given by the absolute frame of reference are renamed according to the object's orientation. They are then permanently attached to the object. Whenever the object turns, the frame of reference turns accordingly.

Except for Mukerjee and Joe's approach, none of the calculi presented here takes rectangles at angles into account. However, as long as only the object's position in relation to the reference object is considered, the object can have any shape and orientation. The relations are based on a tuple of interval relations that are given by the object's projections onto the coordinate system's axes. As shown in figure 4.2a) this technique works equally well for objects at angles.

Relations of Minimum Bounding Rectangles

The 169 Minimum Bounding Rectangle Relations [Pap95] are too fine-grained to be cognitively easy to use, but can be grouped into nine relation classes that correspond to the relations we want to distinguish. Figure 5.6 in chapter 5 shows which relations are part of the relation classes. The others are combined relations that can all be expressed by a combination of the chosen relation classes.

Rectangle Algebra

The rectangle relations [Bal98; Güs89; Muk90] are tuples of interval relations from the rectangle's projections onto the coordinate system's axes. This leads to the same 169 rectangle relations as for *MBR*s. The rectangle relations can be grouped into nine relation classes, shown in figure 4.4. Each describing one of the nine relations needed.

$$\begin{pmatrix} x_{equal \vee starts \vee inside \vee finishes} \\ y_{after \vee metby} \end{pmatrix} \quad \begin{pmatrix} x_{metby \vee after} \\ y_{metby \vee after} \end{pmatrix}$$

$$\begin{pmatrix} x_{metby \vee after} \\ y_{equal \vee stars \vee inside \vee finishes} \end{pmatrix} \quad \begin{pmatrix} x_{metby \vee after} \\ y_{before \vee meets} \end{pmatrix}$$

$$\begin{pmatrix} x_{equal \vee stars \vee inside \vee finishes} \\ y_{before \vee meets} \end{pmatrix} \quad \begin{pmatrix} x_{before \vee meets} \\ y_{before \vee meets} \end{pmatrix}$$

$$\begin{pmatrix} x_{before \vee meets} \\ y_{equal \vee starts \vee inside \vee finishes} \end{pmatrix} \quad \begin{pmatrix} x_{before \vee meets} \\ y_{metby \vee after} \end{pmatrix}$$

$$\begin{pmatrix} x_{equal \vee starts \vee inside \vee finishes} \\ y_{equal \vee starts \vee inside \vee finishes} \end{pmatrix}$$

Figure 4.4: *The nine relation classes for rectangle relations.*

Direction Relation Matrix

The Direction Relation Matrix [Goy00a; Goy00b; Goy01] considers nine basic relations. When its absolute frame of reference is changed into an intrinsic frame of reference, the nine relations coincide exactly with the nine relations needed in our task. Designed for the application in geographic information systems, reasoning processes developed for the Direction Relation Matrix are aimed for recognition of described object configurations available from the GIS' database. The reconstruction of of object configurations has not been considered.

AC Calculus

From all calculi, presented in this thesis, the AC calculus [Rag03] is the only one that clearly considers reconstruction of a whole object configuration. The reconstruction takes place in a grid $G_n(\mathbb{N}^2)$ of rectangular cells. A cell could be interpreted as the object's minimum bounding box. Therefore the AC representation can be regarded as a rectangle representation. Thirteen binary relations that include the nine relations needed are used. If the relations AR and DR are combined to the relation *RIGHT NEUTRAL*, AL and DL to the relation *LEFT NEUTRAL*, AF and DF to the relation *STRAIGHT FRONT*, and AB and DB to the relation *STRAIGHT BACK*, the nine relations remain. However, the calculus does not consider objects with intrinsic orientations. The objects' relationships are seen from the observer's perspective, which is a survey perspective. Object orientations at at angles to the grid are not possible to represent.

All four approaches naturally succeed to distinguish objects at the nine positions and the target object's own intrinsic orientation can be distinguished between four orientations that differ from each other by an angle of at least 90°. For more orientations, all three rectangle representations become ambiguous. Using finer MBR- or Rectangle Algebra relations does not overcome this problem. These relations only consider the objects' alignment but not their orientations. In the AC calculus, objects with orientations at angles to the grid can not be represented. Therefore, AC is restricted to four object orientations each of which is aligned to one of the coordinate axis. Mukerjee and Joe [Muk90] follow a different approach. In order to be able to distinguish objects' orientations they take the objects' relations to the intersecting part of the objects' regions into account.

Mukerjee and Joe's approach

For objects aligned with the coordinate system, the approach of Mukerjee and Joe [Muk90] uses the same representation as the *MBR* relations and the Rectangle Algebra and it provides the same expressibility and shortcomings. Nine positional relations and four orientations are distinguishable. For objects at arbitrary angles a second approach, as described in section 3.11, is applied. Here the objects have an arbitrary orientation and the relation between them is given with respect to their collision parallelogram (CP). This approach distinguishes between eight orientations but fails to describe the objects' positional relationships unambiguously for which reason the configurations in figure 4.5a), 4.5b) and 4.5c) that should be distinguished to object B being in a) LB, in b) LN, and in c) LF of object A, are described the same.

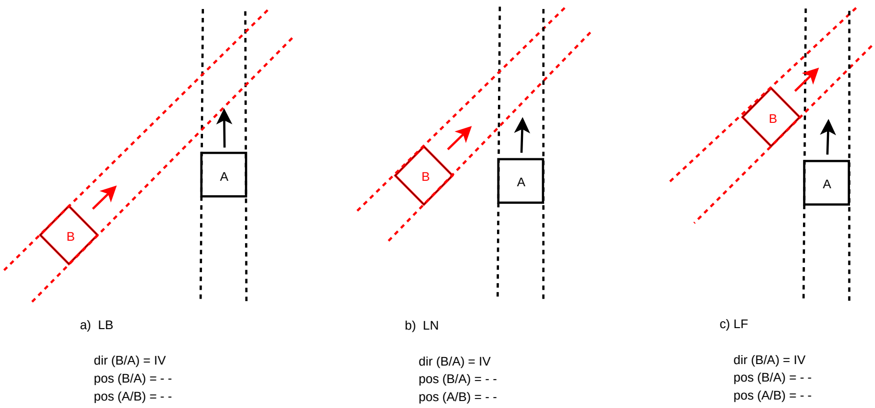


Figure 4.5: *Different positional relations that cannot be distinguished in Mukerjee and Joe's approach.*

A combination of the two approaches does not solve the problem. Even if the positional relation, dir(B/A), pos(B/A), pos(A/B) and dir(A/B) is given, some position orientation combinations are still not possible to be described unambiguously. For instance, both configurations in figure 4.6 are described with: $B/A = (\overset{-}{-}\overset{-}{-})$, $dir(B/A) = II$, $pos(B/A) = ++$, $pos(A/B) = ++$, and $dir(A/B) = III$.

4.2 Line representations

An object representation such as a line or line segment given by an ordered tuple of points [Sch95], or dipole, given by an arrow [Mor00], automatically includes the object's orientation, which makes it impossible to represent orientationless objects. An oriented line does not represent an object homogeneously. The object's extension in the front/back direction is represented by the length of the line, whereas its extension in the left/right direction is not represented. The consequence is that some relations distinguishable for objects represented as rectangles are not distinguishable in a line representation. Figure 4.7a) shows the line relations that are unambiguous representations of the objects' positions. Each red line represents two objects facing opposite directions. For the relations SF, SB, and S, only objects with the directions sf or sb can be represented unambiguously.

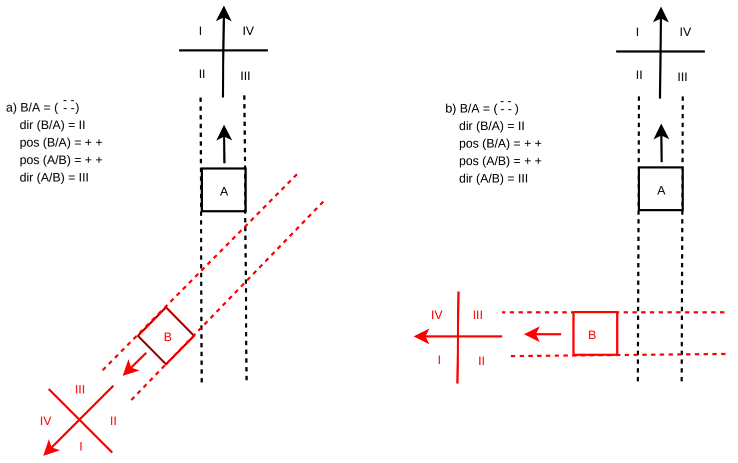


Figure 4.6: *Two different relations that still are described the same in the combined version of Mukerjee and Joe's approach.*

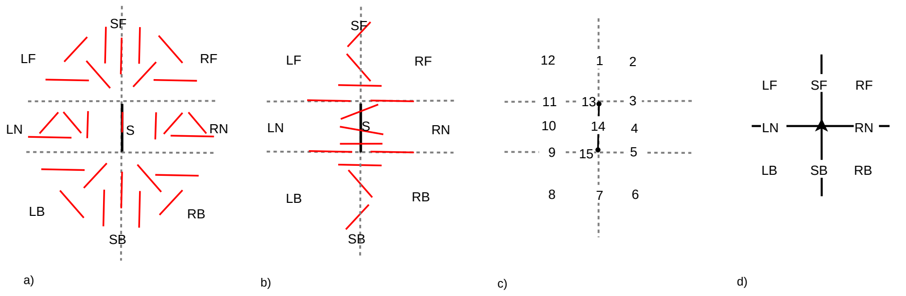


Figure 4.7: *Position relations. a), b), and c) in a line representation. d) in a frame of reference for oriented point objects.*

In figure 4.7b) it is shown, that the objects that are *SF*, *SB*, or *S* of the reference object and have the orientation *rf*, *rb*, *lf*, *lb*, *rn*, or *ln*, can not be distinguished from objects having the same orientations but intersect the three regions *RF*, *SF*, and *LF*, and respectively, *RB*, *SB*, and *LB* or *RN*, *S*, and *LN*. One further difference is that the lines separating the regions *RF* and *LF*, and respectively *RB* and *LB*, or *RN* and *LN* represent the basic relations *SF* and respectively *SB*, or *S* whereas the lines separating the regions *RF* and *RN*, *RN* and *RB*, *LF* and *LN*, *LN* and *LB* do not represent basic relations. Consequently, objects that intersect with two of these neighbouring regions and have the orientation *rn* or *ln* are represented on the line between the regions. Conversely, the objects that are represented on the lines between *RF* and *LF*, and respectively, between *RB* and *LB*, or *RN* and *LN* do not intersect the two regions. A different approach is used for Bipartite Arrangements [Got04]. Objects are represented as lines that only show where the object is in relation to the reference object. In addition, a second representation provides the object's orientation. Thus Bipartite Arrangements allow representation of objects without orientation.

Line Segment Relations

Line Segment Relations [Sch95] are formulated by the orientation of the four triangles that can be established between the four end points (A, B, C, D) that define the two line segments AB and CD. If the triangle ABC is ordered clockwise, C is to the right of the straight line through A and B. If the triangle is ordered anticlockwise, C is to the left of that line. If all four points are collinear, an ordering of the pairs of points on the line is given.

An end point that is not on the straight line through the other line segment can be either to the left or to the right of that line segment. The more fine-grained relations *RF*, *RN*, and *RB*, and respectively *LF*, *LN*, and *LB* cannot be distinguished. For instance, the relationships *RF*, *RN* and *RB* are all coded by $([ABC]\ [ABD]\ [ACD]\ [BCD]) = (--++) = 0011_{bin} = 3_{dec}$ and the positions *LF*, *LN* and *LB* as $([ABC]\ [ABD]\ [ACD]\ [BCD]) = (++--) = 1100_{bin} = 12_{dec}$. Furthermore, the orientations *lf*, *sf*, and *rf*, and respectively *lb*, *sb*, and *rb* are not distinguishable from each other.

Dipole Relation Algebra

The relationship of two dipoles is given by a quadruple of relationships of each point of each dipole to the other dipole. DRA_c [Dyl05] only distinguishes between the positions *RIGHT*, *LEFT*, *SF*, *SB* and *S*. The more fine-grained relations *RF*, *RN*, and *RB*, and respectively *LF*, *LN*, and *LB* are not distinguished. At the positions *RIGHT*, *LEFT* and *S*, the four orientations *sf*, *rn*, *sb*, and *ln* are distinguished, whereas at the positions *SF* and *SB*, only the orientations *right* and *left* are available.

DRA_f adds 48 basic relations to the 24 that DRA_c provides, which leads to the result that the positions $SF \wedge RF$, $SF \wedge LF$, $SB \wedge RB$, and $SB \wedge LB$ can also be distinguished. Furthermore, at the positions *SF* and *SB*, the objects' orientations *sf* and *sb* can be described. Unfortunately, DRA_f suffers from the same problem as DRA_c and the Line Segment Relations. The more fine-grained relations *RF*, *RN*, and *RB* as well as *LF*, *LN*, and *LB* cannot be distinguished. The three relations to the right are all coded as *ArrllB* whereas the three relations to the left are coded *AllrrB*. Furthermore, only four orientations can be distinguished. DRA_{fp} adds some orientation information that allows the differentiation of eight orientations for all available positions. However, the relations *RF*, *RN*, and *RB* as well as *LF*, *LN*, and *LB* are still not distinguishable.

Bipartite Arrangements

Bipartite Arrangements [Got04] use a local frame of reference and represent the objects as line segments. The difference to Dipole Relation Algebra and Line Segment Relations is that the representation of the target object's orientation is detached from the representation of its position. Furthermore, the frame of reference, shown in figure 4.7c), distinguishes the positions F_l, F_m, F_r, B_l, B_m, B_r, D_l, D_r, and I. These relations identically match the needed relations *RF*, *RN*, *RB*, *LF*, *LN*, and *LB* as well as *SF*, *SB* and *S*. Fourteen of the 23 basic relations distinguished by Gottfried represent configurations where the target object overlaps with several reference regions. The remaining nine relations are the basic relations that we consider necessary to distinguish.

The target object's orientation is given by the circle next to the line that represents the object. The circle describes exactly eight different orientations that coincide with the desired orientations: *sf*, *rf*, *rn*, *rb*, *sb*, *lb*, *ln*, and *lf* and therefore leads to a distinctive description of all cases that need to be differentiated. Due to the detached orientation information, the line that represents the object can overlap all regions that the real world object is at least partly inside. Furthermore, objects without orientation can be represented.

4.3 Ternary point representations

The abstraction of an object to a point does not preserve its orientation. In this case, the orientation information has to be given separately. An alternative is used by $\mathcal{OPRA}_m$ where objects are *oriented points*, which can be regarded as arrows with infinitely small length [Mor06].

Flip-Flop- [Lig93], $\mathcal{LR}$- [Sci04] , Single- and Double Cross Calculus [Fre92b; Zim96], and TPCC [Mor03a], are ternary point calculi. A typical position description of a target object in a ternary point calculus would be formulated as: 'The church (target object) is to the right of the town hall (relatum), seen from the post office (origin).' In our scenario, origin and relatum coincide, as the observer is located at the reference object. This can be seen as a special case of the ternary point calculus, where the line from origin to the relatum becomes infinitely small and legitimizes the use of oriented points. Another solution that makes it possible to use a ternary point calculus, even though only two different positions are available, is to represent the reference object as two points. In this case one point represents its back (origin) and the other its centroid or its front (relatum). This representation is similar to a line representation.

Flip-Flop- and $\mathcal{LR}$ Calculus

The Flip-Flop Calculus [Lig93] does not take the special case into account when origin and relatum lie at the same position. Its refinement, the $\mathcal{LR}$ Calculus, includes the relation *equal*, where all three points are at the same position, and the relation *e12*, where origin and relatum are equal. However, the position for the target object in *e12* will not be further distinguished other than being *DIFFERENT* from origin and relatum.

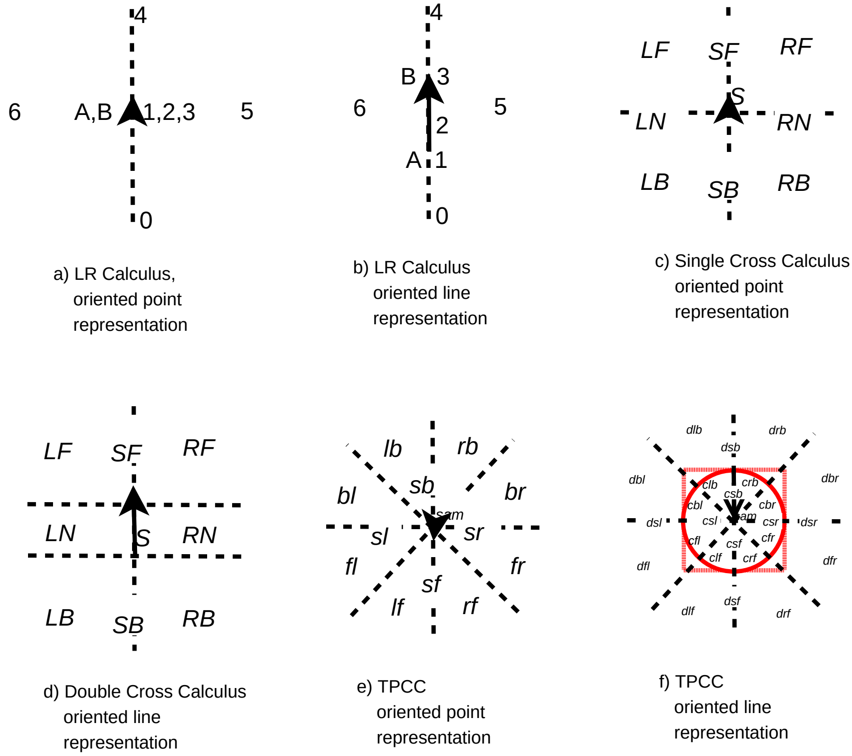


Figure 4.8: *Different approaches to change the frame of reference of ternary point calculi to binary point calculi and line representations for the special case where origin and relatum coincide. a) and b) LR Calculus, c) Single Cross Calculus d) Double Cross Calculus e) and f) TPCC.*

The $\mathcal{LR}$ Calculus [Sci04] can be modified, in such a way that the point representing origin and relatum is an oriented point. Figure 4.8a) shows that after this modification, the five $\mathcal{LR}$ relations equal *SF* (4), *SB* (0), *S* (A, B, 1, 2, 3), *LEFT* (6), and *RIGHT* (5). However, the relations *RF*, *RN* and *RB* as well as *LF*, *LN* and *LB*, are not available.

At the positions *SF* and *SB*, Flip-Flop and $\mathcal{LR}$ Calculus distinguish between the four orientations *sf*, *rn*, *sb*, and *ln*, but at the positions *RIGHT* and *LEFT* only between the two orientations *right* and *left*. Representing the object as a line from its back to its front as shown in figure 4.8b), instead of an oriented point, leads to the same expressibility regarding orientations. The difference with the previous approach is a finer granularity with regard to the position *S*, which is not of interest.

Single Cross Calculus

When origin and relatum coincide in the Single Cross Calculus [Fre92b; Zim96], and the object is represented as an oriented point, the frame of reference becomes the one shown in figure 4.8c) which equals the frame of reference that we consider suitable for point objects. The nine positional relations can clearly be differentiated but for the target object's intrinsic orientation only at the positions *SF*, *SB*, *LN*, and *RN* eight orientations are distinguishable. At the positions *RF*, *LF*, *RB*, and *LB* only four different orientations can be described. The same expressibility is gained by using the alternative where the objects are represented as line segments.

Double Cross Calculus

When origin and relatum coincide, and the objects are represented as oriented points, the Double Cross Calculus [Fre92b; Zim96] becomes equivalent to the Single Cross Calculus. In the second alternative, when a line describes the object from its back to its front, the Double Cross Calculus, shown in figure 4.8d), allows for the distinction between all nine regions and becomes the suitable reference frame for line objects. The frame of reference is similar to the one used in Bipartite Arrangements but the representation of objects includes the objects' orientations. Even in this approach, it is only possible to distinguish between four intrinsic orientations of the target object at the positions *RF*, *LF*, *RB*, and *LB*.

Ternary Point Configuration Calculus

The TPCC [Mor03a] is derived from the Single Cross Calculus but makes finer distinctions. A further difference is that the relatum is facing the origin whereas in the Single Cross Calculus, origin and relatum are facing the same direction. The relation names in TPCC use the relatum's front/back orientation and the origin's left/right orientation.

When origin and relatum coincide, the close regions that are defined by a circle around the relatum, with the distance between origin and relatum as radius, disappear. The resulting frame of reference is shown in figure 4.8e). Thirteen regions remain, eight of which are areas, four are lines, and the position of the relatum itself. In TPCC, not only the relations *RF*, *RN*, and *RB* as well as *LF*, *LN* and *LB* are distinguished, but also *RF*, *RB*, *LF*, and *LB* are each divided into two separate regions. This allows for the distinction of eight different intrinsic orientations for the target object. Figure 4.8f) shows the reference frame where the reference object (the red square) is represented as a line from its back to its centroid. The close regions are kept (inside the red circle), but depending on the reference object's shape, it can be assumed that at least parts of the close regions are occupied by the object itself.

4.4 Binary point representations

In a binary point calculus, the target object's relation to the reference object is described by using just the two points, relatum and target. To use a binary point calculus the relatum must either have an intrinsic orientation as is the case in $\mathcal{OPRA}_m$ [Mor05; Mor06; Dyl06], or a global frame of reference must be used, something which is true for Star Calculi [Ren04].

Star Calculi

Star Calculi [Ren04] is a group of binary point calculi with arbitrary granularity, originally using a global frame of reference. Star Calculi can be either projection-based or cone-based, depending on the offset that the first relation has to the reference direction. Star Calculi with no offset provide a projection-based frame of reference. To apply Star Calculi for the given task, the global frame of reference is changed into an intrinsic one by applying the object's intrinsic orientation as its reference direction. $\mathcal{STAR}_1(0)$ consists of one line through the object that splits the plane around it into the five regions *SF*, *SB*, *LEFT*, *RIGHT* and *S*. This approach equals the previously described special case of the $\mathcal{LR}$ Calculus where origin and relatum coincide and the objects are represented by oriented points.

In $\mathcal{STAR}_2(0)$ the nine regions *SF*, *RF*, *RN*, *RB*, *SB*, *LB*, *LN*, *LF* and *S* are distinguished. This results in the same frame of reference as Single- and Double Cross Calculus, with coinciding origin and relatum and an object representation as oriented points. $\mathcal{STAR}_4(0)$ uses seventeen position relations and is able to distinguish between eight orientations that differ by at least 45°. $\mathcal{STAR}_4(0)$ is therefore equivalent to the special case of the TPCC. Further Star Calculi provide an even finer categorization of positions and orientations. $\mathcal{STAR}_8(0)$ for instance, is able to distinguish orientations with a resolution of $22,5°$ and $\mathcal{STAR}_{16}(0)$ with a resolution of $11,25°$.

Oriented Point Relation Algebra

A relation in $\mathcal{OPRA}_m$ [Mor05; Mor06; Dyl06] is a relation tuple where each object's position is given in relation to the other object. This automatically captures the objects' relative orientations. The frame of reference is intrinsic and the objects are represented as oriented points. Assumed that the relationships of two objects are always given in pairs (A/B and B/A) using calculi different from $\mathcal{OPRA}_m$, $\mathcal{OPRA}_1$ and $\mathcal{OPRA}_2$ are similar to the previously described versions of $\mathcal{STAR}_1(0)$ and $\mathcal{STAR}_2(0)$. $\mathcal{OPRA}_4$'s expressibility equals the modified version of $\mathcal{STAR}_4(0)$. The 16x16 relation tuples describe seventeen different positions, and eight different orientations. It is not necessary to use the relation tuple just to describe the target object's position in relation to the reference object. This can be described with only one piece of information. It is therefore possible to represent objects with unknown orientation and add orientation information later, when it becomes available.

4.5 Evaluation

Four rectangle representations, one grid representation, three line segment representations, four ternary point calculi and two binary point calculi have been analyzed. Several general criteria, listed in the beginning of this chapter, must be fulfilled by a spatial calculus to be appropriate for the overall purpose when describing an object configuration from a local perspective and reconstructing it into a global frame of reference.

The advantage of a rectangle representation and the grid representation is their inherent representation of three-dimensional objects two-dimensionally. The representation is symmetrical; all represented dimensions are equally abstracted. It allows also for overlapping relations, such as an object being in more than one basic relation to the reference object at the same time. In addition to this, objects without orientation can be represented. All four rectangle representations and the grid representation are able to distinguish the nine positions that are considered necessary but none of them is able to represent more than four orientations unambiguously.

Lines represent objects asymmetrically. An object's length is abstracted by the line's length, but its width is neglected with the consequence that not all overlapping relations can be represented unambiguously. Line Segment Relations and Dipole Relation Algebra use the orientation of the line to represent the object's orientation and are therefore not able to represent objects with unknown orientation. Neither calculus is able to distinguish between nine positions nor between eight orientations. Bipartite Arrangements represent an object's orientation separately from the line that represents the object at its relative position. This allows for objects with unknown orientation. Furthermore, nine positions and eight orientations can be differentiated and the calculus is thus able to represent overlapping relations.

Using a ternary point calculus seems to be unsuitable in a situation where only two objects are available. However, this situation can be seen as the special case where origin and relatum coincide. Two approaches were discussed. In the first the objects were represented as oriented points whereas in the second the objects were represented as lines. In both alternatives, $\mathcal{LR}$- and Single Cross Calculus could not distinguish between the three different relations *RF*, *RN*, and *RB*, and respectively, *LF*, *LN*, and *LB*. Double Cross Calculus could express the relations *RF* and *LF* but could not distinguish between *RN* and *RB*, and respectively, *LN* and *LB*. In both alternative representations, TPCC was the only calculus that could distinguish between the nine necessary positions. It distinguishes between thirteen position relations (26 in the case where close and distant positions are distinguished). The relations *RF*, *RB*, *LF*, and *LB* are divided into two subregions, which make it possible to distinguish between the eight orientations.

A binary point calculus seems to be more appropriate to represent the reference and the target object. Both $\mathcal{STAR}$ Calculi and $\mathcal{OPRA}_m$ represent a group of calculi with scalable granularity. $\mathcal{STAR}_4(0)$ and $\mathcal{OPRA}_4$ distinguish between seventeen position relations that include the nine relations required as such or as union of three relations each. The fine-grained positional distinction makes it possible with regard to both calculi to distinguish between the eight orientations. Not regarding the close and distant regions in TPCC, the $\mathcal{STAR}_4(0)$, $\mathcal{OPRA}_4$ and TPCC representation equal each other except for region naming. Hereafter, only $\mathcal{OPRA}_4$ will be referred to because it provides this functionality naturally by way of design. TPCC had to be modified for the special case where origin and relatum coincide and $\mathcal{STAR}_4(0)$ originally uses a global frame of reference.

Tables 4.1 and 4.2 provide an overview of the different aspects that have been analyzed for all calculi (except the AC calculus) within this chapter. To summarize, when objects are abstracted to lines, Bipartite Arrangements are a good choice and for point objects it is appropriate to use $\mathcal{OPRA}_4$. Unfortunately, no rectangle representation has been found that covers the requirements. As rectangle representations are regarded to be natural and appropriate for objects that are relatively close to each other, it would be an advantage to have a rectangle representation available that covers the requirements. Therefore, the representation scheme QuaDRO, introduced in chapter 6 has been developed. QuaDRO represents objects as rectangles and covers some, but not all, important aspects that the previously discussed rectangle approaches did not cover. To the author's knowledge, no comparable system designed for description and reconstruction of object configurations does exist. Before QuaDRO is introduced, chapter 5 presents the psychological background and technical requirements for this representation scheme.

		Frame of reference in the original	Frame of reference in modification	Distinct relations in original	Required positions that can be expressed when only the intrinsic frame of reference and other modifications are applied	Orientation classes that can be distinguished when the intrinsic reference frame and other modifications are applied	
Rectangle Representations							
MBR Relations		absolute	intrinsic	169	LF SF RF LN S RN LB SB RB	SF/SB/LN/RN RF/LF/RB/LB S	8 4 -
Rectangle Algebra		absolute	intrinsic	169	LF SF RF LN S RN LB SB RB	SF/SB/LN/RN RF/LF/RB/LB S	8 4 -
Qualitative Model for Space	aligned objects	absolute	intrinsic	169	LF SF RF LN S RN LB SB RB	SF/SB/LN/RN RF/LF/RB/LB S	8 4 -
	objects at arbitrary angles	intrinsic	intrinsic	676	LEFT SF S SB RIGHT	4	
	combined	-	intrinsic		LF SF RF LN S RN LB SB RB	SF/SB/LN/RN RF/LF/RB/LB S	8 7 8
Direction Relation Matrix		absolute	intrinsic	9	LF SF RF LN S RN LB SB RB	SF/SB/LN/RN RF/LF/RB/LB S	8 4 -
Line Representations							
Line Segment Relations		intrinsic pair relations	-	63	LEFT SF S SB RIGHT	4	
Dipole Relation Algebra		intrinsic pair relations	-	DRA_c 24 DRA_f 72 DRA_{fp} 80	LEFT SF S SB RIGHT	DRA_c: left/right 4; SF/S/SB 2 DRA_f: 4 DRA_{fp}: left/right 8; SF/S/SB 4	
Bipartite Arrangements		intrinsic	-	23	LF SF RF LN S RN LB SB RB	8	

Table 4.1: *Summary of the discussed aspects for rectangle- and line representations.*

	Frame of reference in the original	Frame of reference in modification		Distinct relations in original	Required positions that can be expressed when only the intrinsic frame of reference and other modifications are applied		Orientation classes that can be distinguished when the intrinsic reference frame and other modifications are applied			
Ternary Point Representations										
		coinciding points	line representation		coinciding points	line representation	coinciding points		line representation	
Flip-Flop Calculus *LR* Calculus	relative	relative/intrinsic	intrinsic	Flip-Flop 7 LR Calculus 9	EFT / SF S SB / RIGHT	LEFT / SF S SB / RIGHT	left/right SF/SB S	2 4 -	left/right SF/SB S	2 4 -
Single Cross Calculus	relative	relative/intrinsic	intrinsic	10	LF SF RF LN S RN B SB RB	LF SF RF LN S RN B SB RB	SF/SB/LN/RN RF/LF/RB/LB S	8 4 -	SF/SB/LN/RN RF/LF/RB/LB S	8 4 -
Double Cross Calculus	relative	relative/intrinsic	intrinsic	15	LF SF RF LN S RN B SB RB	LF SF RF LN S RN B SB RB	SF/SB/LN/RN RF/LF/RB/LB S	8 4 -	SF/SB/LN/RN RF/LF/RB/LB S	8 4 -
TPCC	relative	relative/intrinsic	intrinsic	27	~~LF~~ SF ~~RF~~ LN S RN ~~LB~~ SB ~~RB~~	~~LF~~ SF ~~RF~~ LN S RN ~~LB~~ SB ~~RB~~	SF/SB/LN/RN RF/LF/RB/LB S	12 8 -	SF/SB/LN/RN RF/LF/RB/LB S	12 8 -
Binary Point Representations										
STAR$_1$(0)	absolute	intrinsic		5	LEFT / SF S SB / RIGHT		left/right SF/SB S		2 4 -	
STAR$_2$(0)	absolute	intrinsic		9	LF SF RF LN S RN B SB RB		SF/SB/LN/RN RF/LF/RB/LB S		8 4 -	
STAR$_4$(0)	absolute	intrinsic		17	~~LF~~ SF ~~RF~~ LN S RN ~~LB~~ SB ~~RB~~		SF/SB/LN/RN RF/LF/RB/LB S		12 8 -	
OPRA$_1$	intrinsic pair relations	intrinsic		4x4+1 =17 relation pairs	LEFT / SF S SB / RIGHT		left/right SF/SB S		2 4 -	
OPRA$_2$	intrinsic pair relations	intrinsic		8x8+1=65 relation pairs	LF SF RF LN S RN B SB RB		SF/SB/LN/RN RF/LF/RB/LB S		8 4 -	
OPRA$_4$	intrinsic pair relations	intrinsic		16x16+1=257 relation pairs	~~LF~~ SF ~~RF~~ LN S RN ~~LB~~ SB ~~RB~~		SF/SB/LN/RN RF/LF/RB/LB S		12 8 -	

Table 4.2: *Summary of the discussed aspects for ternary- binary point representations.*

Chapter 5

General aspects of object configuration description and reconstruction

The rescue scenario outlined at the beginning of chapter 1, demonstrates that a global overview of an object configuration has to be reconstructed from a description that is formulated from one or several local points of view. The description should be neither cognitively complex to produce from the local point of view nor should it be cognitively complicated to reconstruct from the given description. The challenge is that the observer (speaker) and the reconstructor (listener/hearer) have very different perspectives on the object configuration.

To describe an object configuration the positional relationships between the objects have to be given. In sections 5.1 and 5.2 it is shown that the perspective taken (the frame of reference used) in communication depends on people's preferences that are grounded in the language used, their cultural environment [Lev96], and on the task to fulfill [Tve99]. Depending on the circumstances, one out of the three alternative perspectives, *gaze tour*, *route tour* and *survey perspective* is used to describe an object configuration [Tve99]. The information necessary to reconstruct an object configuration into an absolute frame of reference is given in section 5.3. Section 5.4 describes the object relations necessary to describe objects' positions in relation to each other and section 5.5 combines the previously introduced aspects and presents the approach to describe an object configuration for reconstruction made in this thesis. Finally, section 5.6 provides a list of requirements that a representation scheme applicable for the route tour scenario must fulfill.

5.1 Taking perspective

A perspective entails a point of view, a reference object and a frame of reference. In verbal communication, perspectives are mixed depending on the purpose of the communication [Tve99]. Tversky, Lee, and Mainwaring [Tve99] distinguish between three basic perspectives: *gaze tour*, *route tour*, and *survey perspective*, which are frequently used to describe configurations of objects.

5.1.1 Gaze tour

A gaze tour describes the configuration from a fixed viewpoint. Objects' positions are given in relation to other objects. A gaze tour is for instance suitable for the description of rooms seen from the entrance and other small environments that can easily be overlooked from one single viewpoint.

5.1.2 Route tour

In a route tour, the hearer is taken on a mental tour through the environment. The speaker constantly changes the perspective and describes objects' positions in relation to the hearer's assumed position and orientation while the hearer is supposed to follow the route tour mentally. This technique is often applied to describe indoor environments with several rooms where the hearer is mentally walked from room to room.

5.1.3 Survey perspective

Using a survey perspective, the speaker takes a fixed viewpoint above the environment and describes the objects' positions to each other using an absolute frame of reference, for instance cardinal directions.

5.2 Changing perspective

Speakers are able to change perspective. If their culture or the language used contains more than one frame of reference, it is possible to change between them. The position of the furniture in a room can be described using cardinal directions, whereas the position of Belgium can be described with 'when you go from Germany to France you have Belgium to your right' (assumed that you walk in a straight line).

Using an intrinsic or absolute frame of reference, the positions of speaker and hearer do not matter, but using a relative frame of reference speaker and hearer must know the point of view from which the configuration is described. Reference objects are selected in order to produce an easy to understand description for the communication partner. Tversky, Lee, and Mainwaring [Tve99] call this the *Principle of Salience of the Referent Object.* Furthermore, the terms used in the description are selected after the *Principle of Ease of the Reference Terms* [Tve99], which means that the terms *in front of* and *behind* are preferred over the terms *to the left of* and *to the right of* which are considered to be cognitively more difficult [Fra90].

People often produce incoherent descriptions of an environment where they change perspective frequently; even though it is clear that a consistency of perspective provides descriptions that are more coherent. Tversky, Lee, and Mainwaring [Tve99] experienced that the type of task influences the speaker's chosen perspective. If the hearer has a cognitively more difficult task to do, the speaker will take the hearer's perspective to reduce the hearer's cognitive load. If the speaker has the cognitively more difficult task to do, he will take his own perspective to reduce his cognitive load. For more details on the experiments that led to these results and on the cognitive load that is attached to using a certain perspective, see [Tve99].

Levelt [Lev82] describes an experiment that analyzed the way people describe a network of coloured nodes, such as the one shown in figure 5.1, to a hearer that has the task of drawing the network from the given description. He found that two different approaches were used which were both tours from object to object. One was a gaze tour, where the speaker described the configuration from a fixed viewpoint, which is what Levelt calls the *deictic* perspective (relative frame of reference). The other is a route tour, where the speaker mentally moved from object to object and attached his own orientation to the object, so that the object inherited an intrinsic orientation that was used as intrinsic frame of reference to localize the next object. Levelt calls this the intrinsic perspective but points out that this perspective could also be regarded as deictic (relative), as the speaker uses his own (deictic) orientation, standing at another object's position [Lev82].

5.3 Alignment

The observer's task is to produce a description of an object configuration in such a way that the listener is able to reconstruct the configuration from it. The aim is that the reconstruction provides a representation from the survey perspective (map-like) of the original object configuration. This means that the description must contain the information needed to recognize the object configuration from a survey perspective, for instance to compare the reconstruction with a map, or the reconstruction with the original object configuration seen from a survey perspective, for instance from a helicopter.

Zimmermann and Freksa [Zim96] point out that research in cognitive psychology shows that people prefer to make use of rectangular reference systems for spatial orientation. A typical feature to distinguish between configurations is alignment. The object configurations in figure 5.2 are usually regarded as being distinct. The differences can be described by more than just the difference in distance whereas the differences in the configurations in figure 5.3 are obvious but

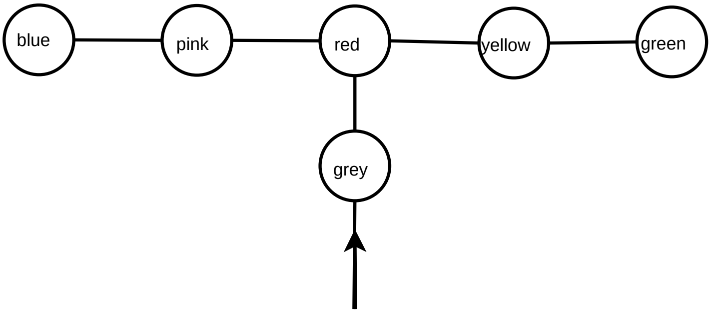


Figure 5.1: *An example of a colored node network, underlying Levelt's experiment [Lev82].*

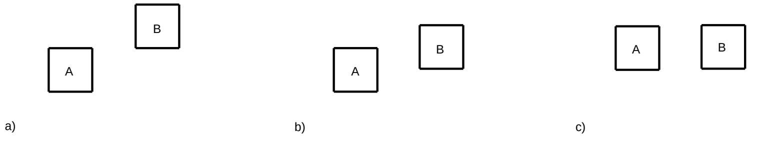


Figure 5.2: *Object configurations with objects aligned differently.*

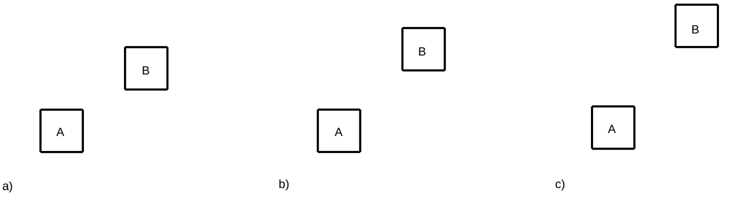


Figure 5.3: *Object configurations with objects aligned alike.*

very difficult to express without referring to the differences in distance. Reconstructing an object configuration containing more than two objects, the overall global alignment of all objects to each other needs to be preserved. This means that the relations used must be transitive. If object A is north of object B and object C is north of object B then there is no doubt about object C's position in relation to object A. Geographic maps represent object configurations in this way. They usually use an absolute frame of reference with cardinal directions, which are used to describe objects in relation to each other [Wil90; Hak94]. Geographic maps usually represent object configuration true to the scale whereby other spatial information besides the object's positional relations are preserved. For instance, the distances between objects in a map are usually proportional to the distances in the original object configuration. However, in the given task the object configuration descriptions are restricted to only positional object relationships.

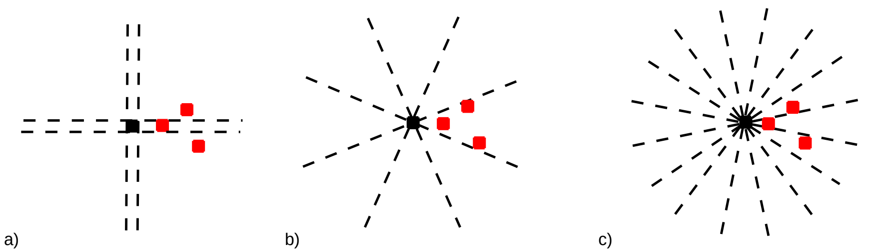


Figure 5.4: *Preservation of global alignment information. a) projection-based with neutral zone b) cone-based frame of reference with eight zones c) cone-based frame of reference with 16 zones.*

5.3.1 Preservation of global alignment information

In order to reconstruct the object configuration into an absolute frame of reference certain object alignment relationships must be preserved within the object configuration description. This information can be regarded as the qualitative counterpart of the object positions in latitude and longitude. Objects that are on the same latitude must be classified as such, as must objects lying on the same longitude. This information is best preserved by a projection-based frame of reference.

In the projection-based frame of reference with neutral zone in figure 5.4a) the three red objects have three different relationships to the black reference object (e.g. northeast, east, and southeast).

A cone-based frame of reference does not preserve this information as reliably. Depending on the distance from the reference object, objects that are distinguished in a projection-based frame of reference, might now be classified within the same relationship. This is the case for the three objects in figure 5.4b) that all three are now classified to be in the relationship east of the reference object. The further away these objects are from the reference object, the error in position increases in comparison to a description of latitude. Such errors might accumulate in the reconstruction. The more zones a cone based frame of reference uses the smaller the error rate becomes, but with increasing distance to the reference object the error is still present. In addition, the user's cognitive load increases with the number of zones used.

5.4 Object representation

Real world objects occupy space and to abstract them to lines or points leads to loss of the information about relative size and alignment. This information might not be important for objects that are relatively far apart from each other. To describe configurations of objects that are close to each other, such as indoor environments (e.g. the description of furniture layout) or outdoor environments (e.g. the description of a city- or garden design) the ability to describe the object's alignment is essential. It is seen as an advantage if the configurations shown in figure 5.2 are distinguishable, which is the case in a rectangle representation.

Any object's position can be represented by its *Minimum Bounding Rectangle* (*MBR*), and the spatial relations between the actual objects can be approximated by their *MBR* relations [Pap95]. The 169 *Minimum Bounding Rectangle relations* that can be distinguished taking alignment of the objects into account have been introduced by Papadias et al. [Pap95], an approach further explained in chapter 3. *MBR* relations are based on Allen's Interval Calculus [All83] that is also explained in chapter 3. For the *MBR* relations, the 13 interval relations: *before*, *meets*, *overlaps*, *starts*, *contains*, *equals*, *finishes*, *started by*, *contained by*, *finished by*, *overlapped by*, *met by*, and *after* are used in tuples of relations where one entry gives the relationship of the rectangles in one dimension, and the second entry gives their relationship in the second dimension.

The frame of reference used for Papadias et al.'s *MBR* relations is absolute. Switching to an intrinsic frame of reference the position of an object must be given in relation to the reference object's intrinsic, or pseudo intrinsic orientation. The nine two-dimensional regions that provide a complete partition of the space around the reference object can be named to express the direction in which a target object can be found. Figure 5.5 shows the result. The position classes (areas) are named with capital letters: *STRAIGHT FRONT* (*SF*), *RIGHT FRONT* (*RF*), *RIGHT NEUTRAL* (*RN*), *RIGHT BACK* (*RB*), *STRAIGHT BACK* (*SB*), *LEFT BACK* (*LB*), *LEFT NEUTRAL* (*LN*), *LEFT FRONT* (*LF*), and *SAME* (*S*). These nine basic relations are all that is needed to completely describe an object configuration. All other relations are compositions of them. In figure 5.6 it is shown that each basic relation includes a group of MBR relations. Assuming that the reference rectangle is facing the top of the page, the first row in Papadias et al.'s

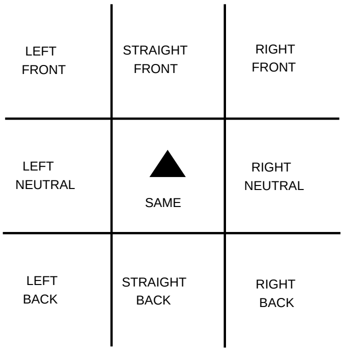


Figure 5.5: *Position classes based on differences in alignment.*

MBR relation table (the *MBR* relations R_{1_1} to R_{1_13}) can be expressed using the intrinsic frame of reference, as:

R_{1_1} ⇒ *LEFT FRONT*
R_{1_2} ⇒ *LEFT FRONT* and aligned with *STRAIGHT FRONT*
R_{1_3} ⇒ *LEFT FRONT* and *STRAIGHT FRONT*
R_{1_4} ⇒ *LEFT FRONT* and *STRAIGHT FRONT* and aligned with *RIGHT FRONT*
R_{1_5} ⇒ *LEFT FRONT* and *STRAIGHT FRONT* and *RIGHT FRONT*
R_{1_6} ⇒ *STRAIGHT FRONT* and aligned with *LEFT FRONT*
R_{1_7} ⇒ *STRAIGHT FRONT* and aligned with *LEFT FRONT* and aligned with *RIGHT FRONT*
R_{1_8} ⇒ *STRAIGHT FRONT* and *RIGHT FRONT* and aligned with *LEFT FRONT*
R_{1_9} ⇒ *STRAIGHT FRONT*
R_{1_10} ⇒ *STRAIGHT FRONT* and aligned with *RIGHT FRONT*
R_{1_11} ⇒ *STRAIGHT FRONT* and *RIGHT FRONT*
R_{1_12} ⇒ *RIGHT FRONT* and aligned with *STRAIGHT FRONT*
R_{1_13} ⇒ *RIGHT FRONT*

An object can be completely within one region, or can be overlapping several regions. Furthermore, an object can be aligned with a neighbouring region. The object's position is described by a combination of the names of all regions the object is at least partly inside and the naming of neighbouring regions the object is aligned with. Leaving the alignment information aside, 36 of the originally 169 relations remain which only give the positional information. Twenty-seven of these are combined relations in the form of more than one region which the object is partly inside.

Many approaches to qualitative reasoning about orientation use a set of eight basic relations (nine if the position of the reference object itself is also taken into account). Freksa's Single Cross Calculus uses the eight regions *straight front*, *right front*, *right neutral*, *right back*, *straight back*, *left back*, *left neutral* and *left front* [Fre92b]. Mukerjee and Joe [Muk90] use eight regions around their reference

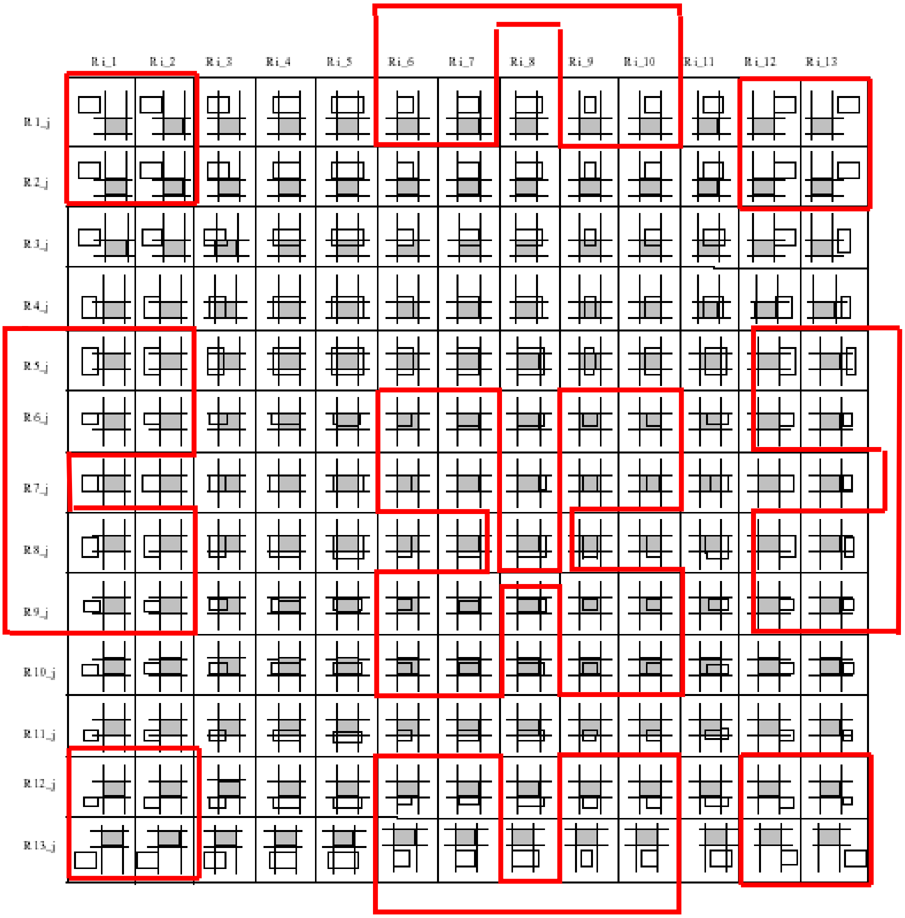


Figure 5.6: *The nine relations are groups of several MBR relations. The groups are circled in red on Papadias' original MBR table [Pap95].*

object that are numbered *1* to *8*. Goyal and Egenhofer [Goy01], in their Direction Relation Matrix and Ligozat [Lig98], for his cardinal directions, settled for the eight regions *north*, *north east*, *east*, *south east*, *south*, *south west*, *west*, and *north west*, using an absolute frame of reference. Fernyhough et al. [Fer00] use the same eight regions as Mukerjee and Joe but give them the intuitive names: *ahead*, *ahead right*, *right*, *behind right*, *behind*, *behind left*, *left*, and *ahead left* to be used within the traffic domain.

5.5 Components of object configuration description

In the scenario described earlier, the observer is in an environment too wide to be overlooked from one position, and therefore has to move around in order to provide position relationships for all objects. This excludes the possibility of a description from a survey perspective or of a gaze tour given from a single viewpoint. The observer describes parts of the configuration from different viewpoints, using intrinsic frames of reference. In order to allow the listener to reconstruct the described configuration into a global frame of reference, with the global alignment information preserved, the observer needs to explain exactly what his viewpoint is for each part of the description. If the orientation of that applied frame of reference is different from the previously applied this orientation change must be traceable for the listener. The listener has a survey perspective onto the reconstruction. Furthermore, he has the notion of the observer's orientation and orientation information for some of the objects. Therefore, the listener is able to switch between the different frames of reference that the observer uses and able to keep track of the object configuration description.

To solve the task of describing an object configuration from different local viewpoints by an observer on site and to reconstruct it into a map-like representation from the survey perspective, the previously discussed aspects of object relation descriptions and information necessary to establish a map-like reconstruction of an object configuration are combined and the following conclusion has been drawn: An appropriate object configuration description can be given by a combination of a route tour with several gaze tours, one for each step within the route tour. In addition to each gaze tour that describes the position of the seen objects in relation to the reference object, the relative positions between these objects have to be described. This is done in a *momentarily applied absolute frame of reference*, a concept that is further explained in section 5.5.3.

5.5.1 Route tour observation

To describe the objects' relative positions a *pseudo intrinsic reference frame*, similar to the reference frame used by Levelt's subjects [Lev82] during the route tour, is used for the objects that function as reference objects. Standing at an object the observer uses his own orientation that he had when he arrived at the object and merges it onto the object as if it would be the object's intrinsic orientation. This way, the listener always knows what orientation an object has, as he knows from which direction the observer approached the object, assuming that the observer always moves in a straight line between objects.

Considering that people in wayfinding or route description tasks usually distinguish between eight direction classes, that Klippel [Kli03a; Kli03b], calls *sharp right*, *right*, *half right*, *straight*, *half left*, *left*, *sharp left* and *back*, eight directions are used in this approach. In order to be consistent with the terminology for the objects' positions introduced below, the eight orientations will be named in lower case: *right back* (*rb*), *right neutral* (*rn*), *right front* (*rf*), *straight front* (*sf*), *left front* (*lf*), *left neutral* (*ln*), *left back* (*lb*) and *straight back* (*sb*). Standing at an object, the observer can leave that object in one of these eight directions.

Using an intrinsic frame of reference for each object leaves the possibility open to actually use the object's own intrinsic frame of reference if available. In this case, the observer has to explain, how the object is oriented, which can be done by naming the position of the object he came from in relation to the new reference object. Using the object's own intrinsic orientation, the observer might approach an object from any direction. In the present case, where the orientation of the arriving observer sets the intrinsic orientation for the reference object the observer will always approach an object from behind (*straight back*). While approaching the object the observer's viewpoint and the place of the new reference object differ. The frame of reference that matches the process of approaching the object is the frame of reference used in Hausa [Hil82].

In figure 5.7 a possible route tour between four objects is pictured. Grey lines inside the object represent the eight directions that the observer can take. The direction that he takes to continue his route is shown with a red arrow, pointing in the *leaving direction*, the arrow line is continued as a dashed red line outside the object to indicate the route. The direction the observer approaches the object from is shown by a dashed black line outside the object that becomes a black arrow pointing at the centre point of the object. This *incoming direction* sets the intrinsic orientation of the object, whereby the object's front is the object's edge or corner the arrow points to. The line from the middle of the object to its front is represented in black. Somewhere, roughly in the middle of the two objects, the dashed line turns from red (leaving direction) to black (incoming direction). In figure 5.7 the observer starts at object A. He decides to move *right front* to object C, his incoming direction sets the intrinsic frame of reference for object C. He proceeds in the direction *right front* to object F. From there he takes the direction *left front* to object H.

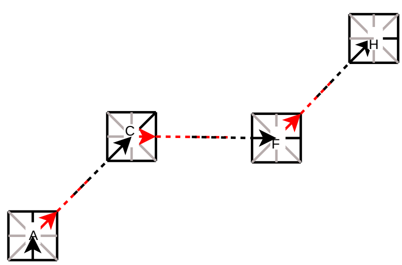


Figure 5.7: *Route tour between reference objects.*

5.5.2 Gaze tour observation

The purpose of the route tour presented above is that the observer describes parts of the object configuration with gaze tours from different positions. Figure 5.8 describes the route starting from object A via object C to object F. At the position of object A the observer gives the relative positions of the objects B, D and C as B being to the *LEFT FRONT*, (B *LF* A), C to the *RIGHT FRONT*, (C *RF* A), and D to the *RIGHT NEUTRAL*, (D *RN* A), of reference object A. He continues his route to object C, which means that he leaves object A in the direction *right front* (*rf*).

The incoming direction at object C is *straight back* (*sb*), because the orientation that the observer has when he arrives at object C is attached to the object as its intrinsic orientation. Standing at object C the observer describes the positions of the objects B, E, F, and D in relation to object C as (B *LB* C), (E *SF* C), (F *RF* C), and (D *RN* C). The relationship of object A to object C does not have to be explicitly given, as the relationship (C *RF* A) and both object's relative orientations are known, the relationship (A *SB* C) can be concluded.

The observer continues his route to object F in the direction *right front* (*rf*) and approaches object F from *straight back* (*sf*). The object adopts his orientation and the positions of the objects E, H, D and G with regard to object F are therefore (E *LB* F), (H *LF* F), (D *RB* F), and (G *RF* F). In the example, the observer gives the relative position of all the objects that he can see from a viewpoint. Some objects are visible from more than one viewpoint and therefore their position is given in relation to more than one object. The more relationships are presented that describe the position of an object the more accurately the object's position can be determined and the less reasoning about unknown relationships is necessary, which leads to a more exact reconstruction.

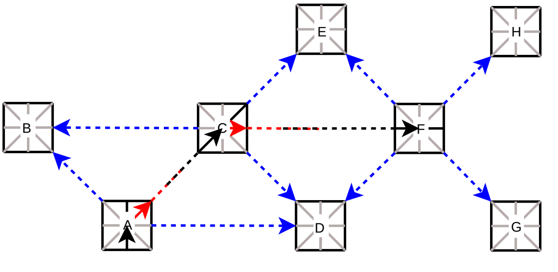


Figure 5.8: *Route tour (black and red) combined with several gaze tours (blue).*

5.5.3 Momentarily applied absolute frame of reference observation

In the example shown in figure 5.8 the observer starts at object A and during his gaze tour from this viewpoint, he describes the positions of object B and object

C in relation to object A. The listener learns first about object A, and represents it in the reconstruction. As it is the first object within the reconstruction, any orientation is possible and the listener is free to choose one out of the eight possible orientations. For simplicity it is assumed that he uses cardinal directions with north at the top of the page. In this example it is assumed that he chooses the orientation *north* for object A. Thereafter he adds object B *LEFT FRONT* of object A in relation to object A's orientation, which translates to *northwest* in the absolute frame of reference.

So far, the objects could just be placed into the reconstruction without any complications. When object C is to be added *RIGHT FRONT*, and therefore *northeast*, of object A, the listener has no information on the relationship object C has to object B. Depending on object B's and object C's sizes, the relationship of object C to object B in the reconstruction could be *northeast*, *northeast_and_east*, *east*, *east_and_southeast*, *southeast*, or *northeast_and_east_and_southeast*.

It is advantageous, if the observer mentions the relationship between the two objects. For the moment the optimal is assumed whereby the observer, standing at object A sees both object B and object C. Ideally, he is able to classify their relationship to each other from his current position. Therefore, one of these objects becomes the reference object and the other's position is described in relation to it. Several alternatives exists of how a frame of reference could be attached to the reference object in order to describe this relationship.

Firstly, a pseudo intrinsic frame whose orientation is similar to the observer's orientation while the observer is facing the reference object. This is similar to the pseudo intrinsic frame of reference used during the route tour where the reference object inherited the observer's orientation by arrival. The difference is that the observer is not estimating the relative position of the target object from the position of the reference object but from his current viewpoint. In the example this translates to the observer standing at object A facing object B and describing object C's position in relation to B seen from object A. This adds additional load to the listener, who not only has to follow the orientation attached to the reference objects, but also from what viewpoints relationships were estimated.

Secondly, if available, the reference object's own intrinsic frame could be used. This alternative contains the same difficulty that the relationship between the objects is estimated from the observer's current viewpoint. Additionally, the orientation of the reference object has to be explicitly transmitted to the listener. Furthermore, this solution does not work for objects without an intrinsic frame of reference. In both alternatives the observer changes his orientation so that he always faces the reference object used for a relationship and the listener must follow all orientation changes. The more changes there are the more complexity is added to the process and the possible sources of error increase.

Finally, the alternative chosen for this thesis, suggests that the observer describes the relationships between the objects he sees from his current position in relation to each other, whereby he keeps the same orientation for all relationships given. This is called a description in a *momentarily applied absolute frame of* *reference.* In the same way that cardinal directions can be used, fixed bearings that the observer establishes according to his own orientation are used to describe the relations between the objects that are seen from the current viewpoint. As mentioned before it is assumed that all objects occupy a two-dimensional space on the ground that can be approximated by their minimal bounding boxes. The frame of reference applied here would therefore become a projection based frame of reference, exactly like the frame of reference used for the reference object at which the observer stands. In order to keep the terminology simple, the same terms, introduced before are used to describe an object's relationship to a reference object in a momentarily applied absolute frame of reference. This means for the objects in figure 5.8 that object C is *RIGHT NEUTRAL* of object B, object B is *LEFT NEUTRAL* of object C, object D is *RIGHT BACK* of object C and *RIGHT BACK* of object B and object B and object C are both *LEFT FRONT* of object D.

5.5.4 Summary of the description process

The steps of the description process are summarized within the activity diagram presented in figure 5.9. The observer moves from object to object in a route tour, and a pseudo intrinsic frame of reference inheriting the observer's orientation, is attached to each object. Therefore, the observer always approaches a new object from *straight back* (*sb*). At each step within the route tour he provides the relative position of all objects he can see from where he stands in relation to the object he stands at in a gaze tour. Eight (nine if the position of the reference object itself is included) position classes are available for the observer to classify object positions.

A description of the relations between the objects in a gaze tour follows. After that, the observer keeps his position and orientation and describes the relations in a momentarily applied absolute frame of reference with a neutral zone, whose axes are given by the observer's orientation. When all relations have been described the observer explains to which of the other objects, previously mentioned, he continues his route. The listener is able to switch between the different frames of reference due to his survey perspective onto the reconstruction area using its own absolute frame of reference. For reasons of convenience, cardinal directions are used.

5.6 Object Configuration Reconstruction

An object configuration shall be reconstructed from a description that is established in the way described in the previous section. The reconstruction has to match all the nine qualitative relations that were considered necessary (*SF*, *RF*, *RN*, *RB*, *SB*, *LB*, *LN*, *LF*, *S*). The reconstruction process should be cognitively easy to fulfill.

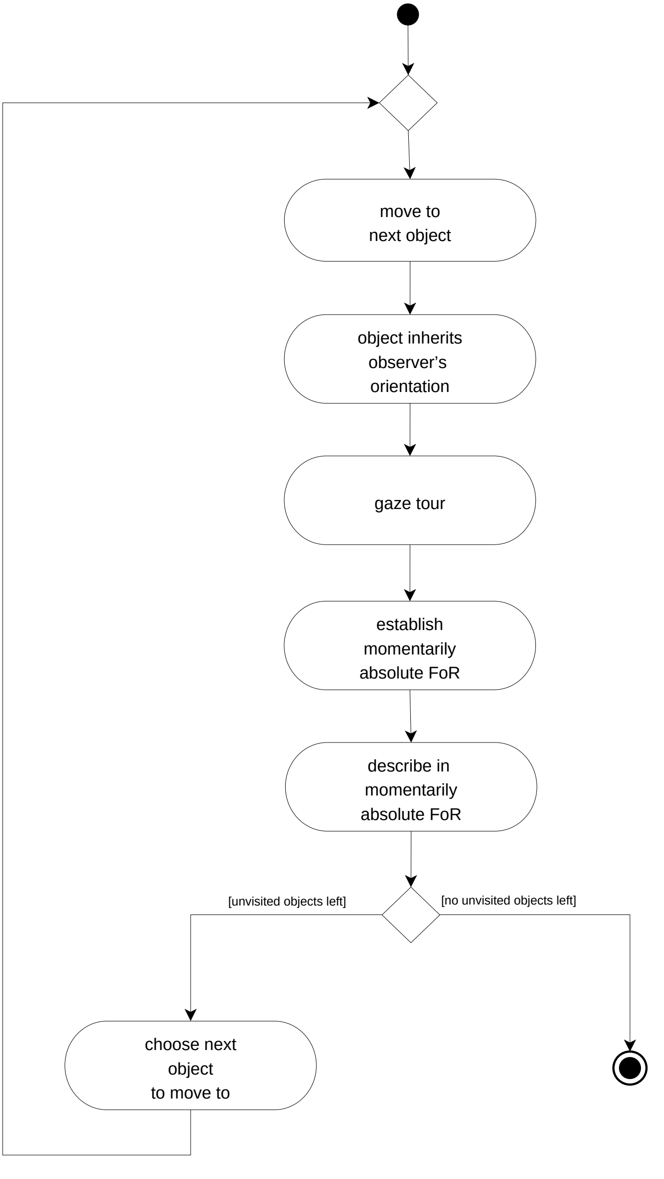


Figure 5.9: *The route tour description process.*

5.6.1 Reasoning with mental models

Research in cognitive psychology examines the human cognitive processes of constructing mental models and reasoning with these models. Even though the focus in this thesis lies on physical object configuration reconstruction rather than reconstruction with mental models, research in mental model theory (MMT) [JL91; JL01] might help the understanding of typical problems and preferences concerning reconstruction processes in general.

Mental Model Theory (MMT)

Johnson-Laird [JL01] and Johnson-Laird and Byrne [JL91] suggested that people when mentally constructing spatial relationships between objects, use a symmetrical mental spatial array into which they insert the objects. The objects' relationships represent the scenarios described by the premisses.

Preferred Mental Model (PMM)

Psychological experiments about human spatial reasoning with mental models show that people usually only construct one mental model that is consistent with the given premisses, even for indeterminate problems. Indeterminate problems are those for which several consistent reconstruction alternatives are possible. When a mental model has to be varied in order to match additional premisses the new mental model is created as a variation from the PMM with as little difference as possible [Rau05; Jah04; Kna95].

First free fit (fff)

When objects have to be inserted into the mental array, the array cells that lie in the given relationship are scanned from the reference object and the object is inserted in the next empty cell found. This strategy is called the first free fit (fff) opposed to the first fit strategy (ff) where the object is inserted into the array cell adjacent to the reference object and an object that could have possibly been there earlier is moved to the next adjacent cell [Rag06].

5.6.2 The SRM model

Ragni et al. [Rag05] present the SRM model (Spatial Reasoning by Models) for human spatial reasoning by means of mental models. It was developed to simulate human spatial reasoning. The spatial working memory is represented by a two-dimensional grid in which a simulated spatial focus places objects, inspects reconstructed object configurations in order to draw conclusions, and manipulates the objects' positions.

In the SRM model, four relations: *to the left of*, *to the right of*, *in front of*, and *behind* are represented. The relations have to be interpreted so that objects that are to the left or to the right of the reference object are on the same horizontal line as this object and objects that are in front or behind the reference object are on the same vertical line as it. A relation is a triple (X, r, Y) where X is the target (referent), Y is the reference object (relatum) and r the relationship between the two. When objects have to be inserted into the grid, the spatial focus locates the reference object and according to the provided relationship, inserts the target into the grid. Mental spatial reasoning is modelled in three phases: the construction phase, the inspection phase, and the variation phase.

Formally the SRM model is a quintuple (I, O, A, F, C) where:

- I is the input mechanism to the model.
- O is a set of object names.
- A is the spatial array.
- F is the focus.
- C is a control process, controlling the focus and other executive functions.

The following examples of input premisses are cited:

The hammer is to the right of the pliers.
The screwdriver is to the left of the pliers.
The wrench is in front of the screwdriver.
The saw is in front of the pliers.
What is the relation between the wrench and the saw?

During the construction phase one premiss at a time is read and the described target object is inserted into the grid. If the reference object was already in the grid the target object is inserted by first free fit [Rag06] into the next free grid cell in order for the relationship given in the premiss to hold. If the reference object is not in the array a new spatial array is generated and reference object and target object are inserted according to the relationship given in the premiss. When reference object and target object are both members of two already existing but separate arrays, the two arrays are connected into one array, and the relationship functions as the connector.

The SRM model stops when it has found one model. This, with reference to the preferred mental model theory (PMM) [Kna95; Jah04; Rau05], is called the PMM. If model variation is necessary, SRM starts from the PMM and generates new models by changing it with minimal changes following the principle of conceptual neighbourhood [Rau05]. The SRM model has been implemented with the ACT-R production system. The result allows to predict reasoning times of human spatial mental reasoning. For more information on the implementation see [Boe06; Rag07].

5.6.3 Requirements for object configuration description and reconstruction

The strategies chosen for an observer on site are outlined in a route tour through the environment. The route tour is combined with several gaze tours from certain reference objects and a description of object relationships in a momentarily applied absolute frame of reference.

The reconstruction process as the focus of this thesis, aims at a physical reconstruction of a described object configuration opposed to a mental reconstruction as in [Rag05; Boe06; Rag07]. Many more objects than in the psychological experiments are to be reconstructed and a person is usually not able to hold all the objects and their relationships in mind. Instead he has to successively reconstruct the configuration physically. The listener is allowed or even expected to use a strategy for the reconstruction. This strategy is meant to make the reconstruction process easier for the listener.

From the MMT and the fff theory it is known that people prefer not to relocate objects once inserted into the model. This is, of course, also an advantage within a physical reconstruction. Furthermore, MMT shows that people prefer to establish only one mental model, the preferred mental model (PMM) and only look for alternative models if necessary. In many cases the PMM is used instead of a correct alternative which might lead to wrong conclusions about implicit relationships within the model. For the physical reconstruction task it is desirable to only work with one reconstruction but to keep this reconstruction correct at all times with all explicit and implicit relationships between the objects included.

The summarized requirements for a description and reconstruction technique allowing for a map-like reconstruction within a global frame of reference are as follows:

- Objects are to be represented as rectangles.
- A rectangular frame of reference with neutral zone is needed.
- An intrinsic (or pseudo-intrinsic) orientation must be attachable to any reference object.
- The frame of reference must be aligned to the underlying global frame of reference even though its intrinsic orientation might be at an angle to the underlying global frame of reference. This is necessary in order to allow for an object configuration reconstruction from the survey perspective.
- The space around the reference object has to be divided into eight position classes.
- Eight orientation classes for the objects and for the orientation of the observer and the direction of the observer's movement have to be distinguishable.

- The position classes (except the class that describes the position of the reference object itself) must coincide with the eight orientation classes.
- The object relations used to describe the relative object positions must be binary.
- Objects whose orientation are unknown have to be representable.
- The reconstruction shall be consistent with the premisses that have been considered within the reconstruction, at any time during the reconstruction process.
- An object's position has to be calculated completely before the object is inserted into the reconstruction.
- The strategy must be general and result in a consistent reconstruction for any number of objects.
- The strategy shall guide the listener during the reconstruction process and help him to avoid typical mistakes.
- The strategy must avoid the necessity of recalculating objects' positions, when new objects have to be inserted into the reconstruction in between already represented objects.

The cognitive effort for both participants, that is, observer and listener, needs to be minimized. For example, the orientation information the observer has to keep in mind concerning his and the objects' orientation within the underlying global frame of reference needs to be minimal. To measure cognitive effort, carefully designed experiments are necessary. However, experiments of this kind are not within the scope of this research. Nevertheless the cognitive effort of a description or reconstruction process is taken into account by applying the intuitive rule of thumb, that 'the less to keep in mind, the better'.

This chapter has introduced general aspects of object configuration description and reconstruction from a psychological point of view. Furthermore, it describes a cognitively motivated solution to describe an object configuration qualitatively from different local viewpoints in order to reconstruct the configuration based on the resulting description. For the reconstruction process mental model theory has been considered in order to become aware of possible mistakes that might arise even in physical object configuration reconstruction and that should be avoided by a guided reconstruction process. The following chapter introduces the representation scheme QuaDRO (Qualitative Description and Reconstruction of object Configurations) that has been developed to match the requirements of object configuration description and reconstruction presented above.

Chapter 6

QuaDRO, a representation scheme to qualitatively describe and reconstruct object configurations

The overall motivation to develop QuaDRO is the need for natural language communication about observed objects, between a person and an autonomous system. The most salient example is that it could be used in connection with the WITAS helicopter [Doh00] where it could describe any kind of object configuration that mission control might be interested in at a given time. Traffic situations, damaged land areas, after an earthquake or a flood, for instance, places where people are trapped, or oil slicks in the sea after oil tanker accidents, are some possible scenarios. However, QuaDRO's use is not restricted to the above scenarios nor is it restricted to use within the WITAS project. Any kind of object configuration seen from above can be described and reconstructed. Ancient rock formations, island distributions, things lying on a desk or garden design are some examples not within the WITAS project's scope. Even star constellations can be described and reconstructed.

Where people often describe object positions by naming the object's qualitative positional relationships, autonomous systems usually collect quantitative sensor data. To improve natural language communication between an intelligent system and a person, a system must be able to understand and to produce relative object position descriptions like 'to the left of', 'to the right of', 'in front of', etc.

QuaDRO is a representation scheme developed to qualitatively describe and reconstruct object configurations seen from different local viewpoints. An observer might have a limited field of view, which leads to a qualitative description of just a small part of the configuration at a time. Several descriptions from different viewpoints together might provide all the information needed to reconstruct the configuration into a map-like picture. Using QuaDRO, the observer does not have to know his global orientation and can change his location and orientation freely almost without keeping track of it. QuaDRO uses nine positional equivalence classes to describe qualitative positional relationships between two objects. Therefore, the cognitive load for a human observer or listener (reconstructor) is kept minimal.

The description provided by the observer is a set of qualitative positional constraints between objects. In order to graphically reconstruct the object configuration described, a model of the given set of constraints, mapping the objects onto concrete positions within a Cartesian coordinate system, is needed. For the infinite domain of coordinates in the plane, standard constraint satisfaction strategies such as backtracking or search do not apply. Therefore QuaDRO uses a novel approach for computing a single geometric model for a set of distinct local observations.

Two underlying general scenarios, presented in section 6.1, set the scope of QuaDRO's applicability. In the first scenario, an autonomous system describes an observed object configuration to a person. In the second scenario, a human observer describes an object configuration for an expert system. The two central aims in QuaDRO are (1) to automatically generate an object configuration description (*ocd*) from a given geometrical scene description (*GSD*, a term adopted from [Nov86]); and (2) to automatically reconstruct the configuration from an object configuration description into a reconstructed scene description (*RSD*). In the case of the observer being an intelligent system, as described in the first scenario, the input into the system is the geometrical scene description, produced by a simulator or a vision system, and the output is the corresponding object configuration description. In the case of a human observer, as presented in the second scenario, the input into the system is the written object configuration description and the output is the reconstructed scene description. To verify the qualitative correctness of the process of obtaining an *ocd* from a given object configuration and of establishing a reconstruction from a given *ocd*, these two processes are combined into one, which is briefly described in section 6.2.

Section 6.3 provides a brief introduction to the object configuration description process used in QuaDRO. To establish object configuration descriptions that alleviate the reconstruction process, it is an advantage to consider several aspects of the configuration, such as the order in which the objects are described during the description process. Due to the difficulties that might arise during the reconstruction it is explained how these can be prevented if already considered during description. The reconstruction described in section 6.5 takes place on the QuaDRO grid. The grid allows for scalable granularity of object positions and is described before the reconstruction process in section 6.4. The description and reconstruction operations will be explained using the simple case of a completely described configuration of objects that all have the same orientation or no orientation at all. Most techniques used in QuaDRO can be explained within this example. Additional techniques used to reconstruct configurations of objects with intrinsic orientation are described in chapter 7. Techniques applied to reconstruct object configurations from incomplete object position information are presented in chapter 9.

6.1 Scenarios

The two scenarios illustrated in figure 6.1a) and 6.1b) and described below, sketch the typical high-level flow of information in situations where QuaDRO is applied. These scenarios are important for two reasons. (1) They illustrate the area of application for the QuaDRO representation scheme. (2) They describe the methodology used to validate the design. In order to verify the different parts of the description and reconstruction processes, the parts of both scenarios fulfilled by the autonomous system are combined to one chain of processes, illustrated in figure 6.1c).

6.1.1 Obtaining an object configuration description from a given object configuration

An intelligent agent functioning as observer talks to a human listener. The agent observes an object configuration using some kind of observing device. The data collected by the device is evaluated to a quantitative representation. In this representation, all observed objects are recognized and identified. The object's positions are kept in a x-y-coordinate system. This description can be regarded as a form of geometrical scene description (*GSD*). A *GSD* is a description of already recognized objects in a world coordinate system where all objects are described with their positions and orientations [Nov86].

Problems that arise due to, for example, noise in the camera data are not within the scope of this work. The image processing tool that produces the *GSD* will solve these problems in one way or another; it follows that the *GSD* more or less reflect real world scenarios. To avoid being dependent on frequently reproduced *GSD*s from real world situations requiring autonomous systems' outdoor missions, a simulator is used to generate *GSD*s. The *ocd* is a qualitative description of the quantitative information contained in the *GSD*. Instead of coordinates, qualitative position terms, as introduced by [Fre92a], and describing the qualitative areas of the frame of reference presented in 5.5 are used, such as: *STRAIGHT FRONT* (*SF*), *RIGHT FRONT* (*RF*), *RIGHT NEUTRAL* (*RN*), *RIGHT BACK* (*RB*), *STRAIGHT BACK* (*SB*), *LEFT BACK* (*LB*), *LEFT NEUTRAL* (*LN*), *LEFT FRONT* (*LF*), and *SAME* (*S*). Knowledge quantitatively deposited in the *GSD* needs to be imparted to the human listener in a natural language via a dialogue system. The spoken text must be able to comprehend all information necessary for the listener to be able to reconstruct the configuration. It is therefore based on the *ocd*, and its range of phrases and expressions depends on the

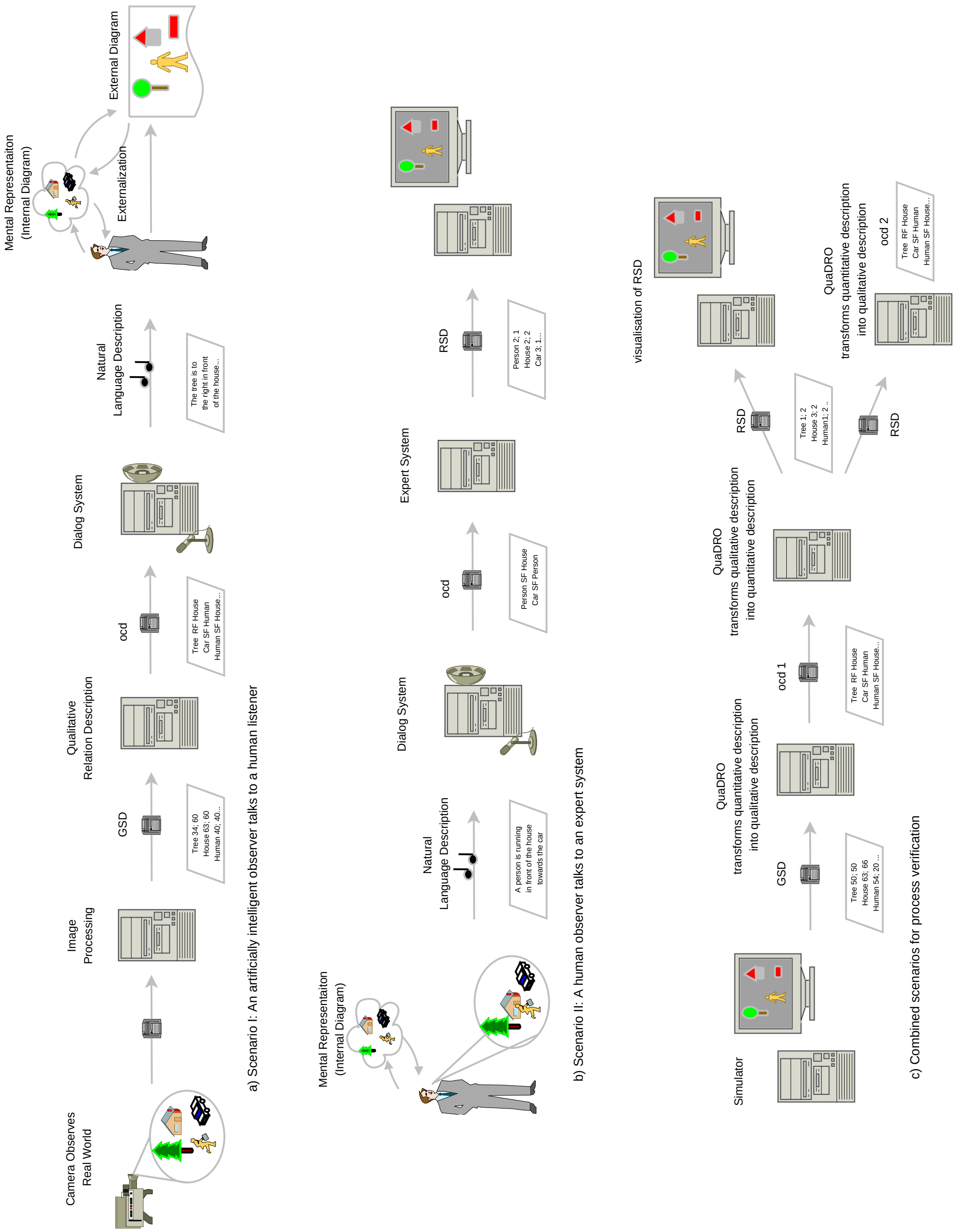


Figure 6.1: *Example scenarios. a) An artificially intelligent autonomous observer describes to a human listener. b) A human observer describes to an artificially intelligent system. c) Combined scenarios for process verification.*

dialogue system's abilities. The dialogue system itself is not within the scope of this work and for the remainder of this thesis, there is no differentiation between the natural language description based on the *ocd* and the *ocd* itself.

A listener has the job of reconstructing the object configuration from the provided *ocd*. In the dialogue situation, the *ocd* will be heard only. Listening to it leads to a mental representation [Bar02; Kos05; Nig05] of the objects and their relative positions. Eventually, the listener has to externalize this representation for instance as a sketch on a piece of paper, a blackboard, or a digitalization table. Perhaps, he might even have to rebuild the configuration with model objects or building blocks. Externalizing the representation implies needing to decide on a distinct position for each object. An object described as being *RF* of a reference object is somewhere within the infinite region *RF*. Placing the object into the reconstruction implies to decide on its specific position. The human listener may construe the whole reconstruction mentally before externalizing it as a whole. However, usually *ocd*s are too comprehensive and he needs to externalize parts of the reconstruction during the reconstruction process. The problem of relocating objects arises when further information adds new constraints to the objects' areas.

QuaDRO provides a strategy for a reconstruction that allows externalized diagrams wherein the objects' qualitative relationships do not need to be retracted in the course of the process. Using this strategy, the partly reconstructed configuration at any time provides qualitatively correct object position relationships of all objects already represented. However, in the case of incomplete configuration descriptions, the size of objects previously placed into the reconstruction might have to be adjusted. Nevertheless, for incomplete descriptions, the reconstruction is qualitatively correct after each completed reconstruction step.

6.1.2 Obtaining a reconstruction from a given object configuration description

This scenario describes a human observer talking to an expert system. The person, not using any measuring tools, describes an object configuration observed from one or several viewpoints to the expert system. The observer gains a mental representation of the parts of the configuration creating a more or less complete whole. This internal representation together with the real world situation, functions as the basis for his natural language description of the configuration. The extent to which each of the two aspects are used is not within the scope of this research. Knowing that the configuration has to be described for the purpose of reconstruction, the description will concentrate on the relative object positions. Additional distance information is not needed. Ideally, the observer's description formulation is not restricted and it is the task of a dialogue system to extract an *ocd* from the natural language. The *ocd* will be passed on to the expert system for reconstruction. However, depending on the dialogue system's power, the observer has to stick to certain routines producing the description. In the remainder of this text, therefore, there will be no distinction between the actual natural language description and the *ocd* that the dialogue system extracts from it.

The *ocd* is the basis from which the expert system gains its understanding of a given situation. Qualitative position relations usually allow an object to be somewhere within a certain area. The expert system is expected to be able to operate directly on these region constraints. Nevertheless, the vague object positions induce a lot of uncertainty, not only about the concrete object positions but also about how to reason with the information. Reasoning under uncertainty and reasoning with fuzzy information are challenging research areas where qualitative strategies and common sense reasoning strategies gain more and more importance. One solution promoted in this thesis is to solve this problem the same way often ascribed to people, namely to decide on a concrete object position that conforms with all given constraints and to take that as the basis for further reasoning. QuaDRO provides a strategy to concretize the object's position. If all relationships necessary to place the object are available, a corresponding number of cells in the underlying reconstruction grid is allocated for the object. In case of incomplete knowledge about the relationships to other objects enough space is allocated so that every possible relationship is covered. The reconstruction process results in the reconstructed scene description (*RSD*) which has the same format as the geometrical scene description.

6.2 Process verification using the scenarios in sequence

To verify the qualitative correctness of the process of obtaining an *ocd* from a given object configuration and to establish a reconstruction from a given *ocd*, these two processes are combined into one sequence. At first, ocd_1 is established on the basis of the geometrical scene description. It is then fed into the reconstruction process. The resulting reconstruction can quantitatively be described by the reconstructed scene description. This in turn is used as input into the description process to establish ocd_2 for the reconstruction. If the reconstruction is correct regarding all qualitative positions using the nine positional equivalence classes, ocd_1 and ocd_2 are identical.

6.3 The description procedure

The purpose of the description procedure is to construct an *ocd* consisting of binary object relationships. Every relationship describes an object's position in relation to a reference object. To obtain such a relationship a projection based frame of reference with neutral zone as shown in figure 5.5 in chapter 5 using the nine following basic relations is applied: *STRAIGHT FRONT* (*SF*), *RIGHT FRONT* (*RF*), *RIGHT NEUTRAL* (*RN*), *RIGHT BACK* (*RB*), *STRAIGHT BACK* (*SB*), *LEFT BACK* (*LB*), *LEFT NEUTRAL* (*LN*), *LEFT FRONT* (*LF*), and *SAME* (*S*). In the case described here (chapter 6), all objects are orientationless, meaning that they all are regarded as having the same orientation so that the intrinsic frame of reference can be applied.

In the corresponding illustrations throughout this chapter, it is assumed that the objects' fronts point towards the top of the page. The object configuration description is complete and can be obtained from a survey perspective. By having this, the observer does not have to change position or orientation.

The reference object in figure 6.2a) is object *1* and the relationship of object *2* to object *1* is therefore (*2 RF 1*) which is read as: object *2* is *RIGHT FRONT* of object *1*. Of course, the configuration can be described by using object *2* as the reference object as well. The description would then be (*1 LB 2*) meaning that object *1* is *LEFT BACK* of object *2*. In figure 6.2b) object *1* is again chosen as a reference object and the relationship of object *2* to object *1* is (*2 RF 1*). The relationship of object *3* to object *1* is (*3 LF 1*). In the latter case though, there is not enough information to describe the scene unambiguously because nothing is known about the relationship of object *2* to object *3*. To describe the configuration completely, one of the relationships (*3 LF 2*) and (*2 LB 3*) needs to be given. This means that the reference object has to be changed for example to object *2*, which has been done in figure 6.2c). Thus the relationship (*3 LF 2*) can be obtained. The reference object could be changed to object *3* to obtain the relationship (*2 RB 3*) respectively. A possible *ocd* for the configuration in the figures 6.2b) and 6.2c) is therefore {(*2 RF 1*), (*3 LF 1*),(*3 LF 2*)}. In total eight different complete *ocd*s are possible for an object configuration of three objects, where the relationships of each object to every other object need to be provided. Note, that by reversing the relationships these *ocd*s can be converted into each other. For the configuration shown in figure 6.2b) and 6.2c) the eight alternatives are:
ocd_1={(*1 LB 2*), (*1 RB 3*), (*2 RB 3*)},
ocd_2={(*1 LB 2*), (*1 RB 3*), (*3 LF 2*)},
ocd_3={(*1 LB 2*), (*3 LF 1*), (*2 RB 3*)},
ocd_4={(*1 LB 2*), (*3 LF 1*), (*3 LF 2*)},
ocd_5={(*2 RF 1*), (*1 RB 3*), (*2 RB 3*)},
ocd_6={(*2 RF 1*), (*1 RB 3*), (*3 LF 2*)},
ocd_7={(*2 RF 1*), (*3 LF 1*), (*2 RB 3*)},
ocd_8={(*2 RF 1*), (*3 LF 1*), (*3 LF 2*)}).

Any object configuration consisting of n objects is completely described by all $n^2 - n$ object relationships. However, this number may be argued to be unnecessarily large, since the relationship (A/B) already contains the information of the reversed relationship (B/A). If, for instance, object A is *RIGHT FRONT* of object B then object B can be nowhere else than *LEFT BACK* of object A. Therefore, $\frac{n^2-n}{2}$ relationships are sufficient. For instance the configuration in figure 6.3a) can be described by the *ocd*: {(2 RF 1), (3 RN 1), (4 RF 1), (3 RB 2), (4 RF 2), (4 RF 3)}.

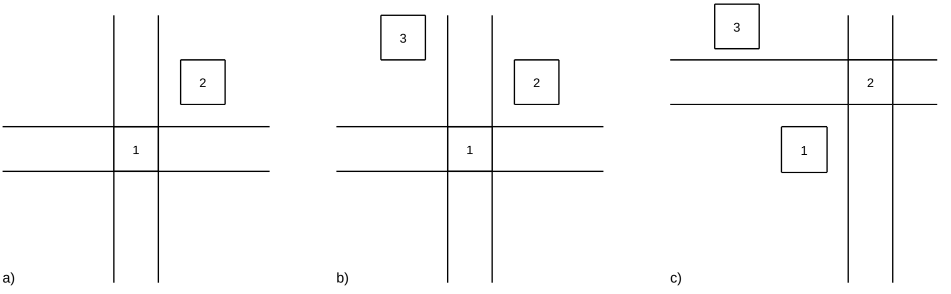


Figure 6.2: *Obtaining an ocd by describing object positions in relation to a reference object. a) (2 RF 1) meaning that object 2 is RIGHT FRONT of object 1. b) and c) Three relationships are needed to describe this configuration unambiguously. Reference object 1 in b) gives the relationships (2 RF 1), (3 LF 1). The reference object has been changed in c) to object 2 to obtain the relationship (3 LF 2).*

6.3.1 Problems encountered during object configuration description

The following example illustrates some problems that arise in the context of description and reconstruction of an object configuration. Herein the listener (reconstructor) is thought of as a person. One alternative is that he reconstructs the relationships successively in the order they are provided and that he accepts that the reconstructed picture might need to be rearranged when further information is obtained. At some points during the reconstruction process he might be forced to decide on an object's position and an object's relationships to other objects, knowing, that these decisions might be wrong and might have to be retracted when further information arrives. How these problems are omitted within the QuaDRO representation scheme follows in the following sections.

Reconstructing the configuration from the previous *ocd*, {(2 *RF* 1), (3 *RN* 1), (4 *RF* 1), (3 *RB* 2), (4 *RF* 2), (4 *RF* 3)} considering the entries, one at the time in the order they are presented leads to uncertainties about the object's positions. For the first relationship (*2 RF 1*) only one solution is possible. Already the second relationship (*3 RN 1*), however, leads to three different possible positions for object *3* that are consistent with the part of the *ocd* that has been processed so far. The next relationship (*4 RF 1*), leaves us with 36 different possible configurations. The fourth relationship, (*3 RB 2*) brings some clarification and eliminates 22 configurations where object *3* is not *RIGHT BACK* of object *2*. The following relationship (*4 RF 2*) eliminates an additional 11 configurations and the final relationship (*4 RF 3*) chooses between the remaining three possibilities; leaving only the correct solution.

To keep up to 36 possible configurations in mind, or to manually reconstruct them, is cognitively difficult for a person and therefore unacceptable for spontaneous human-machine dialogue. Besides, people tend even in mental models not to keep all possibilities open simultaneously. Instead, they decide on one concrete possibility, the preferred mental model [Kna95; Jah04; Rau05] and rearrange this when further information demands it. One possible alternative to reconstruct the configuration from the *ocd* in question is illustrated in figure 6.3b) to 6.3f). The position of the objects *1* and *2* from the relationship (*2 RF 1*) is unambiguous. When object *3* is mentioned in the relationship (*3 RN 1*), however, nothing is known about object *3*'s relationship to object *2*. Therefore, the reconstructor has to place object *3* somewhere in the area *RIGHT NEUTRAL* of object *1* and has to keep in mind that this relationship is already definite, whereas *3*'s relationship to object *2* can, however, still be changed. The same problem appears for object *4* that is known to be *RIGHT FRONT* of object *1* when the entry (*4 RF 1*) is reached, while its relationships to objects *2* and *3* are still open. Figure 6.3c) is consistent with all the information known so far regarding the *ocd*, but unfortunately not with the original configuration shown in figure 6.3a).

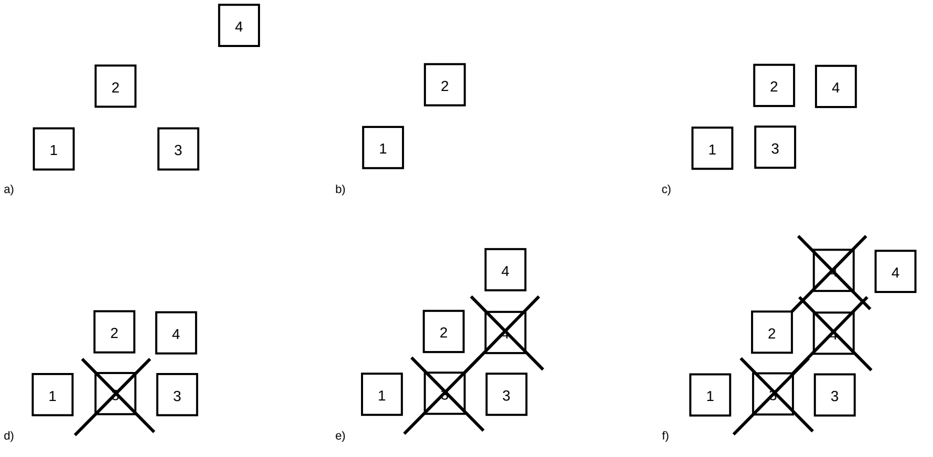


Figure 6.3: *A possible chain of intermediate configurations during the reconstruction from the OCD: {(2 RF 1), (3 RN 1), (4 RF 1), (3 RB 2), (4 RF 2), (4 RF 3)}. a) The original configuration. b) The intermediate configuration after the relationship (2 RF 1) has been processed. c) One, out of 36 possible configurations after the relationships (3 RN 1) and (4 RF 1) have been dealt with. d) The relationship (3 RB 2) makes the reconstructor reorder the objects. e) The reconstructor has to reorder again after the relationship (4 RF 2) came to his attention. f) The final picture after the last relationship (4 RF 3) has been processed, is qualitatively equivalent to the original configuration.*

The next *ocd* entry, (*3 RB 2*), clarifies that object *3* needs to be moved. It can be moved to the position shown in figure 6.3d), which again is consistent with the *ocd* processed so far but not with the original configuration, and it has to be remembered that *3*'s relationships to *1* and *2* are definite but the others are not. The next entry, (*4 RF 2*), makes the reconstructor reconsider object *4*'s position, which might lead to the result shown in figure 6.3e). Thereafter, the last entry clarifies object *4*'s final position. At last the resulting picture, given in figure 6.3f), is consistent with the original object configuration. When the *ocd* provides a complete description of the object configuration, the order in which the object relationships are presented influences the degree of uncertainty. Well ordered *ocd*s avoid retracting already placed objects. For incomplete *ocd*s another strategy is used, that is to allocate as much space for an object as needed to fulfill every possible, so far unknown relationship, to any other object. The reconstruction process for incomplete *ocd*s is described further in chapter 9.

6.3.2 Well ordered ocds

```
Algorithm 1. buildWellOrderedOCDfromSurveyPerspective
=========================================================
input:
 observedObjectList (A sequence of observed objects)

output:
 ocd (a sequence of ocd_entries)
---------------------------------------------------------
describedObjectList <- getFirstObject
                        (observedObjectList)
observedObjectList <- removeFirstObject
                        (observedObjectList)
for all entries in the observedObjectList do
 target <- current observedObjectList entry
 for all entries in the describedObjectList do
  create a new ocd_entry
  ocd_entry.reference <- the current object from the
                          describedObjectList
  ocd_entry.relation <- the relation the
                         target object to the
                         current object from the
                         describedObjectList
  ocd_entry.target <- target
  add ocd_entry to ocd
 end for
 add target to describedObjectList
end for
return ocd
---------------------------------------------------------
```

The problem of reconsideration of objects' relationships, described in this section can be avoided under certain conditions. If all relationships of an object needed to locate it correctly are known before the object is inserted into the reconstruction this problem does not occur. If, in other words, the reconstructor waits until the complete *ocd* has been transmitted and can be reordered, or searched for the appropriate entry when needed there is no problem. An alternative solution is that the observer provides *well ordered ocd*s using algorithm 1 'buildWellOrderedOCDfromSurveyPerspective'. In the preferred order, all relationships of a newly introduced object to all previously mentioned objects are given (in the inner for loop, line 5 to 11) before the next object is introduced (in the outer for loop, line 3 to 13). Using this algorithm saves time and effort during the reconstruction process as it allows the reconstructor to place an object correctly into the scene. He never has to reconsider its qualitative relationships to the other objects.

The input to algorithm 1 is a sequence of all objects observed from the observer's current position. At the moment it is assumed that the observer has a complete overview of the object configuration and all objects that have to be described are visible at once. The algorithm provides an *ocd* which is a sequence of triples of the form <target object, qualitative relation, reference object>. An ocd_entry therefore consists of the slots ocd_entry.target, ocd_entry.relation, and ocd_entry.reference. Each object that has been described as a sequence of ocd_entries is removed from the set of observed objects (observedObjectList) and added to the set of described objects (describedObjectList).

6.4 The QuaDRO grid

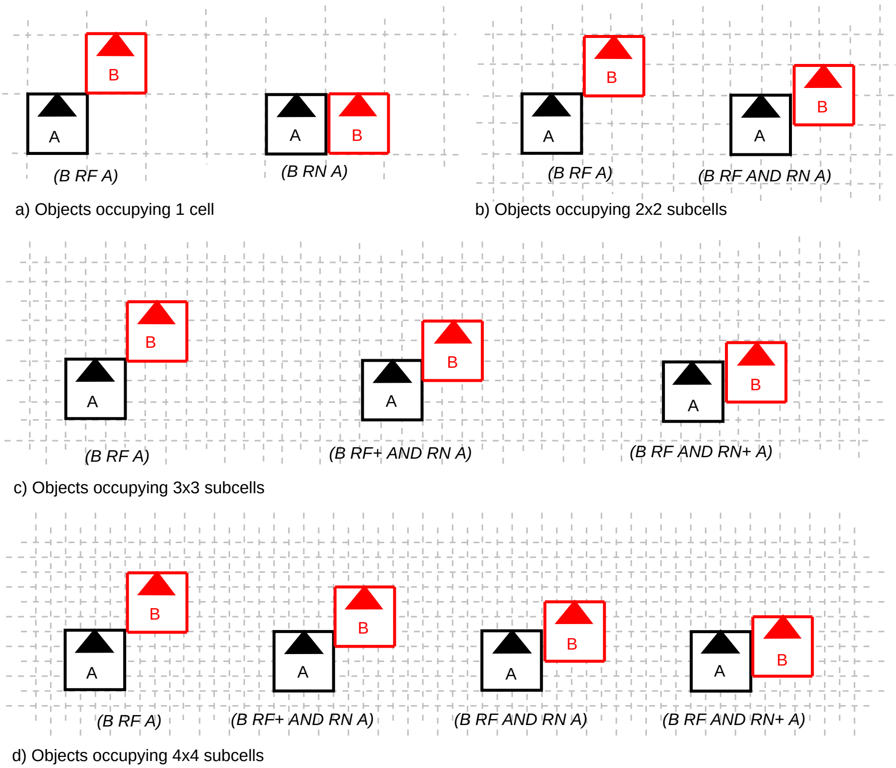


Figure 6.4: *The QuaDRO grid. Objects are represented by nxn (nϵN) (sub)cells. a) n=1 b) n=2 c) n=3 d) n=4.*

The reconstruction process produces a structure over a two-dimensional coordinate system with adjustable granularity, called the QuaDRO grid. The advantages of adjustable granularity is that adjustments can be made for example to the observer's perception capabilities and that the minimal resolution can be used when providing sufficient results in order to save computational effort [Dyl06; Ren04]. This grid is regarded to be qualitative as no metric is applied on it. Therefore, no information about the distance between two objects occupying neighboring cells is available.

The underlying data structure is a Cartesian coordinate system. An object is represented by nxm ($n, m \in \mathbb{N}$) cells. The concrete number of cells representing an object depends on its relationships to other objects and on the available knowledge about these relationships. For the sake of simplicity, throughout this and the following two chapters, the objects are all supposed to be of the same size and represented by nxn ($n \in \mathbb{N}$) cells. Chapter 9 uses objects occupying nxm ($n, m \in \mathbb{N}$) cells. The cell denoted by (x,y) is defined as in [Rag03] by the coordinates (x,y), (x+1, y), (x, y+1), (x+1, y+1). Subcells are defined accordingly.

QuaDRO's expressibility varies with the granularity of the underlying grid. If only the nine basic relations (*SF*, *RF*, *RN*, *RB*, *SB*, *LB*, *LN*, *LF*) are to be expressed, one grid cell for each object is sufficient (n=1). If overlapping relations, such as *RIGHT FRONT AND RIGHT NEUTRAL*, are needed the grid's granularity needs to be increased. An object that was previously represented by one cell, sufficient to distinguish the nine basic relations, must now be represented by four subcells (n=2). This representation allows a distinction between sixteen combined relations.

If the grids granularity represents an object by nine subcells (n=3) 49 relations are distinguishable, which may express where the bigger part of an object is. Figure 6.4c) shows, among other things, the relationship (*B RIGHT FRONT*$^+$ *AND RIGHT NEUTRAL A*) where the bigger part of object *B* is in the *RIGHT FRONT* region indicated by $^+$. In natural language, this could, for instance, be formulated as: 'Object *B* is to the right front and to the right of object *A*, but its bigger part is to the right front.' For n=4, 81 relations make it possible to express where the bigger part of an object is or if the object is equally distributed in the neighboring regions. The relationships (*B RIGHT FRONT A*), (*B RIGHT FRONT*$^+$ *AND RIGHT NEUTRAL A*), (*B RIGHT FRONT AND RIGHT NEUTRAL A*), and (*B RIGHT FRONT AND RIGHT NEUTRAL*$^+$ *A*) are shown in figure 6.4d).

No matter how many relations are distinguished they are all based on the nine basic relations that represent the eight regions around the minimal bounding box of the reference object and the region of the reference object itself. All other relations express cases where an object is distributed between neighboring regions and can be described by the combination of the regions' names where the object is at least partly inside. This allows observers and listeners to learn only the names of the nine basic relations and to combine them in the way needed, thus easily handling them in case of a change in the grid's granularity.

This section described the QuaDRO reconstruction grid with adjustable granularity in general. However, in order to achieve a more straightforward reconstruction process with less relationships to overlook and to concentrate on the elements of the reconstruction process, the remainder of this thesis only deals with the basic relationships.

6.5 The reconstruction process

An object configuration described by an *ocd* is successively reconstructed into the QuaDRO grid. Definition 6.1 formally describes a reconstructed object configuration.

Definition 6.1 ***Configuration***

A configuration in QuaDRO is an eleventuple
Conf = $\langle$O, P, L, OL, G, S, R, OR, OCD, RSD, GCD$\rangle$ where

- O is a finite set of objects.
- P is a mapping from O to the objects' handles or names.
- L is a (finite or infinite) set of locations.
- OL is a mapping from O to L, providing the objects' locations.
- G is a finite set of binary predicates, describing the regions in the underlying global frame of reference.
- S is a finite set of binary predicates, describing the regions in the objects' intrinsic frames of reference. The observer uses the relations in S within the object configuration description (ocd).
- R is a finite set of unary predicates, describing orientations.
- OR is a mapping from O to R, describing the objects' orientations.
- OCD is a subset of O$\times$S$\times$O, using the intrinsic frames of reference. The OCD contains the part of the ocd, that has been reconstructed.
- RSD is a finite set of triples of the form $<$o, OL(o), OR(o)$>$: o $\in$ O.
- GCD is a subset of O$\times$G$\times$O called the global configuration description, using the global frame of reference.

Definition 6.2 ***simple Configuration***

A simple configuration is a configuration Conf = $\langle$O, P, L, X, OL, G_8, S_8, R_1, OR_1, OCD, RSD, GCD$\rangle$ where

- O a finite set of objects.
- P is a mapping from O to $\mathbb{N}^+$ reflecting the order in which the objects have been encountered.
- L $\subseteq \mathbb{N}^+ \times \mathbb{N}^+$, representing the set of points in the Cartesian plane with integer coordinates.
- X $\subseteq$ L $\times$ L = {<x, y>, <x', y'> : x=x' $\wedge$ y=y'} is the equality relation.
- OL is a mapping from O to L with OL(o)=<x, y> : o $\in$ O.
- G_8 is a finite set of binary predicates: {N, NE, E, SE, S, SW, W, NW}.
- S_8 is the set of binary predicates: {SF, RF, RN, RB, SB, LB, LN, LF}.
- R_1 is the unary predicate north, describing an object's orientation.
- OR_1 is a mapping from O to R_1.
  $\forall$ o $\in$ O : OR_1(o) = north.
- OCD is a subset of O$\times S_8\times$O the object configuration description.
- RSD is a finite set of tuples of the form <o, OL(o), OR_1(o)> : o $\in$ O.
- GCD $\subseteq$ O$\times G_8 \times$O the global configuration description.

The set O describes the objects in the configuration. Each object can be addressed by its handle or name that is assigned to it by the mapping P. The set L defines all locations at which objects can be placed. An object's location is described by the mapping OL from the object to its location. The set G defines the qualitative regions used to describe an object's position within the underlying global frame of reference for instance *NORTH*, *EAST*, etc. The observer uses the set S of qualitative intrinsic orientations for instance *STRAIGHT FRONT*, *RIGHT NEUTRAL*, etc. The mapping OR assigns an orientation out of the set R to an object. In order to place an object whose position was described by intrinsic relationships given by the set S into the global frame of reference of the reconstruction area, the objects' global relationships to its reference objects have to be calculated. The OCD contains all ocd entries (provided by the observer) that have been reconstructed using the intrinsic relations. In contrast to that, the GCD contains the corresponding global object relationships. Finally the RSD describes all triples of objects, their concrete positions within the QuaDRO reconstruction area and their orientations within the reconstruction.

A configuration containing only orientationless objects is called a simple configuration and described in definition 6.2. In a simple configuration the objects' handles are natural numbers, the locations are given in a Cartesian coordinate system with integer coordinates. The set G_8 of global relations (definition 6.3) and the set S_8 of intrinsic relations (definition 6.4) are used. The relation *SAME* is excluded, which is especially defined by the equality relation X, in order to prevent situations where more than one object is at the same position. R_1 is a set of orientations, which for simple configurations only contains the orientation north. R_1 is subset of R_8. Both are presented in definition 6.5. The mapping OR_1 maps orientations out of the set R_1 to the objects.

Definition 6.3 G_8 **- *Global relations***

The set G_8 of global spatial relations is a finite set of binary predicates over locations <x,y>,<x',y'> : x, x', y, y' $\in \mathbb{N}^+$ in the Cartesian plane:

N, NE, E, SE, S, SW, W, NW, where:

<x',y'> N <x,y> iff (x' = x) $\wedge$ (y' < y)
<x',y'> NE <x,y> iff (x' > x) $\wedge$ (y' < y)
<x',y'> E <x,y> iff (x' > x) $\wedge$ (y' = y)
<x',y'> SE <x,y> iff (x' > x) $\wedge$ (y' > y)
<x',y'> S <x,y> iff (x' = x) $\wedge$ (y' > y)
<x',y'> SW <x,y> iff (x' < x) $\wedge$ (y' > y)
<x',y'> W <x,y> iff (x' < x) $\wedge$ (y' = y)
<x',y'> NW <x,y> iff (x' < x) $\wedge$ (y' < y)

Definition 6.4 S_8 **- Intrinsic relations**

The set S_8 of intrinsic spatial relations is given by:

{SF, RF, RN, RB, SB, LB, LN, LF}.

Definition 6.5 R_m **- Orientations**

The sets R_m of orientations with parameter m $\in \{1, 4, 8\}$ where $R_1 \subset R_4 \subset R_8$ is defined as follows:

$R_1 = \{\text{north}\}$

$R_4 = \{\text{north, east, south, west}\}$

$R_8 = \{\text{north, northeast, east, southeast, south, southwest, west, northwest}\}$

A reconstruction is said to be OCD-consistent, when an object configuration description of the reconstruction, using the set S of intrinsic relations, is equal to the relationships of the OCD component. Comparing to the scenario described in section 6.2, the OCD represents the reconstructed elements of the input ocd_1. The RSD represents the object's positions and orientations within the reconstruction. ocd_2 is then an object configuration description using the intrinsic relationships of S of the reconstruction over the reconstruction itself. If the configuration is OCD consistent then $ocd_1 = ocd_2$ under the premiss that the objects are introduced in the same order in both cases. Formally OCD-consistency is defined in definition 6.6

Definition 6.6 ***OCD-consistency***

A configuration Conf = $\langle$O, P, L, OL, G, S, R, OR, OCD, RSD, GCD$\rangle$ is OCD-consistent

iff $\forall$ <o, r, o'> $\in$ OCD : <OL(o), r, OL(o')>

6.5.1 Reconstruction algorithms

The reconstruction is done in polynomial time. It starts with algorithm 2 'reconstructObjectConfigurationFromOCD' that takes the *ocd* provided by the observer as input and provides a corresponding RSD, describing all object's concrete positions and orientations within the reconstruction. On the basis of the RSD the graphical representation is established. Furthermore, the algorithm provides a GCD describing the object's relations mentioned within the *ocd* in the global frame of reference. It further fills the OCD-component with all spatial relations from the input *ocd* that have been reconstructed.

The first object can be placed anywhere within the reconstruction area. At this point, QuaDRO places it in the middle of the initially shown part of it, which by default is the quantitative position 7,7 (line 3-6). The object's orientation is set to north (line 7). For every newly introduced object in the *ocd* an rsd_entry is created. Such an entry consists of the object's x-y-position in the coordinate system (rsd_entry.x, rsd_entry.y), the object's handle (rsd_entry.object) and the object's orientation (rsd_entry.orientation). The location for the following objects are calculated by algorithm 3 'placeNextObject' (line 16) whose input is the target object, the position of which has to be calculated, the ocd from which the relationships that have to be considered are taken, and the RSD that is used to know where the objects that are qualitatively described in the ocd, are situated within the reconstruction. The output of algorithm 3 is an x,y-coordinate where the target object has to be placed. This is assigned to the rsd_entry in line 17 and 18 and the object's orientation is set to north (line 19).

Furthermore, algorithm 2 translates the intrinsic object relationships, given in the *ocd* into global object relationships, stored in the GCD. A gcd_entry consists of a target object (gcd_entry.target), a reference object (gcd_entry.reference) which are the same as in the corresponding ocd_entry, and a relation (gcd_entry.relation) which is the translation of the intrinsic relation presented in the corresponding ocd_entry, provided by algorithm 5 'SG' called in line 24.

Algorithm 3 works as follows: Before any relationship is known an object can be anywhere within the reconstruction area. Therefore, the whole reconstruction area becomes the target object's region (*targetRegion* in line 1). The reference relation is the first relation of the target object to be considered. The variable *referenceRelation* keeps this value for later use (line 2). All entries in the *ocd* that describe a relation of the target object are considered next. In the for loop (line 3-14) each such entry is considered. Its reference object (*referenceObject*) and the relation between target and reference object (*relation*) is needed. The reference region (*referenceRegion*) is the region that the target object is allowed to be in according to its relation to the reference object. It is quantitatively specified by the function 'calculateReferenceRegion', described in algorithm 4. The reference region has to be intersected with the target region in order to calculate the new target object's region. This is done by the function 'intersect', which is straight forward and therefore not explicitly provided. In case the resulting region becomes too small to hold another object, it is broadened using the function 'moveObjects'. Dependent on the dimension in which more space is needed this function calls the algorithm 'splitHorizontally' (algorithm 6) or 'splitVertically' that are described further later in this chapter. After that the two steps of calculating the reference region and intersecting are repeated. The for loop (line 3 to 14) ensures that each relationship that follows that considers the same target object is intersected with the previously calculated target region. When the final target region is found, the target object's location is given by the function 'insertObjectUsingDistanceDefault', which according to the object's relationship to the first provided reference object (*referenceRelation*), calculates the object's concrete position within the reconstruction. This function straight forwardly, applies the distance default from definition 6.9 and is therefore not explicitly presented.

```
Algorithm 2. reconstructObjectConfigurationFromOCD
==========================================================
input:
 ocd (a sequence of ocd_entries)

output:
 RSD (a sequence of rsd_entries)
 GCD (a sequence of gcd_entries)
 OCD (a sequence of ocd_entries that
      have been reconstructed)
 ---------------------------------------------------------
    if ocd != {}
     then
      create new rsd_entry
      rsd_entry.x <- 7
      rsd_entry.y <- 7
      rsd_entry.object <- 1
      rsd_entry.orientation <- north
      add rsd_entry to rsd
    end if
    for all entries in the ocd do
     if current ocd_entry.target
        = previous ocd_entry.target
      then %
      else
       create new rsd_entry
       rsd_entry.object <- current ocd_entry.target
       position <- placeNextObject
                   (current ocd_entry.target, ocd, RSD)
       rsd_entry.x <- position.x
       rsd_entry.y <- position.y
       rsd_entry.orientation <- north
       add rsd_entry to RSD
       create new gcd_entry
       gcd_entry.target <- ocd_entry.target
       gcd_entry.referenc <- ocd_entry.reference
       gcd_entry.relation <- SG(ocd_entry, RSD)
       add gcd_entry to GCD
       add ocd_enry to OCD
     end if
    end for
    return RSD, GCD, OCD
----------------------------------------------------------
```

```
Algorithm 3. placeNextObject
==========================================================
input:
 ocd_entry (the target object has to be placed)
 ocd (The object configuration description)
 RSD (The reconstructed scene description)

output:
 targetPosition (The x,y coordinate for the target)
----------------------------------------------------------
target <- ocd_entry.target
targetRegion <- the whole reconstruction area
referenceRelation <- SG(ocd_entry, RSD)
for all entries in the ocd that have target
       as target object do
 referenceObject <- the chosen
                           ocd_entry.reference
 relation <- the chosen ocd_entry.relation
 repeat
  referenceRegion <- calculateReferenceRegion
                      (chosen ocd_entry, RSD)
  targetRegion <- intersect
                    (referenceRegion, targetRegion)
  if targetRegion is empty
   then moveObjects
         (referenceObject, RSD, relation)
  end if
 until targetRegion is not empty
 add the chosen ocd_entry to OCD
end for
targetPosition <- insertObjectUsingDistanceDefault
                    (targetRegion, referenceRelation)
return targetPosition
----------------------------------------------------------
```

```
Algorithm 4. calculateReferenceRegion
=========================================================
input:
 ocd_entry
 RSD

output:
 referenceRegion
---------------------------------------------------------
    referencePosition <- getPosition
                          (ocd_entry-reference, RSD)
    referenceOrientation <- getOrientation
                              (ocd_entry.reference, RSD)
    globalRelation <- SG(ocd_entry, RSD),
                              referenceOrientation)
    referenceRegion <- all <x,y> that are in the
                        globalRelation following
                        definition 6.3
    return referenceRelation
---------------------------------------------------------
```

```
Algorithm 5 SG
=========================================================
input:
 ocd_entry
 RSD

output:
 globalRelation
---------------------------------------------------------
    intrinsicRelation <- ocd_entry.relation
    orientation <- getOrientation(ocd_entry.reference,
                                   RSD)
    globalRelation <-  NS(((SN(orientation)
                          + SN(intrinsicRelation))
                            mod 8), G)
    return globalRelation
---------------------------------------------------------
```

Definition 6.7 SN

The mapping SN assigns numbers $\in \mathbb{N}$ to the members of the sets S_8, G_8, R_8 and D_8.

SN(SF) = 0	SN(N) = 0	SN(north) = 0	SN(sf) = 0
SN(RF) = 1	SN(NE) = 1	SN(northeast) = 1	SN(rf) = 1
SN(RN) = 2	SN(E) = 2	SN(east) = 2	SN(rn) = 2
SN(RB) = 3	SN(SE) = 3	SN(southeast) = 3	SN(rb) = 3
SN(SB) = 4	SN(S) = 4	SN(south) = 4	SN(sb) = 4
SN(LB) = 5	SN(SW) = 5	SN(southwest) = 5	SN(lb) = 5
SN(LN) = 6	SN(W) = 6	SN(west) = 6	SN(ln) = 6
SN(LF) = 7	SN(NW) = 7	SN(northwest) = 7	SN(lf) = 7

Definition 6.8 NS

The mapping NS takes a number and a type out of the set {S, G, R, D} and provides the string representing the respective relation out of the named set.

NS(0, S) = SF	NS(0, G) = N	NS(0, R) = north	NS(0, D) = sf
NS(1, S) = RF	NS(1, G) = NE	NS(1, R) = northeast	NS(1, D) = rf
NS(2, S) = RN	NS(2, G) = E	NS(2, R) = east	NS(2, D) = rn
NS(3, S) = RB	NS(3, G) = SE	NS(3, R) = southeast	NS(3, D) = rb
NS(4, S) = SB	NS(4, G) = S	NS(4, R) = south	NS(4, D) = sb
NS(5, S) = LB	NS(5, G) = SW	NS(5, R) = southwest	NS(5, D) = lb
NS(6, S) = LN	NS(6, G) = W	NS(6, R) = west	NS(6, D) = ln
NS(7, S) = LF	SN(7, G) = NW	NS(7, R) = northwest	NS(7, D) = lf

---

Algorithm 4 'calculateReferenceRegion' uses the ocd_entry and the RSD to calculate the reference region. According to the reference object's location within the reconstruction (line 1), its orientation (line 2), and the global relationship of reference and target object (line 3), the reference region is calculated in line 4 following definition 6.3.

'SG' (algorithm 5) translates the relationships using intrinsic relations out of the set S_8 into global relationships using relations out of the set G_8. In order to easily do so the relations from S_8, G_8, and R_8 are matched to natural numbers by the mapping SN described in definition 6.7. Its counterpart is the mapping NS that takes a number and an indicator out of the set {S, G, R, D} and matches the number onto the string that represents the relation.

Figure 6.5 illustrates the process for the configuration given in figure 6.3a), described by the well ordered *ocd*: {(*2 RF 1*), (*3 RN 1*), (*3 RB 2*), (*4 RF 1*), (*4 RF 2*), (*4 RF 3*)}. Object *1* is placed with its orientation to the top of the page (north). The first *ocd* entry allows the placing of *2 RIGHT FRONT* of object *1*, as done in figure 6.5a). All relationships concerning object *3* follow and can therefore be handled together. Object *3* is *RIGHT NEUTRAL* of object *1* which is somewhere within the grey area in figure 6.5b). So far it is not clear exactly where the object will be in relation to object *2* but the next entry, (*3 RB 2*), resolves the problem and restricts its possible position to the intersection of the areas *RIGHT BACK 2* and *RIGHT NEUTRAL 1* as shown in figure 6.5c). Object *3* can be placed, anywhere within the remaining area, which is done in figure 6.5d).

Object *4*, is next to be inserted into the picture in the area *RIGHT FRONT* of object *1* (figure 6.5e)) which is narrowed down to its intersection with the area *RIGHT FRONT* of object *2* (figure 6.5f)). The resulting area is narrowed down further to its intersection with the area *RIGHT FRONT* of object *3* shown in figure 6.5g). Within the remaining area, object *4* can be placed anywhere. Figure 6.5h) shows the result that qualitatively conforms with the original. Furthermore, every intermediate state during the reconstruction is also qualitatively correct in comparison with the original object configuration.

6.5.2 The distance default

If the configuration is considered as an allocation of objects on a two-dimensional grid, then for a given set of input relationships an infinite number of configurations that are consistent with the input exist, since it is possible to place the objects arbitrarily far away from each other. The reason for this is that no relation is included which limits the distance between the objects. Therefore, a preference is imposed on the possible configurations where objects are adjacent to each other, or as close to each other as possible as defined in definition 6.9. The distance default is motivated by two grounds: (1) the grid is regarded as being qualitative with no metric considering distance assumed, and (2) the default appears to result in reasonable answers to the queries that can be put to the system.

Furthermore, the application of this default is incremental. As the first relationship is received, a configuration is constructed that is consistent with that relationship and with the default distance. Additional input relationships accumulate information to the configuration, but they may also make it necessary to retract a previously applied distance in order to obtain a configuration that conforms to the given input.

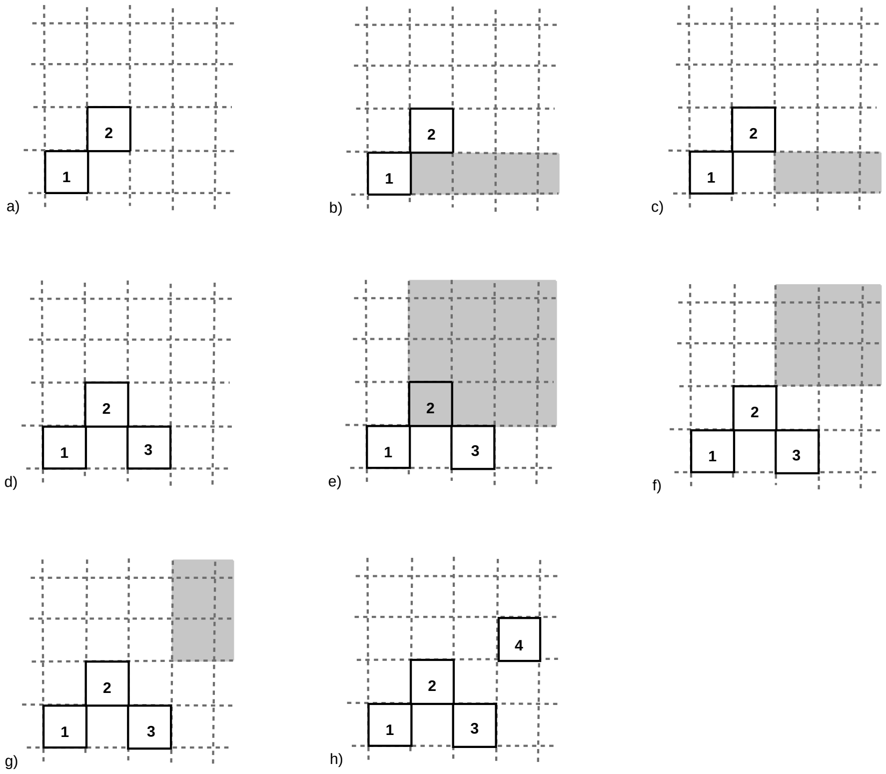


Figure 6.5: *The reconstruction from a well ordered ocd. A place for an object is first completely determined before the next object is taken into account. a) The reconstruction after the relationship (2 RF 1). b) (3 RN 1) identifies the so far possible space for object 3. c) (3 RB 2) narrows the possible space down to the intersection of RIGHT NEUTRAL 1 and RIGHT BACK 2. d) Object 3 is placed. e) (4 RF 1) gives the, so far possible, space for object 4. f) It is narrowed down to its intersection with RIGHT FRONT 2 after the relationship (4 RF 2) has been read. g) shows the spaces remaining after the last relationship (4 RF 3) has been taken into account. f) Finally the configuration is fully reconstructed and qualitatively equivalent to the original.*

Definition 6.9 ***Distance default***

The distance default for configurations using the set G_8 of the eight global relations *N*, *NE*, *E*, *SE*, *S*, *SW*, *W*, and *NW* defines the insertion point <x', y'> for a new object (o'), depending on its relationship, r_1, to the first reference object considered.

Let x* = the set of all integers (for the x coordinate)
Let y* = the set of all integers (for the y coordinate)

if r_1 is *N*	then x'=x*,	y'=Max(y*)
if r_1 is *NE*	then x'=Min(x*),	y'=Max(y*)
if r_1 is *E*	then x'=Min(x*),	y'=y*
if r_1 is *SE*	then x'=Min(x*),	y'=Min(y*)
if r_1 is *S*	then x'=x*,	y'=Min(y*)
if r_1 is *SW*	then x'=Max(x*),	y'=Min(y*)
if r_1 is *W*	then x'=Max(x*),	y'=y*
if r_1 is *NW*	then x'=Max(x*),	y'=Max(y*)

As long as only similar sized objects occupying exactly one cell are considered x* only contains one element for the relationships *N* and *S*. Respectively contains y* only one element for the relationships *E* and *W*.

6.5.3 Obtaining space between objects

Even when the *ocd* is well ordered and the distance default is applied during reconstruction, sometimes a number of objects need to be moved to create space where new objects can be merged into the scene. It is not an option to leave some space between the objects in the reconstruction in order to be prepared to insert further objects in between. If the number of objects to be inserted is unknown from the beginning, the space might be too small, regardless.

Moving objects in order to obtain some space is, for example, the case in the following *ocd*: {(*2 RF 1*), (*3 RF 1*), (*3 RN 2*), (*4 RF 1*), (*4 RF 2*), (*4 SF 3*), (*5 RF 1*), (*5 RN 2*), (*5 LN 3*), (*5 LB 4*)}. The reconstruction works as described above. Object *1* is placed somewhere in the scene, and object *2* is placed *RIGHT FRONT* of it accordingly. Thereafter the area for object *3* is obtained by intersecting the areas *RIGHT FRONT* of object *1* and *RIGHT NEUTRAL* of object *2*, and so object *3* is placed there. Object *4* follows after intersecting the regions *RIGHT FRONT* of object *1*, *RIGHT FRONT* of object *2* and *STRAIGHT FRONT* of object *3*. The result is shown in figure 6.6a).

Continuing with object *5*, the intersection of the areas *RIGHT FRONT* of object *1*, *RIGHT NEUTRAL* of object *2*, *LEFT NEUTRAL* of object *3*, and *LEFT BACK* of object *4*, is too small to hold another object. Obviously, object *5* needs to be placed to the left of objects *3* and *4* and to the right of objects *1* and *2*. Therefore, the objects in the reconstruction are divided into two groups as indicated by a line in figure 6.6b). Algorithm 6 'splitHorizontally' performs a

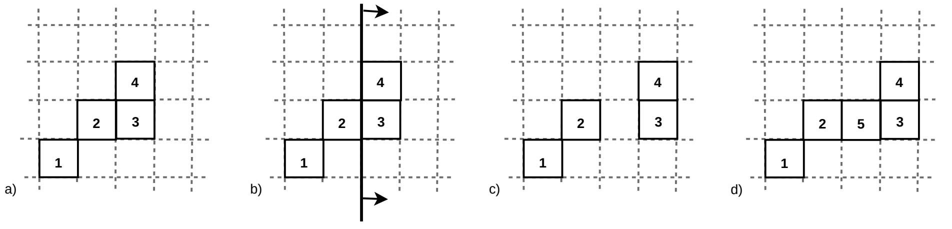


Figure 6.6: *Making horizontal space between the objects, in order to merge object 5 into the reconstruction. a) The reconstructed scene before the procedure. b) The line indicates where space has to be made, the objects in the picture are separated into two groups one of each side of the line. c) The object groups have been moved apart from each other. d) Object 5 has been placed into the newly obtained space.*

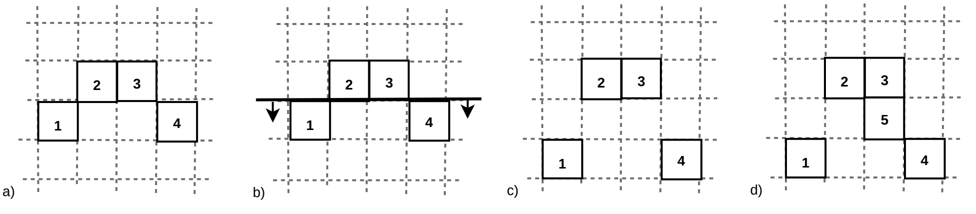


Figure 6.7: *Making vertical space between the objects to merge in object 5. a) The configuration before the procedure. b) The vertical line indicates where the objects are separated. c) The object groups have been moved apart from each other. d) Object 5 has been placed at its correct position.*

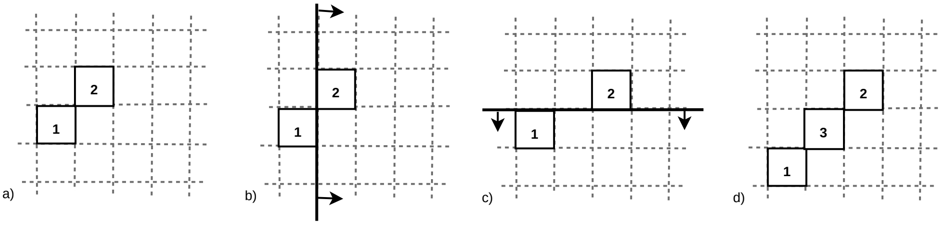


Figure 6.8: *Making space in both dimensions. a) The reconstructed configuration before the procedure is applied. b) The vertical line shows where the objects are separated to obtain horizontal space. c) Horizontal space has been obtained and the horizontal line now shows where the objects are separated to obtain vertical space. d) The sufficient space has been acquired and object 3 has been filled in.*

horizontal split of the reconstruction. It moves the groups, as they are, apart from each other in the horizontal dimension until enough space for one more object, pictured in figure 6.6d), is acuired. At last, object *5* is placed into the newly created space which is shown in figure 6.6e) and ends the reconstruction process for the given *ocd*. Formally the horizontal split is described in definition 6.10. Algorithm 6 takes the RSD and an object_n after that the configuration has to be split as input. Its output is a modified RSD (RSD') describing the reconstructed configuration after the split. The algorithm works straight forward. Every object whose x-coordinate value is greater than object_n's x-coordinate value is moved one step in increasing x-dimension, no other objects are moved.

Due to the projection based frame of reference with neutral zone, dividing the space around the reference object into eight equivalence classes using axis-parallel lines, a separation of the objects into two groups by an axis-parallel line and moving the object groups apart does not influence the objects' relationships.

Definition 6.10 ***Horizontal split***

Let Conf = $\langle$O, P, L, X, OL, G_8, S_8, R_m, OR_m, OCD, RSD, GCD$\rangle$ be a simple configuration. The horizontal split of Conf to the right of n (n $\in \mathbb{N}^+$) is defined as Conf' = $\langle$O, P, L, X, OL', G_8, S_8, R_m, OR_m, OCD, RSD', GCD$\rangle$
where OL' is obtained from OL as follows:

$\forall$ o $\in$ O : OL(o) = <x, y> $\wedge$ x $\leq$ n $\Rightarrow$ OL'(o) = <x, y>
$\forall$ o $\in$ O : OL(o) = <x, y> $\wedge$ x > n $\Rightarrow$ OL'(o) = <x+1, y>

RSD' is the set of triples of the form
<o, OL'(o), OR_m(o)> : <o, OL(o), OR_m(o) > $\in$ RSD

The process to obtain space in vertical dimension, the vertical split formally described in definition 6.11, works accordingly and is illustrated by the example in figure 6.7. The *ocd* that describes the object configuration is: {(*2 SF 1*), (*3 RF 1*), (*3 RN 2*), (*4 RN 1*), (*4 RB 2*), (*4 RB 3*) (*5 RF 1*), (*5 LB 2*), (*5 SB 3*), (*5 LF 4*)}, space in vertical dimension is needed to place object *5*. The same approach as before is applied. Object *5* needs to be behind objects *2* and *3* but in front of objects *1* and *4* therefore, the objects in the reconstruction are divided into two groups, one containing *2* and *3* the other containing *1* and *4* and the groups, as they are, are moved vertically apart from each other. New space appears exactly where needed, to place object *5*. The vertical split and the horizontal split of a simple configuration for an arbitrary splitting point results in a new simple configuration, as is expressed in proposition 6.1. For the correct interpretation of the vertical split it must be realized that the reconstruction area's coordinate system originates at the top left corner. Therefore, a vertical split below n results in moving objects into an increasing y-dimension.

```
Algorithm 6. splitHorizontally
==========================================================
input:
 RSD (describing all objects' positions
      within the reconstruction)
 object_n (The object from that on all objects with
           a higher x-position have to be moved.)

output:
 RSD (after the objects' positions have been
      changed)
 ---------------------------------------------------------
   for all entries in the RSD do
    if rsd_entry.x > object_n.position.x
     then rsd_entry.x <- object.position.x + 1
    end if
   end for
   return RSD
----------------------------------------------------------
```

Definition 6.11 ***Vertical split***

Let Conf = $\langle$O, P, L, X, OL, G_8, S_8, R_m, OR_m, OCD, RSD, GCD$\rangle$ be a simple configuration. The vertical split of Conf below n, (n $\in \mathbb{N}^+$) is defined as Conf' = $\langle$O, P, L, X, OL', G_8, S_8, R_m, OR_m, OCD, RSD', GCD$\rangle$ where OL' is obtained from OL as follows:

$\forall$ o $\in$ O : OL(o) = $<$x, y$>$ $\wedge$ y $\leq$ n $\Rightarrow$ OL'(o) = $<$x, y$>$
$\forall$ o $\in$ O : OL(o) = $<$x, y$>$ $\wedge$ y $>$ n $\Rightarrow$ OL'(o) = $<$x, y+1$>$

RSD' is the set of triples of the form
$<$o, OL'(o), OR_m(o)$>$: $<$o, OL(o), OR_m(o)$>$ $\in$ RSD

Proposition 6.1

If a configuration Conf is OCD-consistent, then both a vertical split and a horizontal split of Conf for an arbitrary n is also a simple configuration, and it is OCD-consistent.

Proof. The intuitive proof for the horizontal split is as follows: At the beginning, the reconstruction correctly contains all objects' relationships. Objects will only be moved in increasing x-dimension and will therefore never cross any reference frame line that separates regions horizontally. For every object that is unmoved and situated *W*, *NW*, or *SW* of the moved object. The moved object is *NE*, *E*, or *SE* of it. These regions are infinite to the east and the object will never leave them by moving eastwards. All other objects are moved in the same way as the first-hand object and therefore its relationships to these objects do not change. ■

The proof for the vertical split is analogous.

Sometimes it is necessary to obtain new space in both dimensions simultaneously. In this case, the two algorithms 'splitHorizontally' and 'splitVertically' are applied sequentially. This applies when reconstructing from the *ocd*: {(*2 RF 1*) (*3 RF 1*), (*3 LB 2*)}. Figure 6.8a) shows the reconstruction after the first relationship. Thereafter, object *3* has to be inserted right and in front of object *1* and left and back of object *2*. The vertical line in figure 6.8b) shows where the objects are moved apart to gain some space in the horizontal dimension. The result of this procedure, together with the horizontal line that illustrates where the objects are to be separated to obtain new space in the vertical dimension, is shown in figure 6.8c). After these two operations, object *3* can be placed into the new space, as shown in figure 6.8f). When well ordered *ocd*s are used, these three ways of providing space are sufficient. Rearranging the objects' relative positions is never necessary, as all qualitative relationships are correct all along the reconstruction process.

6.5.4 The optimal order of objects

The order in which the objects are introduced can reduce the reconstruction effort even more. In the last example in figure 6.8, the objects need to be moved apart horizontally and vertically to obtain some space for a new object. If the *ocd* introduces the objects in a different order, for instance as: {(*3 RF 1*), (*2 RF 1*), (*2 RF 3*)} no such action is needed. An optimal order of objects is therefore well ordered, and does not require the making of space in between already placed objects. Starting the description with an object in the middle of the observed object configuration and continuing helically to the outside is one possible strategy

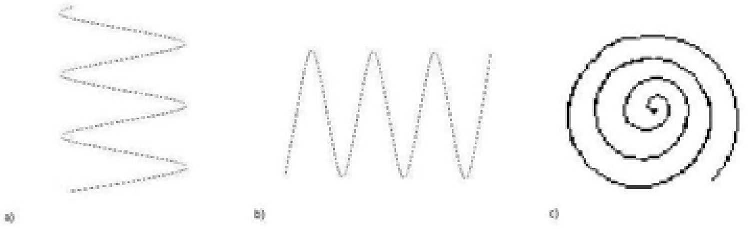


Figure 6.9: *Some description sequences that lead to well ordered ocds.*

Figure 6.10: *Influencing Objects. a) The configuration can be described by two relationships: (2 RF 1) and (3 RF 2). If these two relationships are known the relationship (3 RF 1) does not add any further information. 1 is no influencing object of 3. b) In this configuration all three relationships are necessary. If (2 LF 1) and (3 RF 1) are known the relationship between 2 and 3 is still unclear. 2 is an influencing object of 3.*

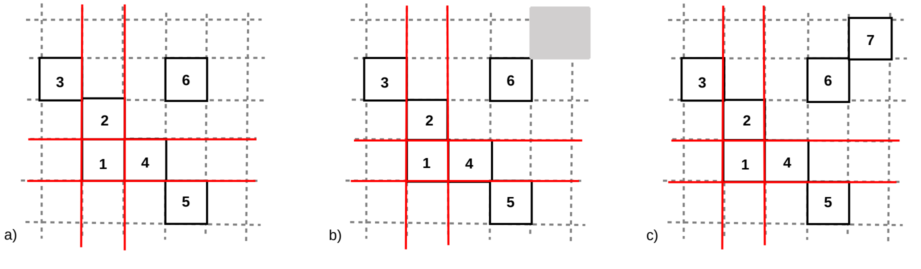


Figure 6.11: *Influencing objects. a) A configuration of six objects. b) Only the relationship (7 RF 6) is required to correctly insert object 7. Once this relationship is known none of the other five objects influence object 7's position any further. c) The completed configuration.*

for an optimal order. Another strategy is to start at a point at the border or the corner of the scene and to describe the objects from there to the opposite border or corner. Figure 6.9 shows some intuitive description sequences leading to optimal orders.

However, it might not always be possible to obtain optimally ordered *ocd*s from the observer. The optimality might depend on the observer's skills and on the infrastructure of the observation area. Furthermore, for instance, if the objects are the result of biological or chemical processes, new objects might appear after the description process has been started and are added at the end of the *ocd*.

6.5.5 Influencing objects

Depending on the configuration, it might be possible to find a suitable description without calculating all $\binom{n}{2}$ relationships. The configuration of three objects shown in figure 6.10a) could be described by the three relationships (*2 RF 1*), (*3 RF 1*) and (*3 RF 2*) but not all these relationships are necessary. The relation *RF* is transitive and therefore the two relationships (*2 RF 1*), and (*3 RF 2*) are sufficient. The region *RIGHT FRONT* of object *2* is equivalent to the intersection of the regions *RIGHT FRONT* of object *1* and *RIGHT FRONT* of object *2*. Therefore, the relationship (*3 RF 1*) does not influence object *3*'s position any further, given that the relationships (*2 RF 1*) and (*3 RF 2*) are already known.

However, for the configuration shown in figure 6.10b), all three relationships, (*2 LF 1*), (*3 RF 1*) and (*3 RN 2*), are necessary. Suppose that the objects *1* and *2* are already in the scene and object *3* has to be placed *RIGHT FRONT* of object *1*. Before it can be inserted, its relationship to object *2* needs to be known as this relationship will narrow down the space for its position to one of the intersections of *RIGHT FRONT* of object *1* with *RIGHT FRONT* of object *2*, *RIGHT NEUTRAL* of object *2*, or *RIGHT BACK* of object *2*. That means that object *2* influences object *3*'s position and is therefore called an *influencing object* of *3*.

In general, influencing objects can be found in regions of the same reference object that are *in-line* with the region where the new object has to be placed and within the region itself. In figure 6.10b), object *3* has to be placed *RIGHT FRONT* of object *1*. Horizontally in-line with the region *RIGHT FRONT* are the regions *STRAIGHT FRONT* and *LEFT FRONT*. Object *2* is situated in region *LEFT FRONT* and is therefore an influencing object of object *3*. A region is horizontally in-line, when it is reachable just by moving horizontally from the region where the new object has to be inserted (in this case *RIGHT FRONT* of object *1*). The regions *LEFT FRONT*, *STRAIGHT FRONT* and *RIGHT FRONT* are horizontally in-line with each other. For objects orientated to the top of the page, the regions *RIGHT NEUTRAL*, *SAME*, and *LEFT NEUTRAL* are horizontally in-line, and so are the regions *RIGHT BACK*, *STRAIGHT BACK*, and *LEFT BACK*. A region is vertically in-line when it is reachable just by moving vertically from the region to place the object. The regions *RIGHT FRONT*, *RIGHT NEU-*

TRAL, and *RIGHT BACK*, and respectively the regions *STRAIGHT FRONT*, *SAME*, and *STRAIGHT BACK* and the regions *LEFT FRONT*, *LEFT NEUTRAL*, and *LEFT BACK* are vertically in-line.

In figure 6.11a) object *7*, (*7 RF 1*), is to be put *RIGHT FRONT* of object *1*, which means that all other objects in the picture are influencing objects. When the next given relationship is (*7 RF 6*), the number of remaining influencing objects immediately reduces to zero, as shown in figure 6.11b). If the next relationship had instead been (*7 RF 2*), four of the previously five influencing objects would have remained. Considering influencing objects while formulating the *ocd* makes it possible to provide *ocd*s with less than $\binom{n}{2}$ relationships. The previous example shows that the number of relationships needed to describe the configuration decreases with every influencing object that is taken into account. It is therefore of advantage to first choose influencing objects that have a strong impact on the number of remaining influencing objects.

6.6 Preserving OCD-consistency

The purpose of the reconstruction algorithm presented above is to successively extend a simple configuration with additional objects while preserving OCD-consistency.

If the *ocd* provided by the observer from which the reconstruction is carried out and given as input to algorithm 2, is not empty, algorithm 2 starts with placing the first object (o_1), which itself is not mentioned in the *ocd*, into the reconstruction area (line 1 to 8). The resulting configuration $Conf_1 = \langle o_1$, P(o_1), L, X, OL(o$_1$), G_8, S_8, R_1, $OR_1(o_1)$, $\emptyset$, <o$_1$, OL(o_1), $OR_1(o_1)$>, $\emptyset\rangle$ obviously is OCD-consistent, since the OCD component so far considered in the reconstruction is empty.

The loop starting in line 10 of algorithm 2 begins with this OCD-consistent configuration. For each new object o' introduced in the *ocd*, algorithm 3 'placeNextObject' is called in line 16 of algorithm 2. So far the configuration has not been changed, therefore, algorithm 3 starts with an OCD-consistent configuration.

Generally, every time the algorithm is called it starts with an OCD-consistent configuration Conf = ⟨O, P, L, X, OL, G_8, S_8, R_1, OR_1, OCD, RSD, GCD⟩. In line 1 the target region (*targetRegion*) for the new object o' is set to the whole reconstruction area. At this moment no relationships of object o' to other objects have been considered.

Let the reconstruction area be defined by the values X_{min}, X_{max}, Y_{min}, and Y_{max}. The target region therefore contains all locations <x,y>:(X_{min} ≤ x ≤ X_{max}) ∧ (Y_{min} ≤ y ≤ Y_{max}). Let it be defined by:
$X_{min_{target}}$=X_{min}, $X_{max_{target}}$ =X$_{max}$, $Y_{min_{target}}$=Y_{min}, $Y_{max_{target}}$=Y_{max}.

By the loop starting in line 3, a set (let it be called N) of triples of the form <o', r, o_i> describing the relationships of the new object o' that is not yet a member of O, to objects $o_i \in$ O, P(o_i) = i, (previously inserted objects) is considered.

Each iteration of the for loop takes the relation r of the new object o', which is called the target, to the provided reference object o_i (*referenceObject*) and calls the function 'calculateReferenceRegion' which returns the *referenceRegion* according to o_i's location within the reconstruction and the relation r using definition 6.3. The resulting referenceRegion is the area defined by $X_{min_{reference}}$, $X_{max_{reference}}$, $Y_{min_{reference}}$, $Y_{max_{reference}}$.

In line 8 the targetRegion is intersected with the referenceRegion and only those locations that remain are:
$\{<x,y>: \mathrm{Max}(X_{min_{target}}, X_{min_{reference}}) \leq x \leq \mathrm{Min}(X_{max_{target}}, X_{max_{reference}}) \wedge \mathrm{Max}(Y_{min_{target}}, Y_{min_{reference}}) \leq y \leq \mathrm{Min}(Y_{max_{target}}, Y_{max_{reference}})\}$.

In case this intersection is empty the function 'moveObjects' is called in line 10. This function, depending on the situation, either calls 'splitVertically' or 'splitHorizontally' both of which return modified configurations but according to the proof provided in proposition 6.1 preserve OCD-consistency. Thus if line 10 has been executed, the previous steps in line 6 to 12 are repeated on a modified but OCD-consistent configuration.

If the target region is not empty, obviously all locations in this area are the only locations that previously belonged to both, *referenceRegion* and *targetRegion*. The *referenceRegion* calculated using definition 6.3 ensures that all positions in the remaining *targetRegion* conform to the relation r $\in$ <o', r, o_i>.

Further iterations of the for loop (line 3 to 14) lead to a target region that conforms to all spatial relations provided for object o'. When the loop finishes in line 14, the target position is chosen by using the distance default (definition 6.9) out of all locations within the *targetRegion*. Therefore the chosen location conforms to all relations considered. Algorithm 3 ends by returning this location to algorithm 2, the only changes that algorithm 3 does to the configuration is done by a horizontal or vertical split which both preserve OCD-consistency. Therefore after algorithm 3 has terminated the configuration remains OCD-consistent.

Algorithm 2 adds the <o' OL(o'), OR_1(o')> to the RSD (line 17 to 20), which changes the configuration by adding the new object o' at its position within the grid. As this position is the intersection of all relations r considered and iteratively calculated it follows from definition 6.3 that the resulting configuration is OCD-consistent. Formally the new configuration Conf' is given by:

Conf' = $\langle$O', P', L, X, OL', G_8, S_8, R_1, OR_1', OCD', RSD', GCD'$\rangle$, where

- O'= O $\cup$ o' The new object o' joins the set of objects
- P' = P $\cup$ P(o').
- OL' = OL $\cup$ OL(o').
- OR_1' = $OR_1 \cup OR_1$(o')
- OCD' = OCD $\cup$ N The OCD joint by the triples in N.
- RSD' = RSD $\cup$ <o', OL(o'), OR_1(o')>.
- GCD' = GCD $\bigcup_{all}$ <o', SG(r), o_i> : <o', r, o_i > $\in$ N.

This chapter introduced the representation scheme QuaDRO, defined many of the technical terms used, presented the most essential algorithms applied, and showed with several examples how a configuration of objects without orientations, described by a complete *ocd* is reconstructed. The following chapter takes QuaDRO's expressibility one step further and introduces the additional techniques used to handle objects with intrinsic orientations.

Chapter 7

Description and reconstruction of a configuration of objects with orientation

In the previously described processes for reconstructing a configuration of objects which lack individual orientations, the intrinsic frame of reference was globally oriented the same for each object. The region *STRAIGHT FRONT* was pointing to the top of the page and the other regions were distributed accordingly. This equals the use of an absolute frame of reference. In other words, the orientation assigned to all objects within the reconstruction has been *north*. As described in chapter 4, an absolute frame of reference can be changed into an intrinsic frame of reference in consideration of the object's orientation. For objects that are aligned with the QuaDRO grid this change clearly allows for four axis-parallel orientation classes where an object's front is to one of its edges. The description and reconstruction processes adjusted to four object orientations are described in section 7.1. In section 7.2 the challenges of broadening the processes to eight object orientations are introduced. The approach made in QuaDRO to handle eight orientations and still use the simple procedures presented in chapter 6 for objects without orientations and in section 7.1 for objects with four axis-parallel orientations, with just small adjustments, is presented in section 7.3.

```
Algorithm 7.
buildOCDfromSurveyPerspectiveUsingFourOrientations
==========================================================
input:
 observedObjectList (A sequence of all observed objects)

output:
 ocd (A sequence of ocd_entries)
 ---------------------------------------------------------
 describedObjectList <- getFirstObject
                        (observedObjectList)
 observedObjectList <- removeFirstObject
                        (observedObjectList)

 for all entries in the observedObjectList do
  target <- current observedObjectList entry
  for all entries in the describedObjectList do
   create a new ocd_entry       //postion description
   ocd_entry.reference <- the current object from the
                          describedObjectList
   ocd_entry.relation <- the relation the
                         current target object has
                         to the current object from
                         the describedObjectList
   ocd_entry.target <- target
   add ocd_entry to ocd
  end for
  create a new ocd_entry    //orientation description
  ocd_entry.reference <- target
  ocd_entry.target <- the last handled object from
                      the describedObjectList
  ocd_entry.relation <- the relation of the last
                        handled object form the
                        describedObjectList to the
                        target object
  add ocd_entry to ocd
  add target to describedObjectList
 end for
 return ocd
----------------------------------------------------------
```

7.1 Objects with four axis-parallel orientations

Algorithm 7 'buildOCDFromSurveyPerspectiveUsingFourOrientations' shows that the description process for configurations of objects with four orientations works in the same way as for configurations of objects without orientation (line 1-11). $\binom{n}{2}$ relationships are needed to describe all objects' positions. The number can be reduced by taking only influencing objects into account and further by ordering the objects optimally. With n-1 further relationships, all objects' orientations can be described. In line 12 to 14 of algorithm 7, the orientation is given by a reversed relationship where the object to be inserted functions as the reference object and another object already introduced before is positionally described in relation to the first object. This is motivated by the scenario introduced in chapter 5. The observer does not keep track of his own orientation within a global frame of reference and therefore cannot use the object's absolute orientation. Moving from object to object on a route tour, he describes all object relationships he sees from the object at which he stands, using this object as the reference object. Assuming that object A is visible from object B if object B is visible from object A, the observer will automatically describe the relationships of A to B and B to A.

In figure 7.1a) object *2*'s position is described by (*2 RF 1*) but its orientation is unclear until the relationship (*1 RB 2*) is processed in figure 7.1b). When object *2* is *RIGHT FRONT* of object *1* and object *1* is concurrently *RIGHT BACK* of object *2*, object *2* must have an orientation where the area where object *1* is situated becomes *2*'s *RIGHT BACK* region. An object is completely described by all its relationships to the objects introduced before and a relationship to one of theses objects reversed for orientation information. If the observer is instructed to always present the relationship to the object he came from first, the orientation of the new reference object is given before further objects are introduced.

An object configuration using objects with the four axis-parallel orientations out of the set $R_4 = \{north, east, south, west\}$ is called a 4-oriented configuration and formally described in definition 7.1.

The reconstruction process presented in algorithm 8 'reconstructObjectConfigurationFromOCDusingOrientations' is in princiole similar to the previously described reconstruction using algorithm 2. The difference is that it distinguishes between *ocd* entries describing objects' positions and *ocd* entries describing objects' orientations. Besides which the objects are inserted into the grid, the distance default is applied and space is made in the horizontal or vertical dimensions, if needed, as before. In addition, the object's orientation is added by algorithm 9 'addOrientations' immediately after the object is placed into the reconstruction. To estimate an object's region and to finally insert the object into the reconstruction, the orientations of the first reference object and all influencing objects have to be considered. The algorithm 'placeNextObject', presented in chapter 6 is applied.

Algoritm 9 'addOrientations' takes an ocd_entry as input and calculates the orientation of this entry's target. Therefore it needs the further input of the ocd and the RSD. It uses the current ocd_entry and the next ocd_entry if this provides the reversed relationship of target and reference object. The orientation of the target object is calculated using the reversed relationship and the reference object's orientation. In order to do so the algorithm needs the mapping SN, defined in definition 6.7 that assigns a natural number to each intrinsic relation from the set S_8.

Definition 7.1. ***4-oriented Configuration***

A 4-oriented configuration is a configuration Conf = $\langle$O, P, L, X, OL, G_8, S_8, R_4, OR_4, OCD, RSD, GCD$\rangle$ where

- O a finite set of objects.
- P is a mapping from O to $\mathbb{N}^+$ reflecting the order in which the objects have been encountered.
- L $\subseteq \mathbb{N}^+ \times \mathbb{N}^+$, representing the set of points in the Cartesian plane with integer coordinates.
- X $\subseteq$ L $\times$ L = {<x, y>, <x', y'> : x=x' $\wedge$ y=y'} is the equality relation.
- OL is a mapping from O to L with OL(o)=<x, y> : o $\in$ O.
- G_8 is a finite set of binary predicates: {N, NE, E, SE, S, SW, W, NW}.
- S_8 is the set of binary predicates: {SF, RF, RN, RB, SB, LB, LN, LF}.
- OR_4 mapping from O to R_4.
- R_4 is the set of unary predicates: {north, east, south, west}.
- OCD is a subset of O$\times S_8 \times$O the object configuration description.
- RSD is a finite set of tuples of the form <o, OL(o), OR_4(o)>: o $\in$ O.
- GCD $\subseteq$ O$\times$G$\times$O the global configuration description.

```
Algorithm 8.
 reconstructObjectConfigurationFromOCDusingOrientations
==========================================================
input:
 ocd (a sequence of ocd_entries)

output:
 RSD (a sequence of rsd_entries)
 GCD (a sequence of gcd_entries)
 ---------------------------------------------------------
 if ocd != {}
  then
   create new rsd_entry
   rsd_entry.x <- 7
   rsd_entry.y <- 7
   rsd_entry.object <- 1
   rsd_entry.orientation <- north
   add rsd_entry to RSD
 end if
 for all entries in the ocd do

 if current ocd_entry.target
            = previous ocd_entry.target
  then %
 end if

 if (current ocd_entry.target
      = previous ocd_entry.reference)
 and
     (current ocd_entry.reference
      = previous ocd_entry.target)
  then %
  else
   create new rsd_entry
   rsd_entry.object <- current ocd_entry.target
   position <- placeNextObject
                 (current ocd_entry, ocd, RSD)
   rsd_entry.x <- position.x
   rsd_entry.y <- position.y
   rsd_entry.orientation <- addOrientation
                             (ocd_entry, ocd, RSD)
   add rsd_entry to RSD
   create new gcd_entry
   gcd_entry.target <- ocd_entry.target
```

```
     gcd_entry.referenc <- ocd_entry.reference
     gcd_entry.relation <- SG(ocd_entry.relation,
                                 RSD)
     add gcd_entry to GCD
    end if
   end for
  return RSD, GCD
----------------------------------------------------------
```

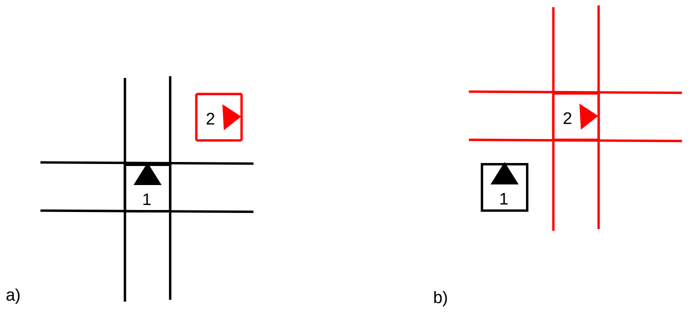


Figure 7.1: *Description of orientation. a) The OCD entry (2 RF 1) specifies the position of object 2 in relation to object 1. b) (1 RB 2) can only be the case if object 2 has the orientation shown.*

```
Algorithm 9. addOrientations
==========================================================
input:
 ocd_entry (ocd-entry.target's orientation is
            to be calculated)
 ocd
 RSD

output:
 orientation (the orientation of ocd_entry.target)
 ---------------------------------------------------------
    current_ocd_entry <- ocd_entry
    next_ocd_entry <- the entry from the ocd following
                      the current ocd entry
    if (next_ocd_entry.target
        = current_ocd_entry.reference)
    and
   (next_ocd_entry.reference = current_ocd_entry.target)
     then
      referenceOrientation <- getOrientation(the
                                 rsd_entry that
                                 corresponds to
                                 next_ocd_entry.taget)
      targetOrientation <- NS ((
                          (SN(next_ocd_entry.relation)
                           + SN(referenceOrientation))
                              mod 8), R)
    end if
    return orientation
----------------------------------------------------------
```

An example for four object orientations

An example of the processes is given for the configuration shown in figure 7.2. The observer introduces the objects in the order *1*, *2*, *3*, *4*, and *5*. He first describes the position of one object completely in relation to all other already described objects. He adds one reversed relationship for the object's orientation, which results in the *ocd*: {(*2 RN 1*), (*1 RN 2*)(*3 RF 1*), (*3 RB 2*), (*1 LB 3*), (*4 SF 1*), (*4 RB 2*), (*4 LB 3*), (*1 RN 4*), (*5 RF 1*), (*5 RB 2*), (*5 LB 3*), (*5 LF 4*), (*1 LF 5*)}.

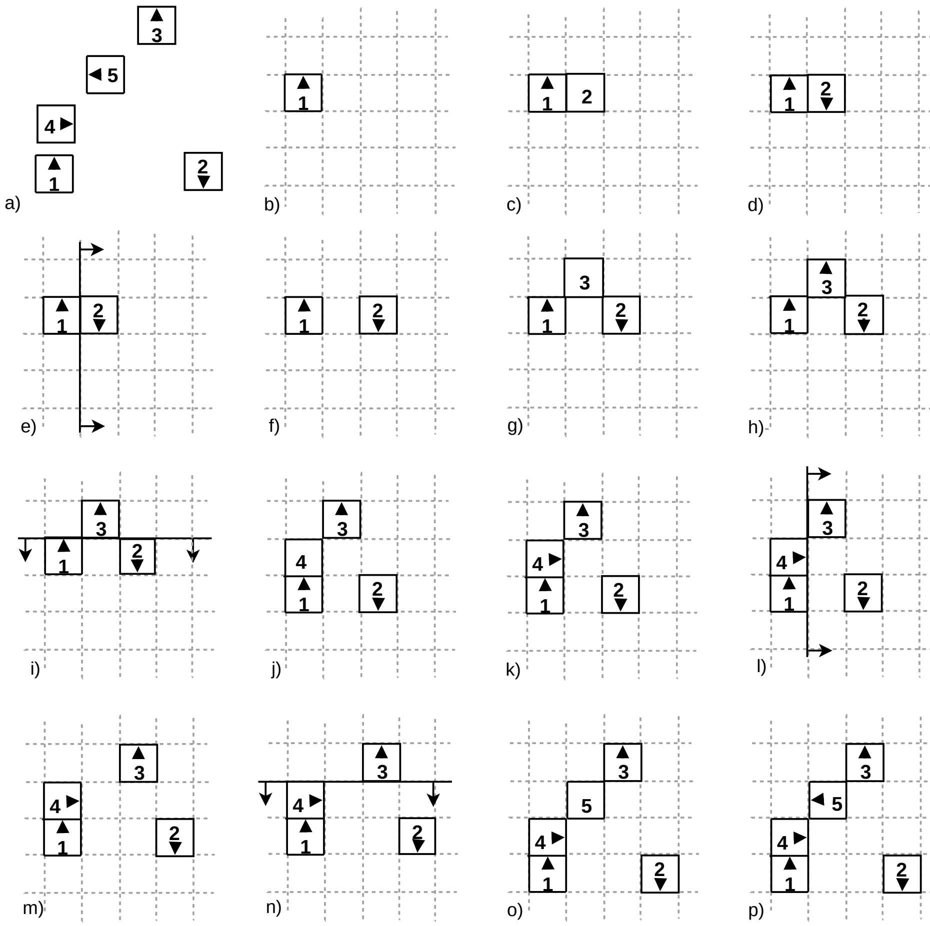


Figure 7.2: *The reconstruction process of the configuration of objects with orientations shown in a), demonstrated systematically in the pictures b) - p).*

The listener processes the information in the same order and reconstructs the scene systematically. First object *1* is placed into the scene (figure 7.2b)) and its orientation is set to *north*. As only relative positions are considered, the orientation attached to object *1* does not influence the result. Regardless of which orientation is assigned to the first object, all relationships in the scene will be the same as in the original in figure 7.2a). After object *1* has got its orientation object *2* (*2 RN 1*) can be placed directly (figure 7.2c)), and the relationship (*1 RN 2*) clarifies the orientation of object *2* as described above (figure 7.2d)). The process is continued as described before for orientationless objects, and in addition the orientation for each inserted object is added.

7.1.1 Preserving OCD-consistency for 4-oriented configurations

The proof of OCD-consistency for 4-oriented configurations is very similar to the proof of OCD-consistency for simple configurations. The difference between simple and 4-oriented configurations is the used set of orientations which is R_1 for simple configurations but R_4 for 4-oriented configurations. The reconstruction starts by placing the first object o_1 at position 7,7 and attaches the orientation north to it.

The result is the configuration $Conf_1 = \langle o_1$, P(o_1), L, X, OL(o$_1$), G_8, S_8, R_4, $OR_4(o_1)$, $\emptyset$, <o$_1$, OL(o_1), $OR_4(o_1)$>, $\emptyset\rangle$ that obviously is OCD-consistent, since the OCD component so far considered in the reconstruction is empty.

The reconstruction algorithm 'reconstructObjectConfigurationFromOCDusingOrientations' is nearly the same as algorithm 2 'reconstructObjectConfiguration'. The difference is that because of the additional ocd entries, containing orientation information, a second test (line 14) is done to filter out these ocd entries from those that describe object positions. This does not influence any objects' positions and has therefore no influence on OCD-consistency.

Generally, every time algorithm 3 is called it starts with an OCD-consistent configuration Conf = ⟨O, P, L, X, OL, G_8, S_8, R_4, OR_4, OCD, RSD, GCD⟩. In line 1 the target region (*targetRegion*) for the new object o' is set to the whole reconstruction area. At this moment no relationships of object o' to other objects has been considered.

The loop starting in line 3, considers a set (let it be called N) of triples of the form <o', r, o_i> describing the relationships of the new object o' that is not yet a member of O, to objects $o_i \in$ O, P(o_i) = i, (previously inserted objects) is considered.

Each iteration of the for loop takes the relation r of the new object o', which is called the target, to the provided reference object o_i (*referenceObject*) and calls the function 'calculateReferenceRegion' which returns the *referenceRegion* according to o_i's location within the reconstruction and the relation r using definition 6.3. The resulting referenceRegion is the area defined by:
$X_{min_{reference}}$, $X_{max_{reference}}$, $Y_{min_{reference}}$, $Y_{max_{reference}}$.

In line 8, the targetRegion is intersected with the referenceRegion and the only remaining locations are:

{<x,y>: Max($X_{min_{target}}$,$X_{min_{reference}}$) $\leq$ x $\leq$ Min($X_{max_{target}}$, $X_{max_{reference}}$) $\wedge$ Max($Y_{min_{target}}$,$Y_{min_{reference}}$) $\leq$ y $\leq$ Min($Y_{max_{target}}$, $Y_{max_{reference}}$)}.

In case this intersection is empty the function 'moveObjects' is called in line 10. Depending on the situation, this function either calls 'splitVertically' or 'splitHorizontally' both of which return modified configurations but according to the proof provided in proposition 6.1 preserve OCD-consistency. Thus, if line 10 has been executed, the previous steps in lines 6 to 12 are repeated on a modified but OCD-consistent configuration.

If the target region is not an empty set of positions, obviously all locations in this area are the only locations that previously belonged to both *referenceRegion* and *targetRegion*. The *referenceRegion* calculated using definition 6.3 ensures that all positions in the remaining *targetRegion* conform to the relation r $\in$ <o', r, o_i>.

Further iterations of the for loop (line 3 to 14) lead to a target region that conforms to all spatial relations provided for object o'. When the loop finishes in line 14, the target position is chosen by using the distance default (definition 6.9) out of all locations within the *targetRegion*. Therefore the chosen location conforms to all relations considered. Algorithm 3 ends by returning this location to algorithm 8, the only changes that algorithm 3 does to the configuration is done by a horizontal or vertical split which both preserve OCD-consistency. Therefore after algorithm 3 has terminated the configuration remains OCD-consistent.

Algorithm 2 adds the <o' OL(o'), OR_4(o')> to the RSD (line 20 to 23), which changes the configuration by adding the new object o' at its position within the grid. As this position is the intersection of all relations r considered and iteratively calculated following definition 6.7, definition 6.8 and definition 6.3 the modified configuration is OCD-consistent.

A further difference is that after the new object is in place, an orientation is attached to it wherefore algorithm 9 'addOrientation' is called. Adding an orientation to an object does not change any objects' positions and does not therefore influence OCD-consistency.

7.2 Objects with eight orientations

Considering that, in tasks of way finding or route description people distinguish often between eight direction or orientation classes [Kli03a], [Kli03b], it would be advantageous if eight orientation classes could be handled in QuaDRO without losing the simplicity of the description and reconstruction processes.

With only axis-parallel orientations, it is possible to split the reconstructed scene in x- or y-dimension and to move the resulting parts of the configuration apart from each other obtaining space between already represented objects. Besides axis-parallel moving of a group of objects, no rearrangement of objects is necessary.

Unfortunately, when objects at angles to the grid are included in the configuration the reconstruction process becomes more complicated. Reconstructing for instance the configuration described in the *ocd*: {(*2 RF 1*),(*1 SB 2*), (*3 RN 1*), (*3 RN 2*), (*2 SF 3*), (*4 SB 1*), (*4 RB 2*), (*4 LN 3*), (*3 LN 4*), (*5 LN 1*), (*5 LB 2*), (*5 LF 3*), (*5 SB 4*), (*1 SB 5*), (*6 RN 1*), (*6 RB 2*), (*6 LF 3*), (*6 LB 4*), (*6 SB 5*), (*2 SB 6*)}, only objects *1* to *5* can be placed following the same rules earlier introduced for objects with axis-parallel orientations. The reconstruction including these first five objects is shown in figure 7.3a).

Object *6* has to be inserted in the middle of the reconstruction to the *RIGHT NEUTRAL* of object *1* and to the *RIGHT BACK* of object *2*. Space to place object *6* correctly needs to be obtained. Dividing the plane horizontally and moving object *2* apart from the other objects provides enough space for object *6* within the intersection of the two regions. Unfortunately, this action influences the relationship of object *1* to object *2* and in order to preserve the relationship (*1 SB 2*) object *2* has to be moved one additional step to the right. Even if all other objects to the right of object *2* are moved in the same way, the result is not sufficient to preserve the relationships (*3 RN 2*) and (*2 SF 3*). Therefore, object *3* has to be moved one step further to the right. This, in turn, influences the relationship between the objects *3* and *4* and object *4* has to be moved two steps downwards to fulfill the relationships (*3 LN 4*) and (*4 LN 3*). The last change has an impact on the relationship of object *4* to object *5* and object *5* has to be moved two steps to the left in order to conform with the relationship (*5 SF 4*). Figure 7.3b) shows the resulting reconstruction with object 6 included.

This more complicated reconstruction process contains a source of error, as the objects have to be moved in different directions depending on the direction of movement of the previously moved object in combination with their relationship. It is easy to miss an influenced relationship. To avoid errors the listener would have to constantly check all relationships within the reconstruction in order to guarantee their continued validity.

Another problem occurs for objects that are in one of the regions *RIGHT FRONT*, *RIGHT BACK*, *LEFT FRONT*, and *LEFT BACK* of an object with an orientation at an angle to the grid. Using these relations as description of an objects position, the object's alignment information to other objects within the underlying absolute frame of reference is lost. For instance, object *2* in figure 7.4a) is described as being *RIGHT BACK* of object *1*. In the reconstruction, using cardinal directions, assuming that *north* is at the top of the page, this relation can be translated into object *2* being *southwest*, *south*, or *southeast* of object *1*. To distinguish between the three relationships a higher granularity of the reference frame is needed. The regions *RF*, *RB*, *LF*, and *LB* are to be subdivided into three different regions. Figure 7.4b) shows the result, which is a reference frame with sixteen positional relations. Even though the sixteen regions are named intuitively, this approach adds a lot more cognitive load to a human observer or listener. Firstly, because of their number, which exceeds a cognitive convenient number of relation classes, and secondly, due to an easy mix up between the

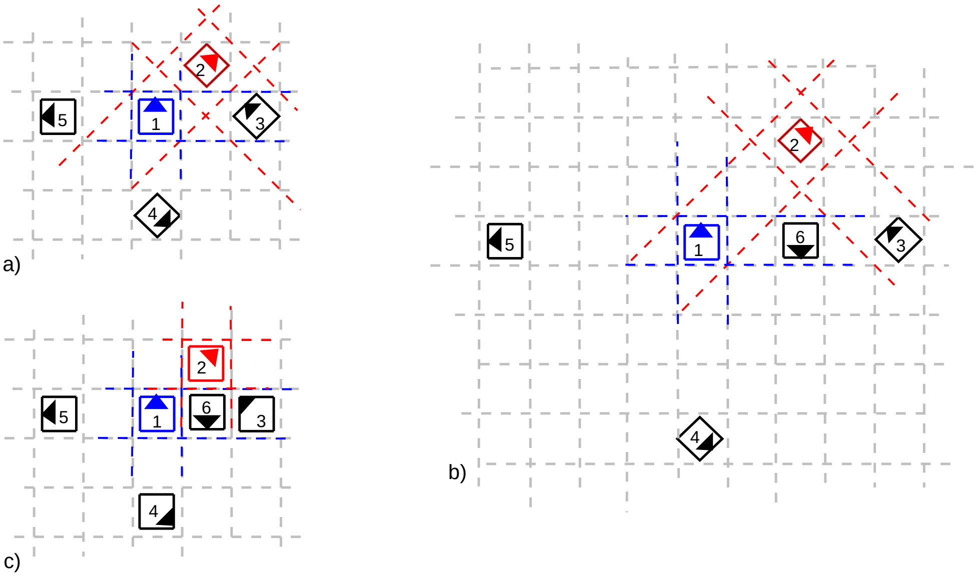


Figure 7.3: *Reconstruction of a configuration that includes objects at angles to the grid. a) A reconstruction of five objects where object 6 has to be placed RIGHT NEUTRAL of object 1 and RIGHT BACK of object 2. b) The same reconstruction after object 6 has been inserted. c) The configuration reconstructed using QuaDRO.*

regions. Even if the region were named differently, for instance with numbers, it is difficult for a human observer to accurately describe object relationships.

7.3 QuaDRO's approach to eight orientations

QuaDRO makes an approach to handle eight different orientations and still be able to use the simple strategies introduced for orientationless objects, to obtain space.

In order to use the simple strategies previously introduced all objects in QuaDRO are represented by rectangles that are aligned with the underlying global frame of reference. Furthermore, the projection-based frame of reference with neutral zone presented in figure 7.5a) is used. The object's front points to one of the rectangle's edges. For objects with orientations at angles to the underlying global frame of reference the frame of reference is rotated by 45° and the object's front points to one of the rectangle's corners as shown in figure 7.5b).

The latter approach originates from the following considerations. The projection based frame of reference with neutral zone with sixteen relationships, presented in figure 7.4b), for objects with orientations at angles to the grid, can be simplified by regrouping the qualitative regions. For example, the regions *Left*

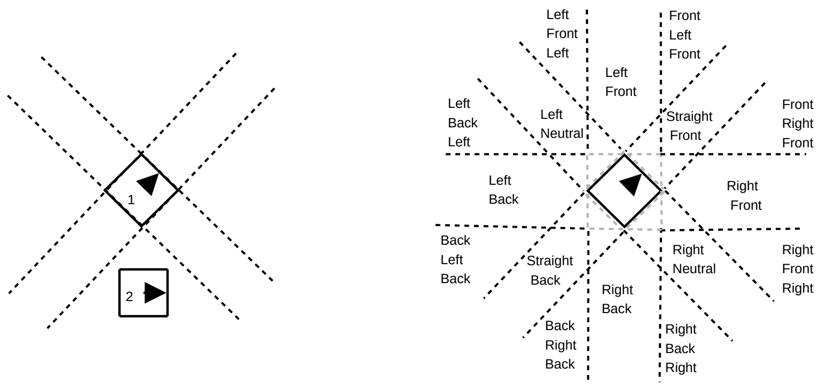


Figure 7.4: *Necessary relations to provide sufficient information for objects at angles to the grid. a) The relationship (2 RB 1) in a projection based intrinsic frame of reference for an object with orientation at an angle to the underlying grid. b) 16 positional relations that allow for an unambiguous distinction between eight orientations.*

Front Left, *Left Neutral*, and *Left Back Left* are grouped to *LN*, and the regions *Back Right Back*, *Straight Back*, and *Back Left Back* are grouped to *SB*. This results in the frame of reference shown in figure 7.5b) with eight different regions using the same intuitive names that have been used for objects at orientations aligned with the underlying absolute frame of reference.

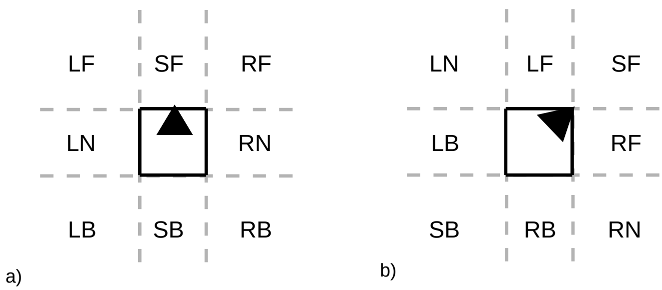


Figure 7.5: *QuaDRO's frame of reference. The regions around the reference object are named according to the object's intrinsic front. a) Projection based frame of reference with neutral zone. b) The frame of reference rotated by* 45°

Even though this approach might not seem very intuitive at first glance and adds an additional load onto the observer to recognize if an object has an overall aligned orientation or an orientation at an angle to the underlying absolute reference frame, and to switch between the two frames of references according to the object's orientations, it has the following advantages:

- All possible lines separating qualitative regions, are in x- or y-dimension and therefore allow the use of all techniques introduced in chapter 6 for objects without orientation and objects with just four axis-parallel orientations.

- An object's orientation can be described by the duality of its relationship and its reversed relationship to a reference object, using only the eight basic relations instead of sixteen relations as shown in the previous section or twelve relations as, for instance, is the case in TPCC [Mor03a] presented in chapter 4.

- The similar representation of all objects aligned with the grid allows the listener to represent objects with unknown orientation and to add the orientation when it becomes available without need to redraw the objects.

- The observer, as described in section 5, might use a pseudo intrinsic frame of reference, the objects' own intrinsic frame of reference or a momentarily applied absolute frame of reference to describe the objects' relationships. When an object's frame of reference is changed, its relationships to other objects change accordingly. Using the suggested types of frames of reference all objects that are in a certain relationship to the reference object within one frame of reference are in the same new relationship to the reference object after the frame of reference has been changed. For instance will all objects that are *LEFT FRONT* of a reference object be *LEFT NEUTRAL* when the reference object's orientation changes 45° clockwise.

Using this approach the *ocd* for the example above would be different according to the different reference frames. The relationships (*4 RB 2*), (*5 LB 2*), and (*6 LB 4*) given above are changed to (*4 SB 2*), (*5 SB 2*), (*6 LN 4*). During the reconstruction process, no object has to be moved and object *6* can be inserted directly into the picture, which has been done in figure 7.3c). The advantage using objects at angles is that more information about distances between objects can be gained from the qualitative configuration description. The example in figure 7.3 shows that objects had to be moved several steps apart from each other in order to fulfill certain relationships. However, by only using nine positional equivalence classes, the reconstruction might not represent the objects' alignment in the underlying absolute frame of reference. In contrast, the approach used in QuaDRO preserves the objects' alignment to the absolute frame of reference but leads to reconstructions that are more compact. Free space between objects is often automatically condensed.

Formally, an 8-oriented configuration is presented in definition 7.3. Besides the mapping R_8, presented in definition 7.5 that assigns one out of the set of eight orientations: {*north*, *northeast*, *east*, *southeast*, *south*, *southwest*, *west*, *northwest*} to an object, an 8-oriented configuration's definition is similar to the definition of simple and 4-oriented configurations. Because all objects represented are aligned with the underlying grid, the horizontal and the vertical split as defined in chapter 6 are applicable. The reconstruction process for 8-oriented configurations is done in polynomial time and the reconstruction provides OCD-consistency.

Definition 7.3 ***8-oriented Configuration***

An 8-oriented configuration is a configuration Conf = $\langle$O, P, L, X, OL, G_8, S_8, R_8, OR_8, OCD, RSD, GCD$\rangle$ where

- O is a finite set of objects.
- P is a mapping from O to $\mathbb{N}^+$ reflecting the order in that the objects have been encountered.
- $L \subseteq \mathbb{N}^+ \times \mathbb{N}^+$, representing the set of points in the Cartesian plane with integer coordinates.
- $X \subseteq L \times L = \{<x, y>, <x', y'> : x=x' \wedge y=y'\}$ is the equality relation.
- OL is a mapping from O to L with $OL(o)=<x, y> : o \in O$.
- G_8 is a finite set of binary predicates: {N, NE, E, SE, S, SW, W, NW}.
- S_8 is the set of binary predicates: {SF, RF, RN, RB, SB, LB, LN, LF}.
- R_8 is the set of unary predicates {north, northeast, east, southeast, south, southwest, west, northwest}.
- OR_8 mapping from O to R_8
- OCD is a subset of $O \times S_8 \times O$ the object configuration description.
- RSD is a finite set of tuples of the form $<o, OL(o), OR_8(o)> : o \in O$.
- GCD $\subseteq O \times G_8 \times O$ the global configuration description.

7.3.1 Preserving OCD-consistency for 8-oriented configurations

The proof of OCD-consistency for 8-oriented configurations is almost the same as the proof of OCD-consistency for 4-oriented configurations. The difference between 4-oriented and 8-oriented configurations is the used set of orientations which is R_4 for 4-oriented configurations but R_8 for 8-oriented configurations and the mapping OR_4 for. However, this does not influence the reconstruction algorithm and the same algorithm 'reconstructObjectConfigurationFromOCDusingOrientations' inclusive of all called functions is the same.

This chapter presented an innovative approach to handle eight different object orientations for a qualitative object configuration reconstruction in a global frame of reference from a complete object configuration description. In the next chapter the QuaDRO technical prototype is presented and some implemented examples are provided.

Chapter 8

The QuaDRO technical prototype

The QuaDRO representation scheme has been developed with the intention to alleviate and support human-machine communication in a dialogue situation. An observer (human or machine) in the field describes an object configuration from a survey perspective or from different local viewpoints. The description consists of binary object position relationships. The listener (human or machine) reconstructs the described configuration from the description into a global frame of reference (a piece of paper, computer screen, digitalization table). The purpose of this research is to develop a representation scheme that humans, autonomous agents or expert systems, can use to communicate about object relationships. Therefore, both sides of the process need to be cognitively easy for a person to understand and to fulfill. Figure 6.1a) and 6.1b) in chapter 6 show the high level flow of information between the two communication partners. The QuaDRO technical prototype has been implemented as a proof of concept for many of the developed methods and techniques. In figure 6.1c) the high level flow of information in the prototype is presented.

This chapter reassembles the processes realized within the technical prototype. Much of the information provided has been mentioned in previous chapters describing the representation system QuaDRO. The prototype consists of an observer process, operating in the observation area and a listener process, operating in the reconstruction area. Both areas are represented by grids of equal sized cells in which objects of different orientations can be placed. Observer- and listener processes use the QuaDRO representation scheme (frame of reference). The observer deploys it to generate an object configuration description (*ocd*), and the listener applies it to reconstruct the configuration from that *ocd*. The technical prototype for QuaDRO has been implemented in Java on the Haphazard virtual environment simulation engine [And05].

The observation area

The QuaDRO observation area contains a grid of adjustable length and width. The grid fields are rectangular and all of the same size. The version implemented so far considers only objects that are assumed to be of the same size and occupying one cell each. Thereby the granularity of the eight relations *SF*, *RF*, *RN*, *RB*, *SB*, *LB*, and *LF* can be achieved. The cells that represent the objects can be seen as equally sized and to the underlying Cartesian coordinate system, axis-parallel containers for real world objects. This abstracts from an object's shape and size but not from its two-dimensionality.

Objects without orientation and objects with eight different orientations can be represented. For ease of use the orientations are called *north*, *northeast*, *east*, *southeast*, *south*, *southwest*, *west* and *northwest*, assuming that the orientation *north* is facing the top of the observation area and the other orientations are represented accordingly. An object configuration is put into the grid by the *object configuration generator*, which produces a random configuration of objects each with one of the eight respective orientations, or a configuration of orientationless objects. Furthermore, several hand-coded examples exist that show the processes for specific object configurations.

The observer

Several observation strategies are implemented where the observer either has a survey perspective on the whole configuration or a smaller view field which forces him to move around in order to see all the objects. The observer can start his observation from any cell within the observation grid. It is further possible to implement alternative suitable strategies, either following a preprogrammed path, using one of the parsing strategies suggested in chapter 4, for instance, or pursuing a more reactive behavior.

The listener

The listener operates on the QuaDRO reconstruction area, a grid of adjustable size with equally sized cells into which the reconstructed object configuration has to be inserted. The basis for the reconstruction is given by the *ocd* only. If necessary, as is the case for the second observation alternative, the *ocd* is sorted into a well ordered *ocd*, whereby duplicate relationships are removed. Objects are inserted one by one into the reconstruction area, by calculating their position whilst considering influencing objects.

As long as no relationship concerning an object is known, the object can be anywhere within the reconstruction area. Assuming that the reconstruction area spans 30x30 coordinates, the preliminary values describing the region are: $prel\_x\_min = 1$, $prel\_x\_max = 30$, $prel\_y\_min = 1$, and $prel\_y\_max = 30$. This area is narrowed down according to the relationships to the reference- and influencing objects. If, for instance, the relationship to object *1* is *RIGHT FRONT* and object *1* is at the position (7;7) facing in decreasing y-dimension the new values bordering the area are: $prel\_x\_min = 8$, $prel\_x\_max = 30$, $prel\_y\_min = 1$, $prel\_y\_max = 6$. After all influencing objects have been taken into account the resulting area is given by the values x_min, x_max, y_min, and y_max. Depending on which relationship the new object has to the first given reference object, the x- and y-values are chosen according to the distance default.

8.1 An example for eight object orientations using the QuaDRO prototype

QuaDRO's observation area is shown on the left of figure 8.1 as an example configuration of objects with eight different orientations. The objects' positions are quantitatively described in the geometrical scene description (*GSD*) presented on the left in figure 8.2. They occupy one grid cell each and are displayed in green, with a yellow edge or corner indicating their front and orientation. The observer is displayed as a blue square also with a yellow edge or corner indicating its orientation and direction of movement. A red 3x3 grid that moves with the observer shows its field of view. Each object that comes into this field is identified and its qualitative relationships to all other objects that have previously been observed are established. In this example, the observer always knows its own position and the position and orientation of any object observed. It is therefore possible to provide a complete object configuration description containing all necessary relationships to reconstruct the configuration without doubt of any object's positional relationships to other objects. The resulting *ocd* is similar to an *ocd* generated from a survey perspective. In this example, the observer starts the observation from the middle of the observation area and continues outwards in a spiral path. The resulting qualitative *ocd* submitted to the listener is presented in figure 8.3.

The listener reconstructs the object configuration following the order the objects are mentioned within the *ocd*. The process starts by placing the first object at coordinate 7,7, which is about the middle of the window initially shown of the reconstruction area. All other objects are successively inserted at their relative positions. The reconstructed configuration is qualitatively correct with regard to all qualitative relationships between the objects, which are identical to the original, considering both the object's intrinsic frames of reference and the underlying absolute frame of reference. However, due to the distance default objects may appear closer to each other than in the original. Furthermore, all movements of objects by horizontal or vertical splits are either done in increasing x- or in increasing y-dimension and the configuration develops more to the bottom and right side of the reconstruction area. In general, the strategy to place the first object in the middle of the reconstruction field might lead to a more shifted reconstruction, as the first object in the *ocd* does not need to be the object in the middle of the configuration. When all objects are placed, the reconstructed scene description (*RSD*), listed in figure 8.2b), contains the objects' quantitative positions within the reconstruction.

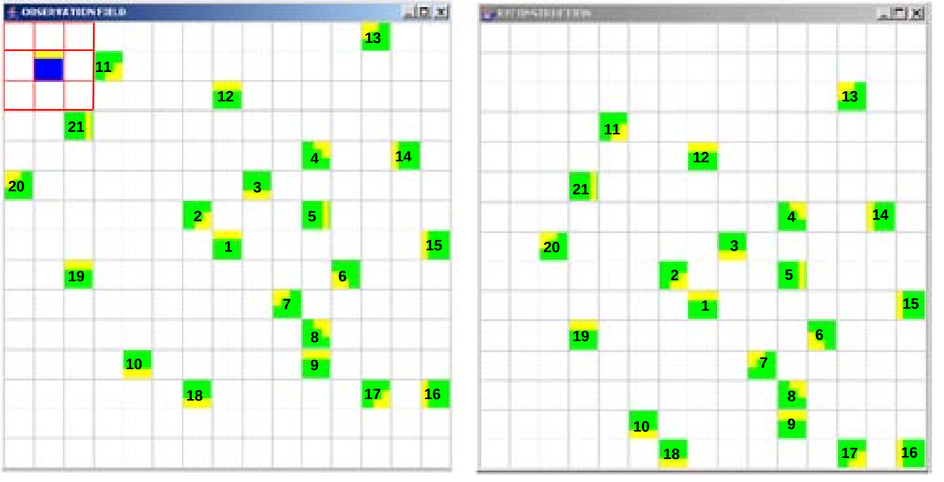


Figure 8.1: *Observation and reconstruction in QuaDRO. Left: The observation area containing objects with eight different orientations. Right: the corresponding reconstructed configuration from the OCD shown in figure 8.3. The numbers in both pictures represent the order in that the observer introduced the objects.*

```
******************   GSD   **********************
*                                               *
*       [0]     [5]     :Northwestobject        *
*       [2]     [3]     :Eastobject             *
*       [2]     [8]     :Northobject            *
*       [3]     [1]     :Southeastobject        *
*       [4]     [11]    :Southobject            *
*       [6]     [6]     :Southeastobject        *
*       [6]     [12]    :Southobject            *
*       [7]     [2]     :Northobject            *
*       [7]     [7]     :Northobject            *
*       [8]     [5]     :Southobject            *
*       [9]     [9]     :Northwestobject        *
*       [10]    [4]     :Northeastobject        *
*       [10]    [6]     :Eastobject             *
*       [10]    [10]    :Northeastobject        *
*       [10]    [11]    :Northobject            *
*       [11]    [8]     :Southwestobject        *
*       [12]    [0]     :Northwestobject        *
*       [12]    [12]    :Southeastobject        *
*       [13]    [4]     :Westobject             *
*       [14]    [7]     :Westobject             *
*       [14]    [12]    :Westobject             *
*                                               *
*************************************************
```

```
*******************   RSD   *********************
*                                               *
*     [2]       [7]     :Northwestobject        *
*     [3]       [5]     :Eastobject             *
*     [3]       [10]    :Northobject            *
*     [4]       [3]     :Southeastobject        *
*     [5]       [13]    :Southobject            *
*     [6]       [8]     :Southeastobject        *
*     [6]       [14]    :Southobject            *
*     [7]       [4]     :Northobject            *
*     [7]       [9]     :Northobject            *
*     [8]       [7]     :Southobject            *
*     [9]       [11]    :Northwestobject        *
*     [10]      [6]     :Northeastobject        *
*     [10]      [8]     :Eastobject             *
*     [10]      [12]    :Northeastobject        *
*     [10]      [13]    :Northobject            *
*     [11]      [10]    :Southwestobject        *
*     [12]      [2]     :Northwestobject        *
*     [12]      [14]    :Southeastobject        *
*     [13]      [6]     :Westobject             *
*     [14]      [9]     :Westobject             *
*     [14]      [14]    :Westobject             *
*                                               *
*************************************************
```

Figure 8.2: *Representation of object positions. Left: The Geometrical Scene Description (GSD) describing the object configuration graphically shown in figure 8.1 in the observation area. Right: The Reconstructed Scene Description (RSD), quantitatively describing the reconstructed configuration shown in figure 8.1.*

Object Configuration Description (OCD)

(2, LF, 1)	(9, RB, 4)	(11, LF, 12)	(15, SF ,11)	(18, LB, 1)	(20, SB, 2)
(1, SF ,2)	(9, RN, 5)	(13, RF ,1)	(15, RB, 12)	(18, RF ,2)	(20, RN, 3)
(3, RF ,1)	(9, SF ,6)	(13, LN, 2)	(15, SB, 13)	(18, RF ,3)	(20, SB, 4)
(3, LN, 2)	(9, SB, 7)	(13, LB, 3)	(15, LB, 14)	(18, SB, 4)	(20, LB, 5)
(2, RF ,3)	(9, RB, 8)	(13, SF ,4)	(14, RF ,15)	(18, RB, 5)	(20, RN, 6)
(4, RF ,1)	(8, SF ,9)	(13, LF, 5)	(16, RB, 1)	(18, SF ,6)	(20, SF ,7)
(4, LN, 2)	(10, LB, 1)	(13, SB, 6)	(16, SF ,2)	(18, LN, 7)	(20, LN, 8)
(4, LB, 3)	(10, RN, 2)	(13, RN, 7)	(16, LF, 3)	(18, SB, 8)	(20, LF, 9)
(3, SB, 4)	(10, RF ,3)	(13, SF ,8)	(16, RN, 4)	(18, LB, 9)	(20, RB, 10)
(5, RF ,1)	(10, SB, 4)	(13, RF ,9)	(16, RF ,5)	(18, LF, 10)	(20, RN, 11)
(5, LF, 2)	(10, RB, 5)	(13, LB, 10)	(16, LN, 6)	(18, SF ,11)	(20, LB, 12)
(5, LF, 3)	(10, SF ,6)	(13, LN, 11)	(16, SB, 7)	(18, LB, 12)	(20, LN, 13)
(5, RB, 4)	(10, LN, 7)	(13, RF ,12)	(16, RN, 8)	(18, LN, 13)	(20, LF, 14)
(4, LN, 5)	(10, SB, 8)	(12, LN, 13)	(16, RB, 9)	(18, LF, 14)	(20, RF ,15)
(6, RB, 1)	(10, LN, 9)	(14, RF ,1)	(16, LF, 10)	(18, LF, 15)	(20, RF ,16)
(6, SF ,2)	(9, LN, 10)	(14, LN, 2)	(16, SF ,11)	(18, SF ,16)	(20, SB, 17)
(6, LF, 3)	(11, LF, 1)	(14, LB, 3)	(16, RB, 12)	(18, RB, 17)	(20, RB, 18)
(6, RN, 4)	(11, SB, 2)	(14, RF ,4)	(16, SB, 13)	(17, LN, 18)	(20, LF, 19)
(6, RF ,5)	(11, RB, 3)	(14, LF, 5)	(16, LB, 14)	(19, LB, 1)	(19, SB, 20)
(5, RN, 6)	(11, LN, 4)	(14, SB, 6)	(16, LN, 15)	(19, RN, 2)	(21, LF, 1)
(7, RB, 1)	(11, LB, 5)	(14, RN, 7)	(15, RN, 16)	(19, RF ,3)	(21, SB, 2)
(7, SF ,2)	(11, RN, 6)	(14, SF ,8)	(17, RB, 1)	(19, SB, 4)	(21, RB, 3)
(7, LF, 3)	(11, SF ,7)	(14, RF ,9)	(17, SF ,2)	(19, RB, 5)	(21, LN, 4)
(7, SB, 4)	(11, LN, 8)	(14, LB, 10)	(17, LF, 3)	(19, RF ,6)	(21, LB, 5)
(7, RB, 5)	(11, LF, 9)	(14, SF ,11)	(17, RN, 4)	(19, SF ,7)	(21, RN, 6)
(7, SF ,6)	(11, RB, 10)	(14, RB, 12)	(17, RF ,5)	(19, LN, 8)	(21, SF ,7)
(6, RN, 7)	(10, SF ,11)	(14, SB, 13)	(17, LN, 6)	(19, LF, 9)	(21, LN, 8)
(8, RB, 1)	(12, SF ,1)	(13, RF ,14)	(17, SB, 7)	(19, RB, 10)	(21, LF, 9)
(8, SF ,2)	(12, LN, 2)	(15, RN, 1)	(17, RN, 8)	(19, RN, 11)	(21, RB, 10)
(8, LF, 3)	(12, RB, 3)	(15, SF ,2)	(17, RB, 9)	(19, LB, 12)	(21, RN, 11)
(8, RB, 4)	(12, LN, 4)	(15, LF, 3)	(17, LF, 10)	(19, LN, 13)	(21, LB, 12)
(8, RN, 5)	(12, LB, 5)	(15, RN, 4)	(17, SF ,11)	(19, LF, 14)	(21, LN, 13)
(8, SF ,6)	(12, RN, 6)	(15, RF ,5)	(17, RB, 12)	(19, LF, 15)	(21, RF ,14)
(8, SB, 7)	(12, SF ,7)	(15, SB, 6)	(17, LB, 13)	(19, RF ,16)	(21, RF ,15)
(7, LN, 8)	(12, LN, 8)	(15, RN, 7)	(17, LF, 14)	(19, SB, 17)	(21, RF ,16)
(9, RB, 1)	(12, LF, 9)	(15, SF ,8)	(17, LF, 15)	(19, RB, 18)	(21, SB, 17)
(9, SF ,2)	(12, LB, 10)	(15, RF ,9)	(17, SF ,16)	(18, RB, 19)	(21, RB, 18)
(9, LF, 3)	(12, SF ,11)	(15, LB, 10)	(16, LF, 17)	(20, LF, 1)	(21, SF ,19)
					(21, RN, 20)
					(20, RB, 21)

Figure 8.3: *The OCD for the configuration shown in figure 8.1 to the left. The OCD was established on a spiral path from the middle of the observation area outwards.*

8.2 The observation scenario in QuaDRO

The scenario addressed in chapter 5 describes an observer on the ground whose field of view is limited. If the environment is too wide to give an overview from one position, the observer has to move around in order to provide position relationships for all objects. This excludes the possibilities of a description from a survey perspective or a gaze tour [Tve99; Lev82] from a single position. Instead he describes the object configuration performing a route tour between certain objects and describing the relationships of all observed objects in relation to the visited objects in a gaze tour and in relation to each other in a momentarily applied absolute frame of reference. This section presents the description and reconstruction processes that are used to realize the scenario and shows an example observation and reconstruction in the QuaDRO technical prototype.

8.2.1 The route tour scenario

The observer performs a route tour between certain objects. Each of these objects functions as a reference object for the time the observer is present. At each visited object, all objects observed from this point are described in relation to the reference object in a gaze tour. Thereafter the relations between the object in the same frame of reference, now functioning as a momentarily applied absolute frame of reference, are given. At the first reference object, the applied pseudo intrinsic frame of reference functions at the same time as the underlying absolute frame of reference. The observer does not have to keep track of his orientation within this absolute frame of reference, but he needs to know if he moves in alignment or at an angle to it. When he moves in alignment with it, the next reference object uses the frame of reference where the object's orientation is to one of its edges. When the observer moves at an angle to the underlying absolute frame of reference the next reference object uses the frame of reference rotated the by 45° where the object's orientation is at one of its corners.

As described in [Tve99] people are able to change perspectives during a task. Furthermore, they are often willing to accept a higher cognitive load if they feel that this might alleviate the cognitive load for their communication partners. Therefore, the observer is asked to switch between both frames of reference. The only information for him to remember is what frame of reference he used at the last route-tour stop. If he leaves the object in one of the relative directions *sf*, *sb*, *rn*, or *ln* he must not change the frame of reference, but if he leaves in one of the directions *rf*, *rb*, *lf* or *lb* the frame of reference must be changed. Furthermore, the observer is assumed to recognize objects that he has observed before and is observing again from a different viewpoint. An activity diagram illustrating the route tour is presented in figure 8.4. Algorithm 10 'buildOCDbyGazeTour' and algorithm 11 'buildOCDinMomentarilyAppliedAbsoluteFoR' illustrate the two main subprocesses of a route tour.

Algorithm 10 takes as input the observer's current position as an x,y-coordinate

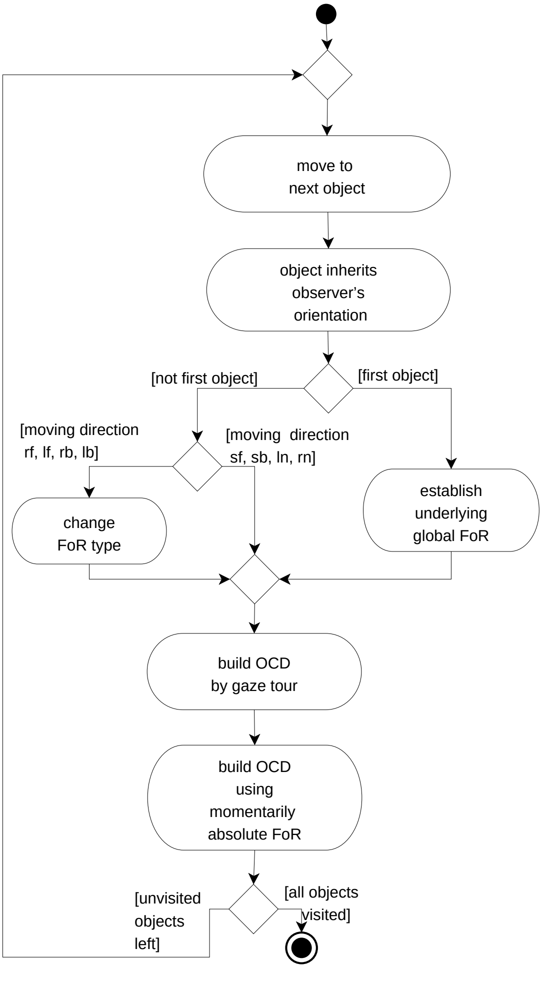


Figure 8.4: *The route tour process.*

and the object that is located at this position. Furthermore it gets a set of all objects that are within the observer's field of view, while he stands at his current position. For every observed object a new ocd_entry is created that contains information of which object (ocd_entry.target) is in which relations (ocd_entry.relation) to the object at the current position (ocd_entry.reference). When all objects have been described in this way, the algorithm returns the ocd.

The algorithm 'buildOCDinMomentarilyAppliedAbsoluteFoR' also receives as input the object located at the observer's current position, and the set of all objects observed from this position. It then creates an ocd entry for every object (outer for loop) to every other object in this set (inner for loop). The algorithm returns the accumulated ocd_entries, collected in the ocd.

```
Algorithm 10 buildOCDbyGazeTour
=========================================================
input:
 currentPosition (The observer's current position)
 currentObject (The object at the current position)
 observedObjectList (A sequence of all objects observed
                     from the current position)

output:
 ocd (A sequence of ocd_entries.)
 --------------------------------------------------------
    for all entries in the observedObjectList do
    create a new ocd_entry
    ocd_entry.target <- the current object from the
                        ObservedObjectList
    ocd_entry.relation <- the relation the
                          current target object has
                          in relation to the
                          orientation of the object
                          at the current position
    ocd_entry.reference <- the object at the current
                           position
    add ocd_entry to ocd
    end for
    return ocd
---------------------------------------------------------
```

```
Algorithm 11 buildOCDinMomentarilyAppliedAbsoluteFoR
=========================================================
input:
 currentObject (The object at the current position)
 observedObjectList (A sequence of all objects observed
                     from the current position)

output:
 ocd (Asequence of ocd_entries.)
 --------------------------------------------------------
 for all entries in the observedObjectList do
  reference <- current observedObjectList entry
  for all entries in the observedObjectList do
   target <- current observedObjectList entry
   if target == reference
    then %
    else
    if target != reference
    then
     create a new ocd_entry
     ocd_entry.target <- target
     ocd_entry.relation <- the relation of the
                           target object to the
                           reference object in
                           the FoR originated
                           by the currentObject's
                           orientation
    ocd_entry.reference <- reference
    add ocd_entry to ocd
   end if
  end for
 end for
 return ocd
---------------------------------------------------------
```

The listener needs to keep track of the observer's movements in order to translate the relationships given in different frames of references into the absolute frame of reference used in the reconstruction. As the observer first describes where the objects are in relation to his first viewpoint the listener knows what relative orientation the observer takes when he moves to the next object. If the observer moves freely in order to find new objects, he has to inform the listener of all orientation changes. In order for the listener to know when the frame of reference has changed, the observer must inform the listener when he changes position. Figure 8.5 shows a possible route tour between three objects. The object configuration description (*ocd*) for the example is: {*(2 LB 1)*, (*3 RB 1*), (*4 RF 1*), (*5 SF 1*), (*2 LB 5*), (*4 RB 5*), (*3 RB 5*), (*4 RF 2*), (*4 RF 3*), (*3 RN 2*), *continuing right front to object 4*, (*1 SB 4*), (*3 SB 4*), (*6 RF 4*), (*7 SF 4*), (*3 SB 6*), (*1 SB 6*), (*7 LN 6*), (*3 SB 7*), (*1 SB 7*), (*3 RN 1*), *continuing straight front to object 7*, (*4 SB 7*), (*6 RN 7*), (*5 LB 7*), (*5 LN 6*), (*4 LB 6*), (*5 LN 4*)}.

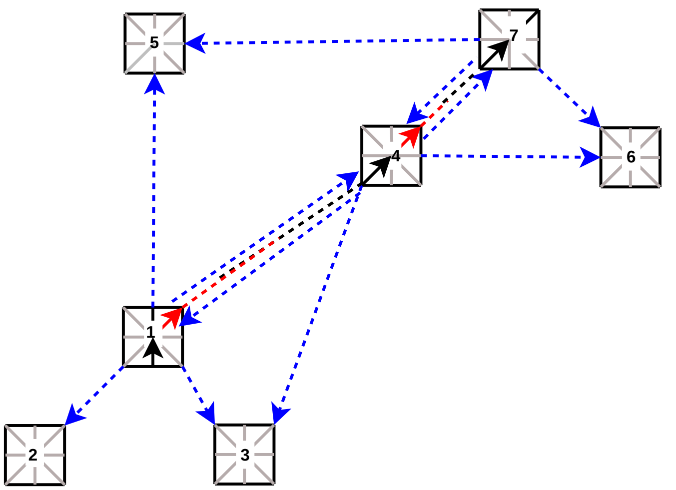


Figure 8.5: *Route tour (red/black) combined with several gaze tours (blue).*

As in chapter 5, the listener uses the terms *north*, *northeast*, *east*, *southeast*, *south*, *southwest*, *west*, and *northwest* as global directions in the reconstruction. The intrinsic relationships can be translated into global relationships using the two formulas:

FoRGlobalOrientation
= NS(((SN(previousFoRGlobalOrientation) + SN(movingDirection)) mod 8), R)

globalRelation
= NS(((SN(FoRGlobalOrientation) +SN(intrinsicRelation)) mod 8), G)

Knowing in which direction the observer has moved enables the listener to follow the angle that the applied frame of reference has in relation to the underlying global frame of reference. This translated *ocd* is better described by the term global configuration description (*gcd*) as it contains only global relationships between objects. The observer might have described the relationship of two objects from different viewpoints. These relationships translate to the same relationship in the global frame of reference. For a smooth reconstruction, it is advantageous if the description is sorted into a well ordered *gcd* where all duplicate relationships are removed. In the previous example, the well ordered *gcd*, where the listener chooses the orientation *north* for object *1* is: {(*2 southwest 1*), (*3 southeast 1*), (*3 east 2*), (*4 northeast 1*), (*4 northeast 2*), (*4 northeast 3*), (*5 north 1*), (*5 northeast 2*), (*5 northwest 3*), (*5 northwest 4*), (*6 east 4*), (*6 northeast 3*), (*6 northeast 1*), (*6 southeast 5*), (*7 northeast 4*), (*7 northwest 6*), (*7 northeast 3*), (*7 northeast 1*), (*7 east 5*)}. The activity diagram in figure 8.6 provides an overview of the reconstruction process.

Algorithm 12 ‘reconstructObjectConfigurationFromGCD’ takes the *gcd* as input and provides an RSD and the GCD component of all relations within the *gcd* that have been reconstructed. If the provided *gcd* is not empty the algorithm starts by placing the first object, that is not mentioned in the *gcd* into the reconstruction at position 7,7. (line 3 - 7). For all entries in the *gcd* that introduce new objects an rsd_entry is created in line 13. The object’s position is calculated by the function ‘placeNextObjectGlobal’ called in line 15 and described in algorithm 13. The returned position is attached to the rsd_entry for that object in lines 16 and 17. The rsd_entry is added to the RSD (line 18) and the gcd_entry is added to the GCD component of all reconstructed relationships. Algorithm 13 is almost similar to algorithm 3 (‘placeNextObject’). The difference is that it uses the *gcd* instead of the *ocd* and global relationships instead of intrinsic relationships. Therefore, in line 7 the referenceRegion can be found directly by applying definition 6.3 and no further translation of relationships is required.

A reconstruction accomplished by the previous two algorithms does not contain the object’s orientations. During the route-tour description the objects were assumed to be orientationless, the intrinsic frame of reference were attached to them, which were identical oriented as the observer. Therefore, the reconstruction containing only orientationless objects does not lead to a loss of information. A configuration like this is called a global configuration, formally described in definition 8.1. Compared to the previously introduced configurations a global configuration lacks the components R and OR for object orientations and the components S and OCD that are used for the object’s intrinsic orientations.

OCD-consistency is not applicable for global configurations, since no OCD-component is available. Instead, reconstruction is done with the *gcd* as input and the GCD-component is successively established with every *gcd* entry encountered. Therefore, for these types of configuration GCD-consistency (definition 8.2) is necessary to assure that the reconstruction mirrors all given constraints in the input *gcd*.

```
Algorithm 12. reconstructObjectConfigurationFromGCD
=========================================================
input:
 gcd (a sequence of gcd_entries)

output:
 RSD (a sequence of rsd_entries)
 GCD (a sequence of gcd_entries that have been
      reconstructed)
 --------------------------------------------------------
    if gcd != {}
     then
      create new rsd_entry
      rsd_entry.x <- 7
      rsd_entry.y <- 7
      rsd_entry.object <- 1
      add rsd_entry to rsd
    end if
    for all entries in the gcd do
     if current gcd_entry.target
        = previous gcd_entry.target
      then %
      else
       create new rsd_entry
       rsd_entry.object <- current gcd_entry.target
       position <- placeNextObjectGlobal
                    (current gcd_entry.target, gcd, RSD)
       rsd_entry.x <- position.x
       rsd_entry.y <- position.y
       add rsd_entry to RSD
       add gcd_entry to GCD
     end if
    end for
    return RSD, GCD
---------------------------------------------------------
```

```
Algorithm 13. placeNextObjectGlobal
==========================================================
input:
 gcd_entry (the target object has to be placed)
 gcd (The global object configuration description)
 RSD (The reconstructed scene description)

output:
 targetPosition (The x,y coordinate for the target)
----------------------------------------------------------
target <- gcd_entry.target
targetRegion <- the whole reconstruction area
referenceRelation <- gcd.entry.relation
for all entries in the gcd that have target
       as target object do
 referenceObject <- the chosen
                          gcd_entry.reference
 relation <- the chosen gcd_entry.relation
 repeat
  referenceRegion <- All x,y coordinates that fulfill
                     the definition for this relations
                     following definition 6.3
  targetRegion <- intersect
                  (referenceRegion, targetRegion)
  if targetRegion is empty
   then moveObjects
     (referenceObject, rsd, relation)
  end if
 until targetRegion is not empty
 add the chosen gcd_entry to GCD
end for
targetPosition <- insertObjectUsingDistanceDefault
                  (targetRegion, referenceRelation)
return targetPosition
----------------------------------------------------------
```

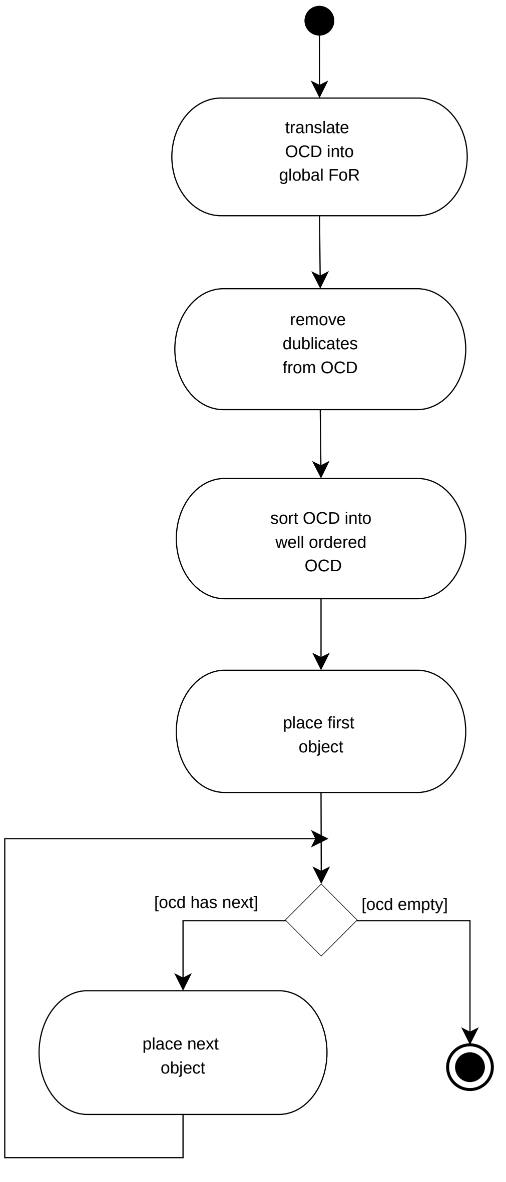


Figure 8.6: *The reconstruction process.*

Definition 8.1 ***global Configuration***

A simple configuration is a configuration Conf = $\langle$O, P, L, X, OL, G_8, RSD, GCD$\rangle$ where

- O a finite set of objects.
- P is a mapping from O to $\mathbb{N}^+$ reflecting the order in which the objects have been encountered.
- L $\subseteq \mathbb{N}^+ \times \mathbb{N}^+$, representing the set of points in the Cartesian plane with integer coordinates.
- $X \subseteq L \times L = \{<x, y>, <x', y'> : x=x' \wedge y=y'\}$ the equality relation.
- OL is a mapping from O to L with $OL(o)=<x, y> : o \in O$.
- G_8 is a finite set of binary predicates: {N, NE, E, SE, S, SW, W, NW}.
- RSD is a finite set of tuples of the form $<o, OL(o)>: o \in O$.
- GCD $\subseteq$ O$\times G_8 \times$O the global configuration description.

Definition 8.2 ***GCD-consistency***

A configuration Conf = $\langle$O, P, L, OL, G, RSD, GCD$\rangle$ is GCD-consistent

iff $\forall <o, r, o'> \in GCD: <OL(o), r, OL(o')>$

8.2.2 Preserving GCD-consistency

The reconstruction process (algorithm 12) preserves GCD-consistency. The proof is similar to the previously presented proofs of OCD-consitency.

If the input (*gcd*) to algorithm 12 is not empty, algorithm 12 starts with placing the first object (o_1), which itself is not mentioned in the *gcd*, into the reconstruction area (lines 1 to 7). The resulting configuration $Conf_1 = \langle o_1$, P(o_1), L, X, OL(o$_1$), G_8, $<o_1, OL(o_1)>$, $\emptyset\rangle$ obviously is GCD-consistent, since the GCD-component so far considered in the reconstruction is empty.

The loop starting in line 9 of algorithm 12 begins with this GCD-consistent configuration. For each new object o' introduced in the *gcd*, algorithm 13 'placeNextObjectGlobal' is called in line 15 of algorithm 12. So far the configuration has not been changed, therefore, algorithm 13 starts with a GCD-consistent configuration.

Generally, every time the algorithm is called it starts with a GCD-consistent configuration Conf = ⟨O, P, L, X, OL, G_8, RSD, GCD⟩. In line 1 the target region (*targetRegion*) for the new object o' is set to the whole reconstruction area. At this moment no relationships of object o' to other objects have been considered.

Let the reconstruction area be defined by the values X_{min}, X_{max}, Y_{min}, and Y_{max}. The target region therefore contains all locations
<x,y>: (X_{min} ≤ x ≤ X_{max}) ∧ (Y_{min} ≤ y ≤ Y_{max}). Let it be defined by $X_{min_{target}}$=X_{min}, $X_{max_{target}}$ =X_{max}, $Y_{min_{target}}$=Y_{min}, $Y_{max_{target}}$=Y_{max}.

By the loop starting in line 3, a set (let it be called N) of triples of the form <o', r, o_i> describing the relationships of the new object o' that is not yet a member of O, to objects $o_i \in$ O, P(o_i) = i, (previously inserted objects) is considered.

Each iteration of the for loop takes the relation r of the new object o', which is called the target, to the provided reference object o_i (*referenceObject*). The referenceRegion is calculated to contain all x,y-coordinates that fulfill the definition of the relation (r) following definition 6.3. The resulting referenceRegion is the area defined by $X_{min_{reference}}$, $X_{max_{reference}}$, $Y_{min_{reference}}$, $Y_{max_{reference}}$.

In line 8 the targetRegion is intersected with the referenceRegion and only those locations that remain are:
{<x,y>: Max($X_{min_{target}}$,$X_{min_{reference}}$) ≤ x ≤ Min($X_{max_{target}}$, $X_{max_{reference}}$)
∧ Max($Y_{min_{target}}$,$Y_{min_{reference}}$) ≤ y ≤ Min($Y_{max_{target}}$, $Y_{max_{reference}}$)}.

In case this intersection is empty the function 'moveObjects' is called in line 10. Depending on the situation, this function calls either 'splitVertically' or 'splitHorizontally' both of which return modified configurations but preserve GCD-consistency in the same way they preserve OCD-consistency. (The proof is straight forward and the same as for OCD-consistency and therefore omitted.) Thus if line 10 has been executed the previous steps in lines 6 to 12 are repeated on a modified but GCD-consistent configuration.

If the target region is not empty, obviously all locations in this area are the only locations that previously belonged to both *referenceRegion* and *targetRegion*. The *referenceRegion* calculated using definition 6.3 ensures that all positions in the remaining *targetRegion* conform to the relation r ∈ <o', r, o_i>.

Further iterations of the for loop (lines 3 to 14) lead to a target region that conforms to all spatial relations provided for object o'. When the loop finishes in line 14, the target position is chosen by using the distance default (definition 6.9) out of all locations within the *targetRegion*. Therefore, the chosen location conforms to all relations considered. Algorithm 13 ends by returning this location to algorithm 12, the only changes that algorithm 13 makes to the configuration is made by a horizontal or vertical split both of which preserve GCD-consistency. Therefore after algorithm 13 has terminated, the configuration remains GCD-consistent.

Algorithm 12 adds the <o' OL(o')> to the RSD (lines 16 to 18), which changes the configuration by adding the new object o' at its position within the grid. As this position is the intersection of all relations r considered and iteratively calculated following definition 6.3 the modified configuration is GCD-consistent.

8.2.3 An example scenario in the QuaDRO prototype

The objects in the observation area, shown on the left of figure 8.7 are orientationless. Generally, the observer can start at any point within the observation area. The size of its field of view can range from a global overview to just one cell visited at a particular time. He is able to move in eight different directions. Any implemented parsing strategy can be applied. The example given here uses a simple parsing strategy that provides a good result in an environment with a certain density of objects.

The observer starts from the lower edge and moves forward until the first object comes into his field of view. His own position is now his first viewpoint, in relation to which he describes the positions of the observed object. He moves to that object which becomes his second viewpoint which then inherits the observer's orientation at arrival. In a gaze tour he describes the objects he sees in relation to the viewpoint and in a momentarily applied absolute frame of reference their relations to each other. In the observation field's graphical representation a visited object shows which orientation the observer had with a yellow edge or corner. On the right of figure 8.7 the observation area is shown following a few moves. In this example, the observation ends when the observer does not see any further objects that he had not previously observed. On the left of figure 8.8, the objects that are only colored green have been observed from several positions but have never been used as viewpoints themselves. In order to obtain an *ocd* that allows for the reconstruction process to be made the same way as described in previous examples the *ocd* is first translated into a description where all relations are given in the same frame of reference and all duplicate relations are removed.

Due to the observer's limited field of view, he can only describe the relationship of two objects if both objects are present within his field of view at the same time. In the case of an object's relationship to an influencing object not being available, the influencing object's possible impact on that object's position is ignored. This might, of course, lead to misplaced objects and, depending on fault accumulation, in certain cases even to wrong reconstructions. The next chapter provides a different approach that considers the case where some relationships are not known.

This chapter briefly introduced the QuaDRO technical prototype developed to describe an object configuration qualitatively and to reconstruct the configuration from that description alone. Two examples were presented to illustrate the observation of objects with eight individual intrinsic orientations and the route tour scenario described in chapter 5. So far the object configuration description has either been a complete description of the observed objects, or as in the implemented route tour scenario, missing object relations have been ignored. In certain situations this might lead to incorrect reconstructions. It is advantageous to use

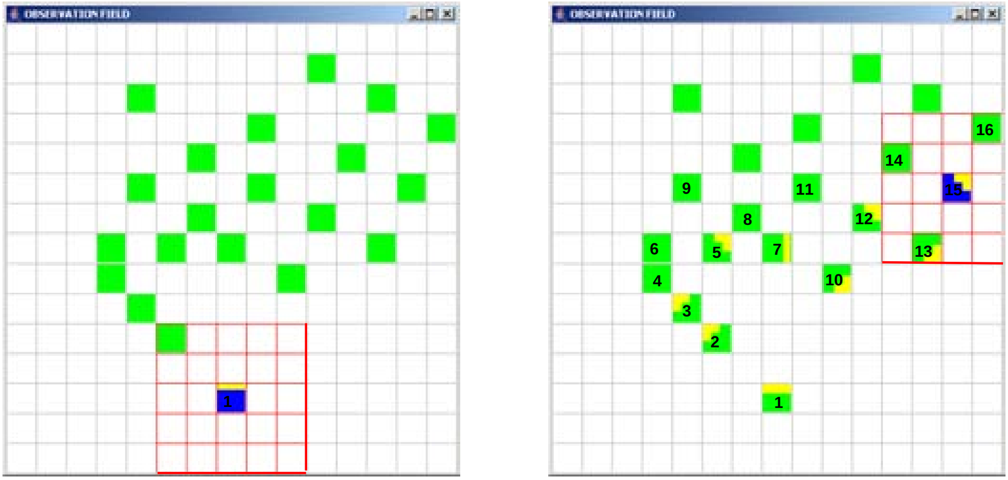


Figure 8.7: *The route tour scenario in QuaDRO. Left: A configuration of objects without own intrinsic orientations. The observer has reached a position from that he sees the first object in the top left corner of the view field. Right: The same configuration after a while. The visited objects show the orientation that the observer used for the description while standing at the objects.*

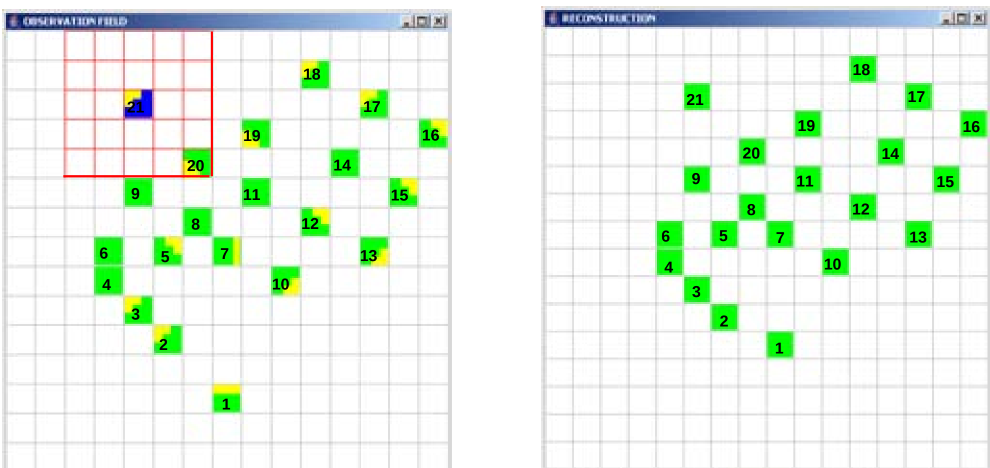


Figure 8.8: *The route tour scenario in QuaDRO. Left: The observation process has come to an end. Not all objects have been visited but all have been observed. Right: The configuration reconstructed from the provided OCD.*

reasoning strategies to achieve at least some knowledge about missing object relationships. Therefore, the following chapter introduces the reasoning techniques suitable for usage in QuaDRO. It further describes how objects with unknown or partly unknown relationships are represented in the reconstruction.

Chapter 9

Object configuration reconstruction using coarse relations

In all processes described earlier, the *ocd* has been a complete description of the object configuration using only the eight basic relations. When a relationship between two objects would not have been available within the *ocd* the reconstruction process would ignore this relationship. Under certain circumstances, ignoring certain relationships might lead to reconstructions that are not qualitatively equivalent to the original object configuration, with regard to the nine equivalence classes. To improve the reconstruction process and to provide qualitatively correct reconstructions even from underspecified *ocd*s, reasoning techniques to obtain coarse relationships between objects are necessary. Section 9.1 explains how QuaDRO can be extended with reasoning techniques in order to gain coarse information about missing object relationships. Section 9.2 shows how objects with coarse relationships are represented within the reconstruction in order to establish qualitatively correct object configuration reconstructions. The reconstruction process for underspecified *ocd*s is presented in section 9.3 and illustrated by an example in section 9.4. In addition to the provided techniques, several further strategies are necessary to cover all possible situations that might arise during the reconstruction process when using coarse relations. These strategies are described in section 9.5.

Definition 9.1 G_{16} **- *General Global Relations***

The set G_{16} of spatial relations is a finite set of binary predicates over locations <x,y>,<x',y'> : x, x', y, y' $\in \mathbb{N}^+$ in the Cartesian plane. It contains the previously defined set G_8 of global relations as a subset and adds the relations SAME and the coarse relations NORTH, EAST, SOUTH, WEST, NEUTRAL-V, NEUTRAL-H, and OPEN.

<x',y'> N <x,y> iff $(x' = x) \wedge (y' < y)$
<x',y'> NE <x,y> iff $(x' > x) \wedge (y' < y)$
<x',y'> E <x,y> iff $(x' > x) \wedge (y' = y)$
<x',y'> SE <x,y> iff $(x' > x) \wedge (y' > y)$
<x',y'> S <x,y> iff $(x' = x) \wedge (y' > y)$
<x',y'> SW <x,y> iff $(x' < x) \wedge (y' > y)$
<x',y'> W <x,y> iff $(x' < x) \wedge (y' = y)$
<x',y'> NW <x,y> iff $(x' < x) \wedge (y' < y)$
<x',y'> SAME <x,y> iff $(x' = x) \wedge (y' = y)$
<x',y'> NORTH <x,y> iff $y' < y$
<x',y'> EAST <x,y> iff $x' > x$
<x',y'> SOUTH <x,y> iff $y' > y$
<x',y'> WEST <x,y> iff $x' < x$
<x',y'> NEUTRAL-V <x,y> iff $y' = y$
<x',y'> NEUTRAL-H <x,y> iff $x' = x$
<x',y'> OPEN <x,y> iff nothing is known about the relationships

Every coarse relation can be expressed as a disjunction of basic relations:

NORTH = $\{$NW $\vee$ N $\vee$ NE$\}$
EAST = $\{$NE $\vee$ E $\vee$ SE$\}$
SOUTH = $\{$SW $\vee$ S $\vee$ SE$\}$
WEST = $\{$NW $\vee$ W $\vee$ SW$\}$
NEUTRAL-H = $\{$W $\vee$ SAME $\vee$ E$\}$
NEUTRAL-V = $\{$N $\vee$ SAME $\vee$ S$\}$
OPEN = $\{$N $\vee$ NE $\vee$ E $\vee$ SE $\vee$ S $\vee$ SW $\vee$ W $\vee$ NW $\vee$ SAME$\}$

9.1 Reasoning

As mentioned in chapter 3, many existing qualitative calculi have been developed for reasoning about implicitly available information. See for instance [Fra91; Fre92b; Lig93; Ren04; Mor05; Sch95; Mor00; Dyl05; Güs89; Bal98; Ski05]. Using QuaDRO's terminology in an absolute frame of reference a typical reasoning question could be: Given the relationships (*2 NE 1*) and (*3 NE 2*) what is the relationship of object *3* to object *1*? An absolute frame of reference is used, since reasoning is done within the reconstruction process after the relationships provided have been translated into the reconstruction's frame of reference. The answer to this question is: (*3 NE 1*), which under these circumstances is the only possible relationship of the two objects. Composition tables based on conceptual neighbourhoods of spatial relations [Fre91a; Fre92c; Fre92a; Zim96] are often used to provide answers to these kind of questions. The relationship of object *2* to object *1* is presented in the table's row and the relationship of object *3* to object *2* in its column. At the intersection of row and column, all possible relationships of object *3* to object *1* can be found. The composition table for QuaDRO is given in figure 9.1. The table is based on similar sized objects. For instance the relationships in the third row, (*2 E 1*), and first column, (*3 N 2*), are evaluated to (*3 NE 1*) which is only correct for equally sized objects. If the objects are of different size the two relationships (*3 NE 1*) and (*3 E 1*) are possible.

However, the usual question in the tasks presented here is: Given the relationship of object *2* to object *1* and the relationship of object *3* to object *1* what is the relationship of object *3* to object *2*? For example, in the case (*2 NE 1*) and (*3 NW 1*). To use the composition table the relationship (*2 NE 1*) needs to be inverted to (*1 SW 2*). The new relationship can then be looked up in the composition table in the intersection of (*1 SW 2*) and (*3 NW 1*) which turns out to be the coarse relationship WEST, which can also be interpreted as the disjunction of the relationships ((*3 NW 2*) $\vee$ (*3 W 2*) $\vee$ (*3 SW 2*)). A composition table where the relationship of object *3* to object *2* can be found without the need to first reverse the relationship, is given in figure 9.2. This table principally contains the same information as the table in figure 9.1 and its presence here is just for convenience. Most composition table entries allow only one basic relationship of the two objects. Even if such a relationship is missing from the *gcd* there is implicitly only this single alternative and the object is automatically placed correctly into the reconstruction area without the need for reasoning.

Reasoning becomes necessary for relationships of objects that have been introduced as influencing objects in chapter 6. The composition table shows that in all these cases, there are at least three possible basic relationships, in some cases even nine possible basic relationships. We call the triples of possible relations *coarse relations* and name them *EAST* containing the relations *NE*, *E*, and *SE*, *WEST*, containing the relations *NW*, *W*, and *SW*, *NEUTRAL-V* (v for vertically) containing the relations *N*, *SAME*, and *S*, *NORTH* containing the relations *NE*, *N*, and *NW*, *NEUTRAL-H* (h for horizontally) containing the relations *W*,

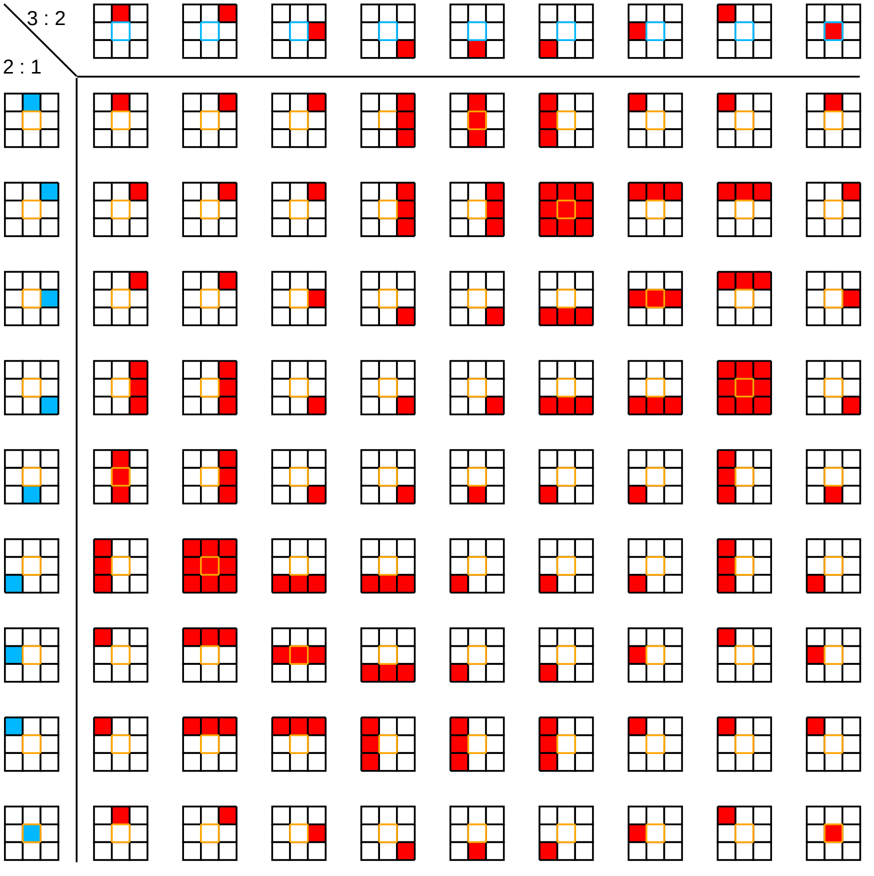


Figure 9.1: *The composition table for QuaDRO. The first column shows the relationship of object 2 (blue) to object 1 (yellow), the first row the relationship of object 3 (red) to object 2 (blue). In the intersections the relationships of object 3 (red) to object 1 (yellow) are found.*

SAME, and *E*, and *SOUTH*, containing the relations *SE*, *S*, and *SW*. The relation *OPEN* denotes the case where all nine relationships are possible. Together with the previously defined set of global relations G_8 these build a set of sixteen global relations called G_{16} as defined in definition 9.1.

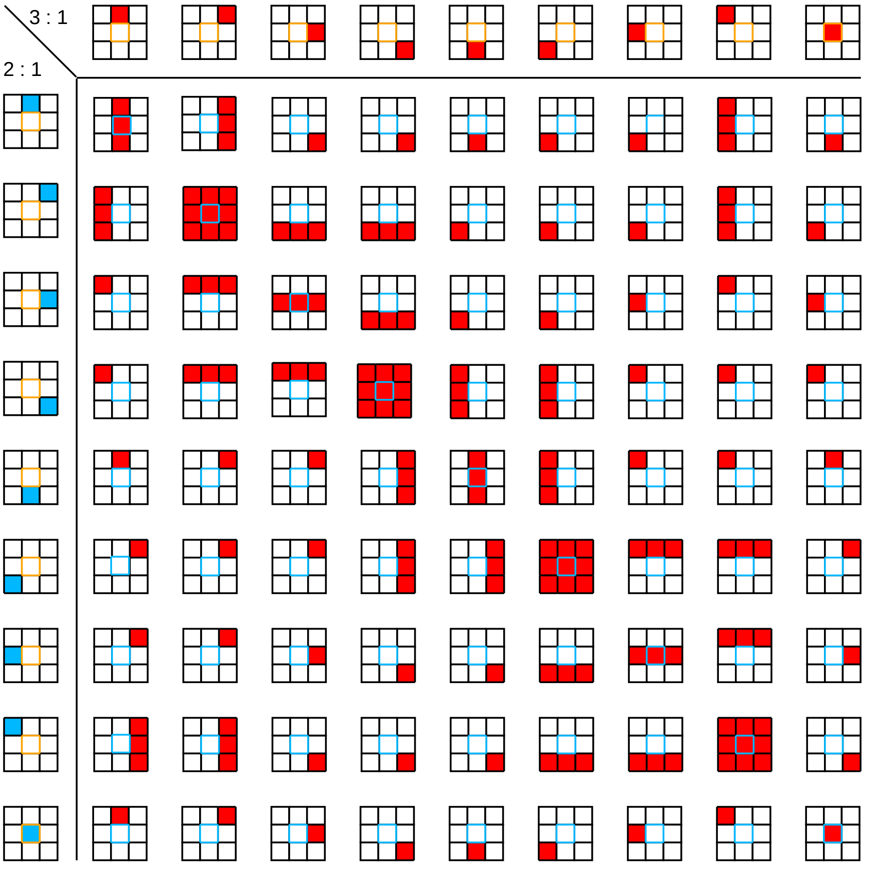


Figure 9.2: *A more convenient composition table for the type of reasoning questions typically used in QuaDRO. The table shows the relationship of object 2 (blue) to object 1 (yellow) in the first column and the relationship of object 3 (red) to object 1 (yellow) in the first row. The intersection provides the relationship of object 3 (red) to object 2 (blue).*

9.2 The allocation of space for objects with coarse relationships

In the reconstruction processes described so far, the position for an object was calculated by intersecting the regions given by the reference object and the influencing objects to a resulting area (target region) where the object is allowed to be. This area is defined by the values $X_{min_{target}}$, $X_{max_{target}}$, $Y_{min_{target}}$, $Y_{max_{target}}$. Following the distance default, this area's cell nearest the already reconstructed part of the configuration is chosen as the object's position.

When an object's relationship to an influencing object is not known and reasoning will provide a coarse relationship that is a disjunction of three or more possible basic relationships, the object's position becomes ambiguous. If one definite position out of the possible positions were chosen, there is only a one in three chance to pick the correct one. In order to provide the correct result a nondeterministic approach could establish all possible reconstruction alternatives in parallel. One of these alternatives qualitatively matches the original configuration. However, without comparing the reconstruction to the original it is not possible to know which one it is. It is likely that many incorrect reconstructions will be terminated by ending in contradictory states, however, this is unfortunately not guaranteed.

The intention in QuaDRO is to embed the nondeterminism into just one reconstruction that, in addition, is qualitatively correct in any reconstruction step, regarding the objects' relative positions to each other described by the nine given equivalence classes. To achieve this, an object's concrete position is not decided upon as long as this position is unclear. Instead, the minimal space that covers all possible relative positions for the object is allocated. The following reconstruction steps deal with the fact that the object could be at any of these positions and the following objects are inserted in the way that whichever position the object takes, the rest of the reconstruction will still be correct. As long as no further information clarifies the object's position, the whole area that represents the object is treated as the object. Dealing with coarse objects includes some new situations that have to be dealt with in order to develop qualitatively correct reconstruction results.

- The first case has already been discussed. When the object's relationship to an influencing object is unknown, reasoning will be used to decide if the object is *EAST*, *WEST*, *NORTH*, *SOUTH*, *NEUTRAL-H*, or *NEUTRAL-V* of the influencing object or if its relationship is *OPEN*. Enough space is allocated for the object for all the basic relationships that the concluded coarse relationship consist of to be fulfilled.

  In figure 9.3a) the relationships (*2 northwest 1*) and (*3 northeast 1*) are given. Object *2* is an influencing object to object *3*'s position. With help of reasoning the relationship can be identified as (*3 EAST 2*). This means that in the end, object *3* will be either *NE*, *E*, or *SE* of object *2*. From now on at least three cells, one for each possible relation, represent object *3* until further information resolves the ambiguity.

- The more relationships to influencing objects that are missing the bigger the area that represents a coarse object becomes. Figure 9.3b) shows the situation where the relationships (*2 N 1*), (*3 N 2*), (*4 E 1*), (*5 E 4*), and (*6 E 5*) are enough to reconstruct the configuration excluding object *7*. For the position of object *7* only the relationship (*7 N 6*) is given. There is no doubt about object *7*'s relationships to objects *1*, *4*, and *5* but objects

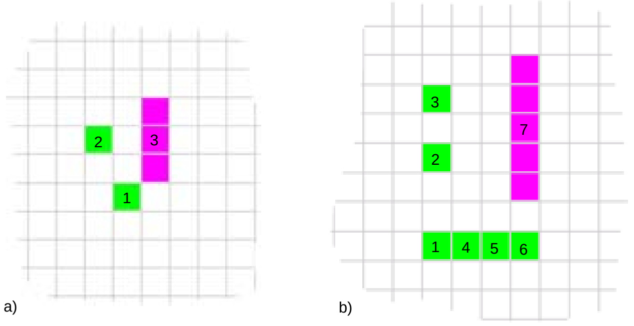


Figure 9.3: *Space allocation for coarse objets. a) gcd: (2 NW 1), (3 NE 1). Reasoning provides the relationship (3 EAST 2). b) gcd: (2 N 1), (3 N 2), (4 E 1), (5 E 4), (6 E 5), (7 N 6). Reasoning provides the relationships (7 EAST 2) and (7 EAST 3).*

2 and *3* are influencing objects. After applying reasoning the coarse relationships (*7 EAST 2*) and (*7 EAST 3*) are available. In order to leave all possible contained basic relationships open the space to be allocated for object *7* needs to cover the situation where *7* is *NE* of object *3* and the situation where it is *SE* of object *2*. In addition, care must be taken that the relationships (*7 SE 3*) and (*7 NE 2*) are possible at the same time. To achieve the latter at least one cell space is needed between objects *2* and *3*. If the space is not already available between these two objects, they have to be moved apart by a vertical split in order to obtain the necessary space.

- When influencing objects are in two dimensions the coarse object's area expands into both dimensions as shown in figure 9.4. In 9.4a) object 5's relationship to object *2* is not presented in the *gcd* and is calculated to *EAST*. Furthermore, object 5's relationship to object *4* is not available and is therefore calculated as *NORTH*. Figure 9.4b) shows object *6* whose relationships to the influencing objects *2*, *4* and *5* are unclear and calculated as (*6 EAST 2*), (*6 NORTH 4*), and (*6 NORTH 5*).

- Opposed to the single cell objects that are situated in the announced cells, are coarse objects represented by an area, in one out of the area's cells. However, a coarse object represented by an area is in principle treated in the same way as a single cell object. It can be moved in x- and y-dimensions by a horizontal or vertical split (definition 6.10 and 6.11) in order to obtain space for other objects. Used as a reference object the frame of reference is applied around the whole area representing the object.

- When a reference object is a coarse object, the target object becoming a coarse object or being represented as a single cell object depends on the

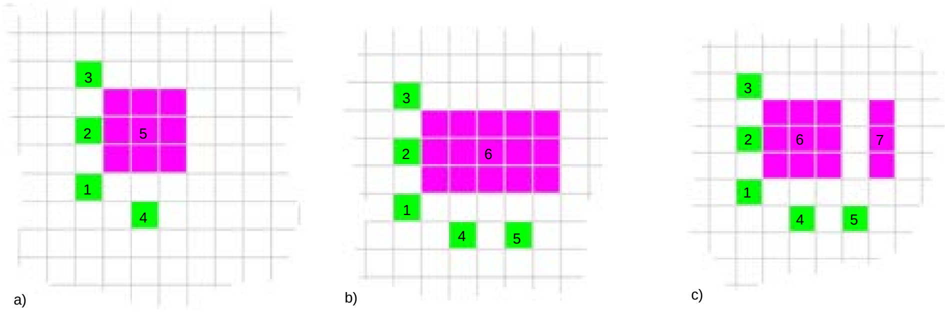


Figure 9.4: *Space allocation for coarse objects. a) gcd: (2 N 1), (3 N 2), (4 SE 1), (5 NE 1), (5 SE 3). Reasoning provides the relationships (5 EAST 2) and (5 NORTH 4). b) gcd: (2 N 1), (3 N 2), (4 SE 1), (5 E 4), (6 NE 1), (6 SE 3). Reasoning provides the relationships (6 EAST 2), (6 NORTH 4), and (6 NORTH 5). c) gcd: (2 N 1), (3 N 2), (4 SE 1), (5 E 4), (6 NE 1), (6 SE 3), (6 NW 5), (7 NE 5), (7 E 6).*

target object's relationship. When the target object is in one of the relationships *NE*, *SE*, *NW*, or *SW* to the reference object and no other influencing object has any further impact on its position it is represented by one cell only. In these relationships, the reference object's size has no impact on the target object's representation.

- If the target object is in the relationship *N*, *S*, *E*, or *W* of the reference object, the target object's size is influenced by the reference object's size. Figure 9.4c) shows the example where the target object *7* is *E* of reference object *6*, which is represented by an area. Depending on the reference object's exact and so far unknown position the position of the target object will be one out of three possible cells. As long as the reference object's position is not clarified, the target object's position is neither clarified and all cells covering the qualitative area for all possible positions need to be allocated for the target object. Whenever further information about object *6* reduces or increases object 6's area, the area of object *7* needs to be adjusted accordingly.

- While applying a vertical or horizontal split, the separating line might intersect a coarse object. In this case the object must be extended, so that the part of the object on one side of the line is moved and the cells that appear in the middle between the two object parts become part of the object.

9.3 The reconstruction process for coarse objects

QuaDRO's reconstruction process for objects at uncertain positions follows the activity diagram presented in the previous chapter in figure 8.6. The *ocd* is first translated into a global description, all duplicate relationships are removed and the remaining relationships are sorted into a well ordered *gcd*. After all areas describing all relationships of the target object have been intersected, but before an object can be inserted, it should be checked whether there are further influencing objects whose relationships are not presented in the *gcd*. Influencing objects are all objects that are north, east, south, or west of the target region.

The relationship of the target object to the influencing object is calculated by reasoning. Considering the possible coarse relationship provided by the composition table, enough space to fulfill all possible basic relationships the coarse relationship consists of is allocated. After the object's area has been established it needs to be checked for *interfering* objects. Interfering objects are influencing objects that have not yet been considered and that are west, east, north, or south of the allocated area for the target object. Because the target's relation to these objects is so far unknown they must be moved to positions where it is still possible to establish all relationships. As the interfering objects are influencing objects, their impact on the target object's area is calculated next. A reconstructed configuration using the set G_{16} of global relations is called a coarse configuration, formally described in definition 9.2.

Definition 9.2. ***coarse Configuration***

An coarse configuration is a configuration Conf = $\langle$O, P, L, OL, G_{16}, RSD, GCD$\rangle$ where

- O is a finite set of objects
- P is a mapping from O to $\mathbb{N}^+$ reflecting the order in which the objects have been encountered.
- L $\subseteq \mathbb{N}^+ \times \mathbb{N}^+$, representing the set of points in the Cartesian plane with integer coordinates.
- OL is a mapping from O to L x L x L x L describing the four edges of an objects area X_{min}, X_{max}, Y_{min}, Y_{max}
  OL(o) = all <i,j> $\in$ L: $(X_{min}(o) \leq i \leq X_{max}(o)) \wedge (Y_{min}(o) \leq j \leq Y_{max}(o))$ o $\in$ O.
- G_{16} is a finite set of binary predicates, defined in definition 9.1
- RSD is a finite set of tuples of the form <o, OL(o)>: o $\in$ O.
- GCD $\subseteq$ O$\times G_{16}\times$O, the global configuration description.

9.4 An example reconstruction with coarse objects

The reconstruction process establishes a reconstruction with all objects' relationships, that are mentioned within the *gcd*. Relationships to influencing objects are handled by the implicit nondeterministic approach to allocate as much space as needed for an object, in order that all possible basic relationships can be fulfilled. However, it might be possible to send the observer to certain places to look for specific relationships that have not been provided within the original *ocd*. When these further relationships arrive, the reconstruction established so far can be improved by adding the new information and diminishing the areas allocated for coarse objects.

9.4.1 Reconstruction using coarse objects

The following example shown in figure 9.5 demonstrates how space is allocated for coarse objects. Suppose the incomplete *gcd* for an object configuration is: {(*2 NW 1*), (*3 SE 1*), (*4 E 1*), (*5 NE 1*), (*6 NE 5*), (*7 E 5*), (*7 SE 6*)}. The first three objects are handled in the same way as before, shown in figure 9.5a). Object *4* has to be inserted *W* of object *1* whereby object *2* becomes an influencing object on object *4*'s position. Because there is no entry in the *gcd* that clarifies the relationship of object *4* to object *2*, reasoning is used to conclude that object *4* is *SOUTH* of object *2*. This means that at least three cells need to be allocated for object *4*, covering the three basic relationships *SE*, *S* and *SW* that the coarse relation *SOUTH* is composed of. Space to fulfill the relationships *S* and *SW* is available in the reconstruction area but the relationship *SE* can currently not be fulfilled. Therefore, object *2* has to be moved in order to provide the space. The grey area in figure 9.5b) shows the three cells allocated for object *4*. Object *4* will only be in one of them but while it is not known in which one, the reconstruction represents object *4* by the whole area of its potential position.

The next entry gives the information that object *5* is *NE* of object *1*, which makes object *2* an influencing object. As the *gcd* does not contain any information of their relationship the reasoning process comes to the result that object *5* is *EAST* of object *2*. To fulfill this relation, three cells in the vertical dimension, as shown in figure 9.5c), need to be occupied by object *5*. In addition, object *3* is also an influencing object whose relationship is also unavailable from the *gcd*. Therefore, the reasoning process concludes that object *5* is *NORTH* of object *3*. This relation alone would be fulfilled by occupying three cells but together with the relationship (*5 EAST 2*) an area of nine cells has to be preliminarily allocated for the possible position of object *5*. In order to fulfill the possible relationship (*5 NW 3*), which is part of (*5 NORTH 3*) object *3* has to be moved to the right before the area representing object *5* can be completely inserted into the reconstruction, shown in figure 9.5d).

To place object *6* correctly without knowing exactly where object *5* will be placed, it is inserted *NE* of the whole area that has been allocated for object *5*, as shown in figure 9.5e). If later on, the relationships between object *5* and objects *3* and *2* become available and object *5* can be placed in its unambiguous position, this will not influence the position of object *6*. In any case it will be *NE* of object *5*. The insertion of object *7 SE* of object *6* and *E* of object *5* is not as simple. Depending on the real position of object *5*, there are three possible alternatives for object *7*. This means that object *7*'s position is dependent on object *5*'s. In order to cover this, an area containing all possible positions that object *7* might have depending on object *5*'s position, is allocated for object *7*, which is also depicted in figure 9.5e). This figure shows the result of the reconstruction of the configuration described by the *gcd* above.

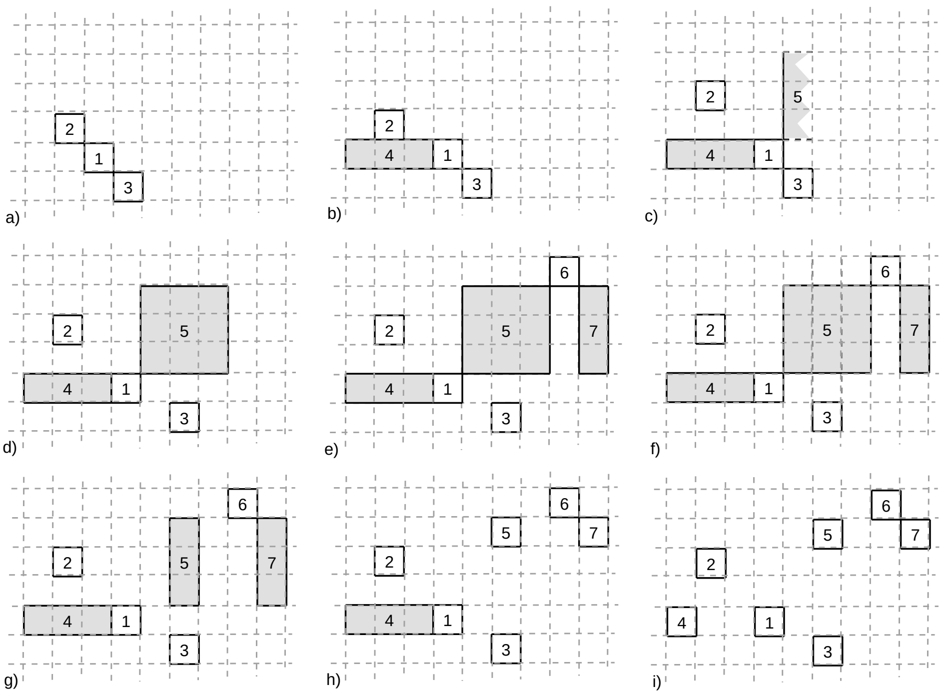


Figure 9.5: *A reconstruction example with coarse object relationships. a) to e) The reconstruction process from the incomplete gcd: (2 NW 1), (3 SE 1), (4 E 1), (5 NE 1), (6 NW 5), (7 E 5), (7 SE 6), where an object with at an ambiguous position is represented by the smallest area that provides enough space that every possible relationship for the so far unknown relationships could be fulfilled. f) to i) The so far missing relationships (5 N 3), (5 NE 2), and (4 SW 2) have become available and the reconstruction is modified by reducing the allocated areas for objects successively to one cell only, taking into account to even resize the area for object 7 which is dependent on object 5.*

9.4.2 Reduction of coarse objects' areas

Assuming that later on new information about the missing object relationships becomes available, the reconstruction can be improved. Suppose that the relationships (*5 N 3*), (*5 NE 2*), and (*4 SW 2*) become available and are handled in the same order. The first relationship leads to an intersection of the allocated area so far for object *5* with the *N* region of object *3*, shown in figure 9.5f). Object *5*'s possible position is now restricted within this intersection, shown in figure 9.5g). The additional relationship (*5 NE 2*) is handled the same way. The intersection of the area allocated for object *5* and the area *NE* of object *2* is calculated. The result represents the remaining possible area for object *5*, which in this case is unambiguous.

Object *7* is the only object that has been inserted into the reconstruction after object *5* as a dependent object. The relationship (*7 E 5*) has to be fulfilled and therefore the area previously allocated for object *7* is now intersected with the area *E* of object *5* at its new position and the intersection becomes the remaining possible space for object *7* which in this case is also unambiguous. The result of the improved reconstruction process so far is shown in figure 9.5h) where only object *4*'s position is still uncertain. The last relationship, (*4 SW 2*), resolves this ambiguity and the complete reconstruction is finally available in figure 9.5i).

9.5 Further necessities to cover all possible cases

The previous example summarized the techniques used to reconstruct object configurations from underspecified *ocd*'s into a global frame of reference taking all possibilities for unknown object positions into account. To be able to cover all possible situations that can arise using underspecified *ocd*s, some further strategies need to be introduced.

- Several objects can have unknown relationships to the same influencing object, which leads to overlapping object areas. In the example in figure 9.6a) object *3* and object *4* are both *NE* of reference object *1*, furthermore the relationship *4 E* of *3* is known, but neither the relationships of object *3* to object *2* or object *4* to object *2* are known. Object *2* is an influencing object for both and reasoning leads to the conclusion that *3* and *4* are both *NORTH* of object *2*. The space allocated for both objects must keep the possibilities open that both objects are to the *NW* of object *2* or to the *NE* of object *2*. Therefore, the areas of the two objects overlap.

- The previous case includes the necessity to resize a coarse object's area after the object has been placed into the reconstruction. In the example scenario above object *3* is most likely to be first inserted *NORTH* of object *2* allocating three cells for it. After the information *4* is E of *3* and *NORTH* of *2* is available the allocated space for object *3* is no longer sufficient.

Object *4* must have the possibility to be to the *NW* of object *2* and object *3* needs to be *W* of object *4*. Therefore one more cell needs to be allocated to the left of the currently allocated area for object *3* allowing both objects to be *NW* of object *2*. In addition, object *4* also allocates four cells.

Figure 9.6b) shows three objects overlapping. All three objects are in the relationship *NORTH* to the influencing object *2*. It must be possible for all of them to end up *NW* as well as *NE* of it. Therefore, three cells to the *NW* and a further three cells to the *NE* of the influencing object have to be allocated and the objects' areas overlap. As in the previous example object *3* which has been inserted first will be resized when object *4* is inserted. Both objects have to be resized again when object *5* appears.

- The object representation as areas of all possible locations of an object includes the possibility that just a part of an areal object influences a target object's position. This describes the case where the object so far represented as an area of all possible positions ends up at a certain single cell and influences another object's position. This potential influence needs to be taken into account in the reconstruction. Whether the areal object turns out ultimately to be at an influencing position or not, the reconstruction must be qualitatively correct during all steps in the reconstruction process, and therefore all possibilities have to be considered in the same reconstruction.

  This problem cannot be solved by adjusting the newly introduced object. Instead, the influencing coarse object has to be resized to cover all possible relationships to the target object. In the example in figure 9.7a) object *3*'s relationship to object *1* is *NORTH*. When object *4* is inserted *W* of object *1*, shown in figure 9.7b) the case where object *3* is in fact *NW* of object *1* needs to be considered because it will influence the position of object *4*. Reasoning provides the relationship (*4 SOUTH 3*).

  In this case, object *4* will either be *SE*, *S* or *SW* of object *3*. In other words object *3* will either be *NE*, *N* or *NW* of object *4*. In this particular case object *3* is an area object and only a part of it is inside the region *NW 1*, which is the inline-region to *W 1* and therefore just this part of object *3* is influencing object *4*'s position. For the correct reconstruction object *3* has to be resized. Allocating more space for object *4* will never solve the problem as object *4* cannot be *SE* of object *3* and still be inside the region *W* of object *1*. Note, that the reconstruction would look the same were the objects' relationships given in a different order: The *gcd*: {(*2 NE 1*), (*4 W 1*), (*3 NW 2*)} would lead to the same reconstruction, whereby in this case object *3* would be placed last and no object inserted earlier would have to be changed.

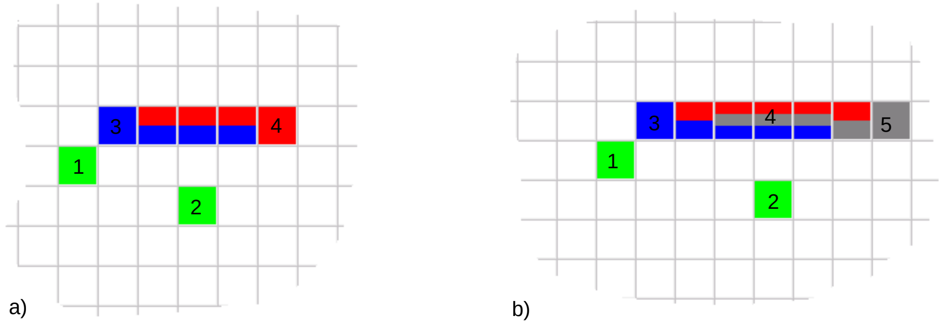


Figure 9.6: *Overlapping coarse object areas. a) gcd: (2 SE 1), (3 NE 1), (4 E 3). Reasoning provides (3 NORTH 2) and (4 NORTH 2). Object 3's area spans all cells that are at least partly red whereas object 4's area is presented by at least partly blue cells. Multicolored cells represent the overlapping part of the two object areas. b) gcd: (2 SE 1), (3 NE 1) (4 E 3), (5 E 4). Reasoning provides (3 NORTH 2), (4 NORTH 2), and (5 NORTH 2). The area occupied by object 3 is presented in red, the area occupied by object 4 in gray and the area occupied by object 5 is shown in blue. Multicolored cells represent the overlapping parts of the areas.*

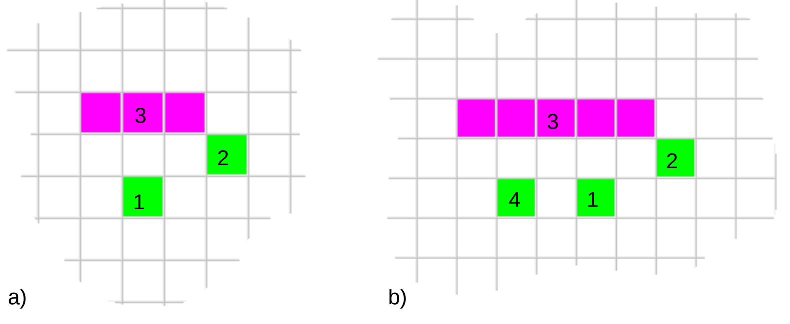


Figure 9.7: *The reconstruction from the gcd: (2 NE 1), (3 NW 2), (4 W 1). a) The first two entries from the gcd have been taken into account. After placing the objects 1 and 2 reasoning has been used to provide the relationship (3 NORTH 1) which lead to the areal representation of object 3. b) The next gcd entry is handled. Object 4 has to be W of object 1 which makes the part of object 3 that overlaps with the area NW of object 2 to an influencing object of object 4's position. Therefore, object 3 has to be adjusted to cover the relationship (4 SOUTH 3) that reasoning has provided for the relationship of object 4 to its influencing object 3.*

This chapter illustrated how neighbourhood-based reasoning can be applied to QuaDRO in order to handle underspecified object configuration descriptions. However, the techniques presented are only partly implemented so far. The following chapter will, besides summarizing and discussing the work presented in this thesis, give a further outlook on future plans to integrate the techniques discussed.

Chapter 10

Summary, discussion, and future work

10.1 Summary

This thesis describes the results of a state of the art research project in the area of qualitative spatial reasoning (QSR). Its overall intention is to support and alleviate human-machine communication about object configurations, such as, for instance, during rescue missions. The result presented is the QuaDRO representation scheme for qualitatively describing and reconstructing object configurations. The description is obtained from several local viewpoints within the object configuration using different perspectives. The configuration's reconstruction from the provided description takes place in a global frame of reference. The result is an abstract map of the described configuration from the survey perspective.

The suggested method of qualitative object configuration description is inspired by psychological research results about taking perspective and classification of space [Tve99; Lev96; Fra91; Lev82]. It is important that the description contains all spatial information necessary to reconstruct the configuration into a global frame of reference. In order to obtain the description the observer performs a route tour between certain objects that function as reference objects for object relationship descriptions. They further function as the origin of a coordinate system for a momentarily applied absolute frame of reference to describe the observed objects in relation to each other. Eight (nine when the position of the reference object is counted itself) positional relations are used to describe object relationships and eight direction classes are used to describe the relative orientation of an object or the observer's direction of movement.

Fourteen qualitative spatial calculi have been analyzed regarding their usability in this description and reconstruction task. For objects represented as line segments, Bipartite Arrangements [Got04] provide the necessary expressibility due to the separation of the objects' position representation and the objects' orientation representation. For point representations, $\mathcal{OPRA}_4$ [Mor05; Mor06; Dyl06] is an applicable choice. Unfortunately, no rectangle representation has been found that provides enough expressibility. However, rectangle representations seem to be cognitively adequate for this task. The representation scheme QuaDRO therefore provides a rectangle representation of the desired expressivity.

A qualitative grid underlies the reconstruction process. For most of the results presented here, only objects of the same size, each occupying one grid cell have been taken into account. Otherwise the number of cells the objects occupy is accountable for the granularity of expressible object relations. Objects are inserted into the reconstruction following the distance default placing them as close as possible to the objects already represented. Object orientations are grouped into eight orientation classes, four aligned to the grid, using a projection based frame of reference with neutral zone, and four at angles to the grid using the same frame of reference rotated 45°. The reason for using these two different frames of reference is justified by the need to provide an effortless procedure to obtain space within the reconstruction to insert new objects. This procedure requires that all objects that are represented are aligned to the underlying grid.

An object's relationships that are not given in the object configuration description (*ocd*) but still have an impact on the object's position are concluded by reasoning based on conceptual neighbourhoods of relations [Fre91a; Fre92a; Fre92c; Zim96]. The reasoning process provides all possible basic relationships that the objects could have to each other. In order to assemble only one reconstruction, coarse relationships are treated as though all contained basic relationships were concurrently true while the ambiguity remains.

10.2 Discussion

During the system development, decisions had to be made to what extent to be responsive to psychological results of typical human behavior in configuration description and to what extent to manufacture an easy reconstruction process. Both components are important in order to develop a representation scheme that is usable by a person from each side of the process. Nevertheless, these two aims are conflicting. Tversky, Lee and Mainwaring [Tve99] experienced that people accommodate the acceptable amount of inconvenience according to the cognitive load the task requires from their communication partners. Therefore, it seems reasonable to balance the endeavors on both sides. The following list discusses decisions and compromises made.

- The presented representation scheme provides the designated results. The object configuration, described from different local viewpoints is represented in a global frame of reference representing all qualitative positional information requested. The reconstruction is equal to a reconstruction established based on the same qualitative relations provided from a survey perspective.

The result is an abstracted map of the object configuration. Distance information is not explicitly included. Therefore, the reconstruction might provide a more condensed picture of the configuration than a true-to-scale representation.

- The representation scheme allows for eight positional relations (different from the reference object) and eight orientations. To describe these 64 combinations only eight basic relations are necessary. The relation names are cognitively easy to understand and in situations where an object overlaps several position classes at the same time its relationship can be described by a combination of basic relations.

- The description process uses typical human strategies and is therefore not awkward for a person to assemble.

  However, the observer needs to provide the description in a structured way and is restricted to the suggested procedures.

- The observer only needs a fraction of global orientation information knowledge in comparison to what would normally be needed to describe an object configuration in a survey-perspective-manner. In order to provide the same expressibility this would, for instance, be eight cardinal direction classes.

  Here the observer only needs to know if an orientation is aligned with an assumed underlying coordinate system or at an angle to it.

- The reconstruction process allows the insertion of objects anywhere in the reconstruction with very little effort. No allowance has to be made to keep the already represented relationships correct while space for objects that have to be inserted is made. However, two problems arise

  - To maintain this advantage even when eight different object orientations are represented, two different frames of reference have to be used. The 45° rotated frame of reference used for objects with orientations at angles to the grid might appear nonintuitive. However, for use in a small-scale environment that already has the reconstruction grid as its underlying global frame of reference, wherein all objects are represented aligned to the grid, its use does not appear difficult.
  - Even though the reconstruction process is simple and intuitive, it does not allow for a simple reconstruction with just pen and paper.

10.3 Future work

The results presented are satisfying as they provide a solution for the initial problem. At the same time they also encourage the continuation of this project's research. Several ideas that had to be left aside in order to condense the work into a reasonable amount for a Ph.D. project remain to be addressed, and several further ideas arose during the work. The following is an (incomplete) list of the most discussed continuations.

1. Coarse reasoning needs to be extended in two ways:
   - The observer is allowed to provide coarse object relationships.
   - Using a chain of reasoning steps including coarse relations in order to obtain information about missing relations in an underspecified *ocd*.
2. Distance information is not included in the representation scheme. Nevertheless some distance information is implicitly available by the number of objects that are between two objects that do not have to be adjacent to each other. It is a challenge to explore how much explicitly stated distance information is necessary to improve the reconstructed object configuration map up to the level where it could be used in the same way as a to a true-to-scale representation.
3. Within this thesis, only static objects have been considered. The attempt to extend this approach to similar sized moving objects has already been started in [Ste06b]. The hope is to include this work in the representation scheme prototype and to broaden it to different sized objects and uncertainty about position and movement in space over time.
4. The representation scheme is brand new. A relatively small prototype has been implemented that functions as a proof of concept for the strategies and techniques developed. The desire is to implement the representation scheme on a true to life scale and deploy it to real outdoor (and even indoor) missions where it can be properly tested.
   - This includes representing objects of different sizes, which will not be difficult to achieve, considering that already objects of uncertain positions are represented as the area of all their possible positions. These areas are in many aspects already treated as objects of different sizes.
   - It further includes using the already provided ability of combined relations with an underlying adjustable reconstruction grid. This is necessary to describe relationships of different sized objects. Nevertheless, it improves expressibility which bears the possibility to even describe relationships of equally sized objects with higher granularity.

Bibliography

[All83] J. F. Allen. Maintaining knowledge about temporal intervals. In *CACM*, vol. 26, pp. 832–843. 1983.

[And05] J. P. Andersson. Hazard: A framework towards connectiong artificial intelligence and robotics. In *IJCAI: Workshop on Reasoning, Representation and Learning in Computer Games*. Edinburgh, Scotland, 2005.

[Are03] M. Arens and H.-H. Nagel. Behavioral knowledge representation for the understanding and creation of video sequences. In A. Günter, R. Kruse, and B. Neumann, editors, *Proceedings of KI 2003: Advances in Artificial Intelligence*, LNCS, pp. 149–163. Springer, Hamburg, Germany, 15-18 September 2003.

[Bal98] P. Balbani, J.-F. Condotta, and L. Cerro. A model for reasoning about bidimensional temporal relations. In A. G. Cohn, L. Schubert, and S. C. Shapiro, editors, *Principles of Knowledge Representation and Reasoning*, pp. 124 – 130. Morgan Kaufmann, Trento, Italy, June 1998.

[Bar02] T. Barkowsky. *Mental representation and processing of geographic knowledge - A computational approach.* Springer, Berlin, 2002.

[Bar05] T. Barkowsky, C. Freksa, M. Hegarty, and R. Lowe. Reasoning with mental and external diagrams: Computational modeling and spatial assistance. In T. Barkowsky, C. Freksa, M. Hegarty, and R. Lowe, editors, *Reasoning with Mental and External Diagrams: Computational Modeling and Spatial Assistance*, pp. vii–x. AAAI Press, Stanford, California, 21-23 March 2005.

[Ber05] S. Bertel. Show me how you act on a diagram and I'll tell you what you think (or: Spatial structures as organizing schemes in collaborative human-computer reasoning). In *Reasoning with Mental and External Diagrams: Computational Modelling and Spatial Assistance*, Papers from the 2005 AAAI Spring Symposium. 2005.

[Boe06] J. Boeddinghaus, M. Ragni, M. Knauff, and B. Nebel. Simulating spatial reasoning using ACT-R. In *Proceedings of the ICCM 06*, pp. 62–67. Lawrence Erlbaum Associates, Mahwah, NJ, 2006.

[Bro00] P. Brown and S. C. Levinson. *Frames of Spatial Reference and Their Acquisition in Tenejapan Tzetal*, chap. 8, pp. 167–197. Erlbaum, 2000.

[Coh01] A. G. Cohn and S. Hazarika. Qualitative spatial representation and reasoning: An overview. In *Fundamenta Informaticae*, vol. 46, pp. 2–32. IOS Press, 2001.

[Con04] G. Conte, S. Duranti, and T. Merz. Dynamic 3d path following for an autonomous helicopter. In *Proceeding of the 5th IFAC Symposium on Intelligent Autonomous Vehicles*. Lisbon, Portugal, 2004.

[Doh00] P. Doherty, G. Granlund, K. Kuchcinski, E. Sandewall, K. Nordberg, E. Skarman, and J. Wiklund. The WITAS unmanned areal vehicle project. In *Proceedings of the 14th European Conference on Artificial Intelligence*, pp. 747–755. Berlin, August 2000.

[Doh04] P. Doherty, P. Haslum, F. Heintz, T. Merz, T. Persson, and B. Wingman. A distributed architecture for intelligent unmanned aerial vehicle experimentation. In *Proceedings of the 7th International Symposium on Distributed Autonomous Robotic Systems*. 2004.

[Dur07a] S. Duranti and G. Conte. In-flight identification of the augmented flight dynamics of the Rmax unmanned helicopter. In *Proceeding of the 17th IFAC Symposium on Automatic Control in Aerospace*. Toulouse, France, 2007.

[Dur07b] S. Duranti, G. Conte, D. Lundström, P. Rudol, M. Wzorek, and P. Doherty. Linkmav, a prototype rotary wing micro aerial vehicle. In *Proceeding of the 17th IFAC Symposium on Automatic Control in Aerospace*. Toulouse, France, 2007.

[Dyl05] F. Dylla and R. Moratz. Exploiting qualitative spatial neighborhoods in the situation calculus. In *Spatial Cognition IV Reasoning, Action, Interaction*, Lecture Notes in Artificial Intelligence, pp. 304–322. 2005.

[Dyl06] F. Dylla and J. O. Wallgrün. On generalizing orientation information in OPRAm. In *Proceedings of the 29th German conference on Artificial Intelligence (KI 2006)*, Lecture Notes in Artificial Intelligence. Springer; Berlin, 2006.

[Ege91] M. J. Egenhofer. Reasoning about binary topological relations (extended abstract). In O. Günther and H.-J. Schenk, editors, *Advances in Spatial Databases*, LNCS 525, pp. 143–160. Springer-Verlag, Zurich, Switzerland, August 1991.

[Eli05a] K. Eliasson. An integrated discourse model for a case-based reasoning dialogue system. In *SAIS Workshop*. 2005.

[Eli05b] K. Eliasson. Integrating a discourse model with a learning case-based reasoning system. In *Ninth Workshop on the Semantics and Pragmatics of Dialogue*. 2005.

[Eli05c] K. Eliasson. Towards a robotic dialogue system with learning and planning capabilities. In *4th Workshop on Knowledge and Reasoning in Practical Dialogue Systems*. 2005.

[Eli06] K. Eliasson. *The Use of Case-Based Reasoning in a Human-Robot Dialog System*. Licenciate thesis, Department of Computer and Information Science, Linköpings Universitet, SE-58183 Linköping, Sweden, 2006.

[Eli07] K. Eliasson. Case-based techniques unsed for dialogue understanding and planning in a human-robot dialogue system. In *Proceedings of the Twentieth International Joint Conference on Artificial Intelligence*, pp. 1600–1605. 2007.

[Fer00] J. Fernyhough, A. G. Cohn, and D. C. Hogg. Constructing qualitative event models automatically from video input. In *Image and Vision Computing*, vol. 18, pp. 81–103. 2000.

[For97] K. Forbus. Qualitative reasoning. In *The Computer Science and Engineering Handbook*, pp. 715 –733. 1997.

[Fra90] N. Franklin and B. Tversky. Searching imagined environments. *Experimental Psychology: Generall 119*, pp. 63–76, 1990.

[Fra91] A. U. Frank. Qualitative spatial reasoning with cardinal directions. In H. Kaindl, editor, *Seventh Austrian Conference on Artificial Intelligence*, Informatik Fachberichte, pp. 157–167. Wien, Austria, September 1991.

[Fre91a] C. Freksa. Conceptual neighborhood and its role in temporal and spatial reasoning. In M. Singh and L. Trave-Massuyès, editors, *Proceedings of the IMACS Workshop on Decision Support Systems and Qualitative Reasoning*, pp. 181–187. Amsterdam, 1991.

[Fre91b] C. Freksa. Qualitative spatial reasoning. In D. M. Mark and A. U. Frank, editors, *Cognitive and Linguistic Aspects of Geographic Space*, pp. 361–371. Kluver Academic Publishers, 1991.

[Fre92a] C. Freksa. Temporal reasoning based on semi-intervals. In *Artificial Intelligence*, vol. 54, pp. 199–227. 1992.

[Fre92b] C. Freksa. Using orientation information for qualitative spatial reasoning. In *Proceedings of the International Conference GIS-From Space to Territory: Theories and Methods of Spatio-Temporal Reasoning*, LNCS. Pisa, 21-23 September 1992.

[Fre92c] C. Freksa. Using orientation information for qualitative spatial reasoning. In A. U. Frank, I. Campari, and U. Formentini, editors, *Theories and Methods of Spatio-Temporal Reasoning in Geographic Space*, no. 639 in LNCD, pp. 162–178. Springer-Verlag, Berlin, 1992.

[Fre93] C. Freksa and R. Röhrig. Dimensions of qualitative spatial reasoning. In N. Carret and M. Singh, editors, *Qualitative Reasoning in Decision Technologies, Proceedings QUADRET '93)*, pp. 483–492. CIMNE, Barcelona, June 1993.

[Gla92] J. Glasgow and D. Papadias. Computational imagery. In *Cognitive Science*, 16, pp. 355–394. 1992.

[Got04] B. Gottfried. Reasoning about intervals in two dimensions. In W. Thissen, P. Wieringa, M. Pantic, and M. Ludema, editors, *Poceedings of the IEEE International Conference on Systems, Man and Cybernetics*. Omnipress, The Netherlands, October 2004.

[Goy00a] R. K. Goyal. *Similarity Assessment for Cardinal Direction between Extended Spatial Objects*. Ph.D. thesis, University of Maine, May 2000.

[Goy00b] R. K. Goyal and M. J. Egenhofer. Consistent queries over cardinal directions across different levels of detail. In A. M. Tjoa, R. Wagner, and A. Al-Zobaidie, editors, *11th International Workshop on Database and Expert System Applications*, IEEE Computer Society, pp. 876–880. Greenwich, UK, September 2000.

[Goy01] R. K. Goyal and M. J. Egenhofer. Similarity of cardinal directions. In C. S. Jensen, editor, *SSTD 2001*, LNCS 2121, pp. 36–55. Springer-Verlag Berlin Heidelberg, 2001.

[Güs89] H. W. Güsgen. Spatial reasoning based on Allen's temporal logic. Tr-89-049, International Computer Science Institute, Berkeley, California, 1989.

[Hak94] G. Hake and D. Grünreich. *Kartographie*. Walter de Gruyter & Co., 1994. 7. Auflage.

[Hei04a] F. Heintz and P. Doherty. DyKnow: A framework for processing dynamic knowledge and object structures in autonomous systems. In *International Workshop on Monitoring, Security, and Rescue Techniques in Multiagent Systems (MSRAS)*. Plock, Poland, 2004.

[Hei04b] F. Heintz and P. Doherty. DyKnow: An approach to middleware for knowledge processing. In *Journal of Intelligent Fuzzy Systems*, vol. 15. 2004.

[Hei04c] F. Heintz and P. Doherty. Managing dynamic object structures using hypothesis generation and validation. In *AAAI Workshop on Anchoring Symbols to Sensor Data*. San Jose, USA, 2004.

[Hei05] F. Heintz and P. Doherty. A knowledge processing middleware framework and its relation to the JDL data fusion model. *Journal of Intelligent Fuzzy Systems*, 2005.

[Hei07a] F. Heintz, P. Rudol, and P. Doherty. Bridging the sense-reasoning gap using the knowledge processing middleware DyKnow. In *Proceedings of the 30th German Conference on Artificial Intelligence*. Osnabrück, Germany, 10-13 September 2007. Extended abstract.

[Hei07b] F. Heintz, P. Rudol, and P. Doherty. From images to traffic behaviours - A UAV tracking and monitoring application. In *Proceedings of the The 10th International Conference on Information Fusion*. Quebec, Canada, 9-12 July 2007.

[Her94a] D. Hernandez. *Qualitative Representation of Spatial Knowledge*. Lecture Notes in Artificial Intelligence. Springer-Verlag, 1994.

[Her94b] G. Herzog and P. Wazinski. Visual translator: Linking perceptions and natural language descriptions. In *Artificial Intelligence Review, 8*, pp. 175–187. 1994.

[Hil82] C. Hill. Up/down, front/back, left/right: A contrastive study of Hausa and English. In J. Weissenborn and W. Klein, editors, *Here and There: Crosslinguistic Studies on Deixis and Demonstration*, pp. 13–42. John Benjamins Press, Amsterdam, 1982.

[Jah04] G. Jahn, P. N. Johnson-Laird, and M. Knauff. Reasoning about consistency with spatial mental models: Hidden and obvious indeterminacy in spatial descriptions. In C. Freksa, M. Knauff, B. Krieg-Brückner, B. Nebel, and T. Barkowsky, editors, *Spatial Cognition IV - Reasoning, Action, Interaction. International Conference Spatial Cognition 2004*, pp. 165–180. Springer, Berlin, Frauenchiemsee, Germany, October 2004.

[JL91] P. N. Johnson-Laird and R. M. J. Byrne. *Deduction*. Erlbaum, Hove (UK), 1991.

[JL01] P. N. Johnson-Laird. Mental models and deduction. In *Trends in Cognitive Science*, vol. 5. Elsevier Science Ltd., October 2001.

[Joh05] R. Johansson, A. Berglund, M. Danielsson, and P. Nugues. Automatic text-to-scene conversion in the traffic accident domain. In L. Pack Kaelbling and A. Saffiotti, editors, *Proceedings of the Nineteenth International Joint Conference on Artificial Intelligence*, pp. 1073–1078. Professional Book Center 2005, Edinburgh, 30 July - 5 August , 2005.

[Kad04a] B. Kadmiry and D. Driankov. A fuzzy flight controller combining linguistic and model-based fuzzy control. In *Fuzzy Sets Systems Journal (FSS)*, vol. 146 of *ISSN: 0165-0114*, pp. 313–347. Elsevier, Netherlands, 2004.

[Kad04b] B. Kadmiry and D. Driankov. Fuzzy gain scheduler for attitude control of an unmanned helicopter. In *(Special issue) on IEEE Transactions on Fuzzy Systems Journal (TFS)*, vol. 12 of *ISSN: 0163-6706*, pp. 502–515. IEEE, August 2004.

[Kli03a] A. Klippel. Wayfinding choremes. In W. Kuhn, M. F. Worboys, and S. Timpf, editors, *Spatial Information Theory: Foundations of Geographic Information Science*, pp. 320–334. Springer, Berlin, 2003.

[Kli03b] A. Klippel. *Wayfinding choremes. Conceptualizing wayfinding and route direction elements.* Ph.D. thesis, Universität Bremen, Bremen, 2003.

[Kna95] M. Knauff, R. Rauh, and C. Schlieder. Preferred mental models in qualitative spatial reasoning: A cognitive assessment of Allens's calculus. In *Proceedings of the Seventeenth Annual Conference of the Cognitive Science Society*, pp. 200–2005. Lawrence Erlbaum Associates, Mahwah, NJ, 1995.

[Kos05] S. M. Kosslyn. Mental images and the brain. In *Cognitive Neuropsychology*, vol. 22, pp. 333–347. Psychology Press Ltd., 2005.

[Lem01a] O. Lemon, A. Bracy, A. Gruenstein, and S. Peters. Information states in a multi-modal dialogue system for human-robot conversation. In *Proceedings of the 5th Workshop on Formal Semantics and Pragmatics of Dialogue (Bi-Dialog 2001)*. 2001.

[Lem01b] O. Lemon, A. Bracy, A. Gruenstein, and S. Peters. The WITAS multimodal dialogue system I. In *Proceedings of the EuroSpeech 2001*. 2001.

[Lem02] O. Lemon, A. Gruenstein, and S. Peters. Collaborative activities and multi-tasking in dialogue systems. In *Traitment automatique des langues,*, vol. 43, pp. 131–154. 2002. Special issue on dialogue.

[Lev82] W. J. M. Levelt. *Speech, Place and Action*, chap. Cognitive Styles in the Use of Spatial Direction Terms, pp. 251–268. John Wiley & Sons Ltd., 1982.

[Lev96] S. Levinson. Frames of references and molyneux's question: cross-linguistic evidence. In P. Bloom, M. Peterson, L. Nadel, and M. Garret, editors, *Language and Space*, pp. 109–169. MIT Press, 1996.

[Lev03] S. C. Levinson. *Space in Language and Cognition: Explorations in Cognitive Diversity.* Cambridge University Press, West Nyack, NY, USA, 2003.

[Lig93] G. F. Ligozat. Qualitative triangulation for spatial reasoning. In *International conference on Spatial Information Theory. A Theoretical Basis for GIS*, pp. 54–68. Elba, Italy, 1993.

[Lig98] G. Ligozat. Reasoning about cardinal directions. *Journal of Visual Languages and Computing*, vol. 9(1):pp. 23–44, 1998.

[Mac77] A. K. Mackworth. Consistency in networks of relations. *Artificial Intelligence*, vol. 8(1):pp. 99–118, 1977.

[Mer04] T. Merz, S. Duranti, and G. Conte. Autonomous landing of an unmanned aerial helicopter based on vision and inertial sensing. In *Proceeding of the 9th International Symposium on Experimental Robotics (ISER 04).* Singapore, 2004.

[Mer06] T. Merz, P. Rudol, and M. Wzorek. Control system framework for autonomous robots based on extended state machines. In *Proceedings of the International Conference on Autonomic and Autonomous Systems (ICAS 2006).* 2006.

[Mor00] R. Moratz, J. Renz, and D. Wolter. Qualitative spatial reasoning about line segments. In W. Horn, editor, *Proceedings of the 14th European Conference on Artificial Intelligence (ECAI).* Berlin, 2000.

[Mor03a] R. Moratz, B. Nebel, and C. Freksa. Qualitative spatial relation about relative position. In C. Freksa, W. Brauer, C. Habel, and K. Wender, editors, *Spatial Cognition III*, no. 2685 in LNAI, pp. 385–400. Springer, 2003.

[Mor03b] R. Moratz, T. Tenbrink, F. Fischer, and J. Bateman. Spatial knowledge representation for human-robot interaction. In C. Freksa, W. Brauer, C. Habel, and K. Wender, editors, *Spatial Cognition III*, no. 2685 in LNAI. 2003.

[Mor05] R. Moratz, F. Dylla, and L. Frommberger. A relative orientation algebra with adjustable granularity. In *Proceedings of the Workshop on Agents in Real-Time and Dynamic Environments (IJCAI 05)*. 2005.

[Mor06] R. Moratz. Representing relative direction as a binary relation of oriented points. In *ECAI 2006 Proceedings of the 17th European Conference on Artificial Intelligence*. 2006.

[Muk90] A. Mukerjee and G. Joe. A qualitative model for space. In *Proceedings of the AAAI*, pp. 721–727. Boston, 1990.

[Nar97] H. Narayanan. Introduction to diagrammatic reasoning. In *Diagrammatic Reasoning Site*. maintained by M. Anderson. University of Hartford, West Hartford, Connecticut, USA, February 1997. http://zeus.cs.hartford.edu/ãnderson/intro.html.

[Nig05] T. Nigel. *The Stanford Encyclopedia of Philosophy*, chap. Mental Imagery. 2005.

[Nor02a] K. Nordberg, P. Doherty, G. Farnebäck, P.-E. Forssén, G. Granlund, A. Moe, and J. Wiklund. Vision for a UAV helicopter. In *Proceedings of IROS'02, workshop on aerial robotics*. Lausanne, Switzerland, October 2002.

[Nor02b] K. Nordberg, P.-E. Forssén, J. Wiklund, P. Doherty, and P. Andersson. A flexible runtime system for image processing in a distributed computational environment for an unmanned aerial vehicle. In *Proceedings of IWSSIP'02*. Manchester, UK, November 2002.

[Nov86] H.-J. Novak. *Textgenerierung auf der Grundlage visueller Daten: Beschreibungen von Strassenszenen*. Ph.D. thesis, Universität Hamburg, Hamburg, 1986.

[Pap95] D. Papadias, Y. Theodoridis, T. Sellis, and M. J. Egenhofer. Topological relations in the world of minimum bounding rectangles: A study with R-trees. In *Proceedings of the SCM Conference on the Management of Data (SIGMOD)*. ACM Press, San Jose, 1995.

[Pet04] P. Pettersson and P. Doherty. Probabilistic roadmap based path planning for an autonomous unmanned aerial vehicle. In *ICAPS-04 Workshop on Connecting Planning Theory with Practice*. 2004.

[Pet06] P. Pettersson. *Sampling-based Path Planning for an Autonomous Helicopter*. Licenciate thesis, Linköpings universitet, Department of Computer and Information Science, Linköpings universitet, SE-58183 Linköping, Sweden, 2006.

[Rag03] M. Ragni. An arragement calculus, its complexity, and algorithmic properties. In A. Günter, R. Kruse, and B. Neumann, editors, *KI 2003: Advances in AI*, LNAI, pp. 580–590. Springer-Verlag Berlin Heidelberg, 2003.

[Rag05] M. Ragni, M. Knauff, and B. Nebel. A computational model for spatial reasoning with mental models. In B. Bara, B. Barsalou, and M. Bucciarelli, editors, *Proceedings of the 27th Annual Cognitive Science Conference (CogSci05)*, pp. 1064–70. Erlbaum, 2005.

[Rag06] M. Ragni, T. Fangmeier, L. Webber, and M. Knauff. Complexity in spatial reasoning. In *Proceedings of the 28th Annual Cognitive Science Conference (CogSci 2006)*. Lawrence Erlbaum Associates, 2006.

[Rag07] M. Ragni and F. Steffenhagen. Qualitative spatial reasoning: a cognitive and computational approach. In *Proceedings of the 28th Annual Cognitive Science Conference (CogSci 07)*. Lawrence Erlbaum Associates, Mahwah, NJ, 2007.

[Ran92] D. A. Randell, Z. Cui, and A. G. Cohn. A spatial logic based on regions and connection. In *Proceedings of the 3rd International Conference on Knowledge Representation and Reasoning*. Morgan Kaufmann, 1992.

[Rau05] R. Rauh, C. Hagen, M. Knauff, T. Kuss, C. Schlieder, and G. Strube. Preferred and alternative mental models in spatial reasoning. In *Spatial Reasoning & Computation*, vol. 5, pp. 239–269. 2005.

[Ren98] J. Renz and B. Nebel. Efficient methods for qualitative spatial reasoning. In *Proceedings of the 13th European Conference on Artificial Intelligence (ECAI'98)*, pp. 562–566. Brighton, UK, August 1998.

[Ren01] J. Renz and B. Nebel. Efficient methods for qualitative spatial reasoning. *Journal of Artificial Intelligence Research*, (15):pp. 289–318, 2001.

[Ren02] J. Renz. *Qualitative Spatial Reasoning with Topological Information.* LNAI 2293. Springer-Verlag, Berlin Heidelberg, 2002.

[Ren04] J. Renz and D. Mitra. Qualitative direction calculi with arbitrary granularity. In *PRICAI 2004 Proceedings of the 8th Pacific Rim International Confrence on Artificial Intelligence*, pp. 65–74. 2004.

[Ren07] J. Renz and B. Nebel. *Handbook of Spatial Logics*, chap. Qualitative Spatial Reasoning Using Constraint Calculi. Kluwer Academic Publishers, 2007.

[San03a] E. Sandewall, P. Doherty, O. Lemon, and S. Peters. Real-time dialogues with the witas unmanned aerial vehicle. In A. Günter, editor, *Annual German Conference on AI*, pp. 52–63. 2003.

[San03b] E. Sandewall, P. Doherty, O. Lemon, and S. Peters. Words at the right time: Real-time dialogues with the witas unmanned aerial vehicle. In A. Günter, R. Kruse, and B. Neumann, editors, *Proceedings of the 26th German Conference on Artificial Intelligence*, vol. 2821 of *LNAI*, pp. 52–63. 2003.

[San05] E. Sandewall, H. Lindblom, and B. Husberg. Integration of live video in a system for natural language dialog with a robot, 2005.

[Sch95] C. Schlieder. Reasoning about ordering. In A. Frank and W. Kuhn, editors, *Proceedings of 3rd International Conference on Spatial Information Theory*. 1995.

[Sci04] A. Scivos and B. Nebel. The finest of its class: The natural, point-based ternary calculus LR for qualitative spatial reasoning. In C. Freksa, M. Knauff, B. Krieg-Brükner, B. Nebel, and T. Barkowsky, editors, *Spatial Cognition IV*, LNAI 3343, pp. 283–303. Springer, Frauenchiemsee, Germany, October 2004.

[Ski04] S. Skiadopoulos and M. Koubarakis. Composing cardinal direction relations. *Artificial Intelligence*, (152):pp. 143–171, 2004.

[Ski05] S. Skiadopoulos and M. Koubarakis. On the consistency of cardinal direction constraints. *Artificial Intelligence*, (163):pp. 91–135, 2005.

[Spr01] R. Sproat. Inferring the environment in a text-to-scene conversion system. In *Proceedings of the First International Conference on Knowledge Capture (K-CAP 2001)*, pp. 147–154. ACM, Victoria, BC, Canada, 21-23 October 2001.

[Ste04a] H. J. Steinhauer. The qualitative description of traffic maneuvers. In H. W. Güsgen, F. D. Anger, C. Freksa, G. Ligozat, and R. V. Rodrguez, editors, *Proceedings of Workshop on Spatial and Temporal Reasoning*, 16th European Conference on Artificial Intelligence (ECAI), pp. 141–148. Valencia, Spain, 23-27 August 2004.

[Ste04b] H. J. Steinhauer. A qualitative description of traffic maneuvers. In T. Barkowsky, C. Freksa, M. Knauff, B. Krieg-Brückner, and B. Nebel, editors, *Spatial Cognition 2004: Poster Presentations*, Report Series of the Transregional Collaborative Research Center SFB/TR 8 Spatial Cognition, pp. 39–42. SFB/TR 8 Spatial Cognition, Frauenchiemsee, Germany, October 2004.

[Ste05a] H. J. Steinhauer. A qualitative model for natural language communication about vehicle traffic. In T. Barkowsky, C. Freksa, M. Hegarty, and R. Lowe, editors, *Reasoning with Mental and External Diagrams: Computational Modeling and Spatial Assistance*, pp. 52–57. AAAI Press, Stanford, California, 21-23 March 2005.

[Ste05b] H. J. Steinhauer. Towards a qualitative model for natural language communication about vehicle traffic. In H. W. Güsgen, C. Freksa, and G. Ligozat, editors, *IJCAI-05 Workshop on Spatial and Temporal Reasoning*, pp. 45–51. Edinburgh, Scotland, 30 July - 5 August 2005.

[Ste06a] H. J. Steinhauer. Qualitative communication about object scenes. In *KI-04 Spatial Cognition Doctoral Colloquium*. Bremen, Germany, 14 -19 June 2006.

[Ste06b] H. J. Steinhauer. Qualitative reconstruction and update of an object constellation. In H. W. Güsgen, G. Ligozat, J. Renz, and R. V. Rodriguez, editors, *ECAI-06 Workshop on Spatial and Temporal Reasoning*, pp. 11 – 19. Riva del Garda, Italy, 28 August - 1 September 2006.

[Ste08a] H. J. Steinhauer. Object configuration reconstruction for descriptions using relative and intrinsic reference frames. In *Proceedings of the 18th European Conference on Artificial Intellicence ECAI-08*, pp. 821–822. IOS Press, Patras, Greece, July 2008.

[Ste08b] H. J. Steinhauer. Object configuration reconstruction from incomplete binary object relation descriptions. In A. R. Dengel, K. Berns, and T. M. Breul, editors, *KI 2008: Advances in Artificial Intelligenc*, pp. 348–355. Springer, Kaiserslautern, Germany, September 2008.

[Ten02] T. Tenbrink, K. Fischer, and R. Moratz. Spatial strategies in human-robot communication. *Künstliche Intelligenz, Themenheft Spatial Cognition*, 2002.

[Tve99] B. Tversky, P. Lee, and S. Mainwaring. Why do speakers mix perspectives? *Spatial Cognition and Computation*, vol. 1:pp. 399–412, 1999.

[Wer12] M. Wertheimer. Experimentelle Studien über das Sehen von Bewegung. *Zeitschrift für Psychologie*, (61):pp. 161–265, 1912. English translation in T. Shipley, editor, Classics in Psychology. New York: Philosophical Library, 1961.

[Wer22] M. Wertheimer. Untersuchungen zur Lehre von der Gestalt. I. Prinzipielle Bemerkungen. *Psychologische Forschung*, (1):pp. 47–58, 1922. Abridged English translation in W.D. Ellis, editor, A Source Book of Gestalt Psychology , pp. 71-88. London: Kegan Paul, Trench, Trubner.

[wik] Wikipedia, the free encyclopedia, hausa language. http://en.wikipedia.org/w/index.php?title=Hausa_languageoldid=146785252, accessed 23 August 2007.

[Wil90] H. Wilhelmy. *Kartographie in Stichworten*. Verlag Ferdinand Hirt AG, 1990. 5. Auflage.

[Wzo06a] M. Wzorek, G. Conte, P. Rudol, T. Merz, S. Duranti, and P. Doherty. From motion planning to control - a navigation framework for an autonomous aerial vehicle. In *Proceedings of the 21th Bristol International UAV Systems Conference*. April 2006.

[Wzo06b] M. Wzorek and P. Doherty. The WITAS UAV ground system interface demonstration with a focus on motion and task planning. In *System Demonstration during the International Conference on Automated Planning and Scheduling (ICAPS 2006)*. 2006. Extended Abstract.

[Wzo06c] M. Wzorek, D. Landén, and P. Doherty. GSM technology as a communication media for an autonomous unmanned aerial vehicle. In *Proceedings of the 21th Bristol International UAV Systems Conference*. April 2006.

[Zim96] K. Zimmermann and C. Freksa. Qualitative spatial reasoning using orientation, distance and path knowledge. *Applied Intelligence*, vol. 6:pp. 49–58, 1996.

Department of Computer and Information Science
Linköpings universitet

Dissertations

Linköping Studies in Science and Technology

No 14 **Anders Haraldsson:** A Program Manipulation System Based on Partial Evaluation, 1977, ISBN 91-7372-144-1.

No 17 **Bengt Magnhagen:** Probability Based Verification of Time Margins in Digital Designs, 1977, ISBN 91-7372-157-3.

No 18 **Mats Cedwall:** Semantisk analys av processbeskrivningar i naturligt språk, 1977, ISBN 91-7372-168-9.

No 22 **Jaak Urmi:** A Machine Independent LISP Compiler and its Implications for Ideal Hardware, 1978, ISBN 91-7372-188-3.

No 33 **Tore Risch:** Compilation of Multiple File Queries in a Meta-Database System 1978, ISBN 91-7372-232-4.

No 51 **Erland Jungert:** Synthesizing Database Structures from a User Oriented Data Model, 1980, ISBN 91-7372-387-8.

No 54 **Sture Hägglund:** Contributions to the Development of Methods and Tools for Interactive Design of Applications Software, 1980, ISBN 91-7372-404-1.

No 55 **Pär Emanuelson:** Performance Enhancement in a Well-Structured Pattern Matcher through Partial Evaluation, 1980, ISBN 91-7372-403-3.

No 58 **Bengt Johnsson, Bertil Andersson:** The Human-Computer Interface in Commercial Systems, 1981, ISBN 91-7372-414-9.

No 69 **H. Jan Komorowski:** A Specification of an Abstract Prolog Machine and its Application to Partial Evaluation, 1981, ISBN 91-7372-479-3.

No 71 **René Reboh:** Knowledge Engineering Techniques and Tools for Expert Systems, 1981, ISBN 91-7372-489-0.

No 77 **Östen Oskarsson:** Mechanisms of Modifiability in large Software Systems, 1982, ISBN 91-7372-527-7.

No 94 **Hans Lunell:** Code Generator Writing Systems, 1983, ISBN 91-7372-652-4.

No 97 **Andrzej Lingas:** Advances in Minimum Weight Triangulation, 1983, ISBN 91-7372-660-5.

No 109 **Peter Fritzson:** Towards a Distributed Programming Environment based on Incremental Compilation,1984, ISBN 91-7372-801-2.

No 111 **Erik Tengvald:** The Design of Expert Planning Systems. An Experimental Operations Planning System for Turning, 1984, ISBN 91-7372-805-5.

No 155 **Christos Levcopoulos:** Heuristics for Minimum Decompositions of Polygons, 1987, ISBN 91-7870-133-3.

No 165 **James W. Goodwin:** A Theory and System for Non-Monotonic Reasoning, 1987, ISBN 91-7870-183-X.

No 170 **Zebo Peng:** A Formal Methodology for Automated Synthesis of VLSI Systems, 1987, ISBN 91-7870-225-9.

No 174 **Johan Fagerström:** A Paradigm and System for Design of Distributed Systems, 1988, ISBN 91-7870-301-8.

No 192 **Dimiter Driankov:** Towards a Many Valued Logic of Quantified Belief, 1988, ISBN 91-7870-374-3.

No 213 **Lin Padgham:** Non-Monotonic Inheritance for an Object Oriented Knowledge Base, 1989, ISBN 91-7870-485-5.

No 214 **Tony Larsson:** A Formal Hardware Description and Verification Method, 1989, ISBN 91-7870-517-7.

No 221 **Michael Reinfrank:** Fundamentals and Logical Foundations of Truth Maintenance, 1989, ISBN 91-7870-546-0.

No 239 **Jonas Löwgren:** Knowledge-Based Design Support and Discourse Management in User Interface Management Systems, 1991, ISBN 91-7870-720-X.

No 244 **Henrik Eriksson:** Meta-Tool Support for Knowledge Acquisition, 1991, ISBN 91-7870-746-3.

No 252 **Peter Eklund:** An Epistemic Approach to Interactive Design in Multiple Inheritance Hierarchies,1991, ISBN 91-7870-784-6.

No 258 **Patrick Doherty:** NML3 - A Non-Monotonic Formalism with Explicit Defaults, 1991, ISBN 91-7870-816-8.

No 260 **Nahid Shahmehri:** Generalized Algorithmic Debugging, 1991, ISBN 91-7870-828-1.

No 264 **Nils Dahlbäck:** Representation of Discourse-Cognitive and Computational Aspects, 1992, ISBN 91-7870-850-8.

No 265 **Ulf Nilsson:** Abstract Interpretations and Abstract Machines: Contributions to a Methodology for the Implementation of Logic Programs, 1992, ISBN 91-7870-858-3.

No 270 **Ralph Rönnquist:** Theory and Practice of Tense-bound Object References, 1992, ISBN 91-7870-873-7.

No 273 **Björn Fjellborg:** Pipeline Extraction for VLSI Data Path Synthesis, 1992, ISBN 91-7870-880-X.

No 276 **Staffan Bonnier:** A Formal Basis for Horn Clause Logic with External Polymorphic Functions, 1992, ISBN 91-7870-896-6.

No 277 **Kristian Sandahl:** Developing Knowledge Management Systems with an Active Expert Methodology, 1992, ISBN 91-7870-897-4.

No 281 **Christer Bäckström:** Computational Complexity of Reasoning about Plans, 1992, ISBN 91-7870-979-2.

No 292 **Mats Wirén:** Studies in Incremental Natural Language Analysis, 1992, ISBN 91-7871-027-8.

No 297 **Mariam Kamkar:** Interprocedural Dynamic Slicing with Applications to Debugging and Testing, 1993, ISBN 91-7871-065-0.

No 302 **Tingting Zhang:** A Study in Diagnosis Using Classification and Defaults, 1993, ISBN 91-7871-078-2.

No 312 **Arne Jönsson:** Dialogue Management for Natural Language Interfaces - An Empirical Approach, 1993, ISBN 91-7871-110-X.

No 338 **Simin Nadjm-Tehrani**: Reactive Systems in Physical Environments: Compositional Modelling and Framework for Verification, 1994, ISBN 91-7871-237-8.

No 371 **Bengt Savén:** Business Models for Decision Support and Learning. A Study of Discrete-Event Manufacturing Simulation at Asea/ABB 1968-1993, 1995, ISBN 91-7871-494-X.

No 375 **Ulf Söderman:** Conceptual Modelling of Mode Switching Physical Systems, 1995, ISBN 91-7871-516-4.

No 383 **Andreas Kågedal:** Exploiting Groundness in Logic Programs, 1995, ISBN 91-7871-538-5.

No 396 **George Fodor:** Ontological Control, Description, Identification and Recovery from Problematic Control Situations, 1995, ISBN 91-7871-603-9.

No 413 **Mikael Pettersson:** Compiling Natural Semantics, 1995, ISBN 91-7871-641-1.

No 414 **Xinli Gu:** RT Level Testability Improvement by Testability Analysis and Transformations, 1996, ISBN 91-7871-654-3.

No 416 **Hua Shu:** Distributed Default Reasoning, 1996, ISBN 91-7871-665-9.

No 429 **Jaime Villegas:** Simulation Supported Industrial Training from an Organisational Learning Perspective - Development and Evaluation of the SSIT Method, 1996, ISBN 91-7871-700-0.

No 431 **Peter Jonsson:** Studies in Action Planning: Algorithms and Complexity, 1996, ISBN 91-7871-704-3.

No 437 **Johan Boye:** Directional Types in Logic Programming, 1996, ISBN 91-7871-725-6.

No 439 **Cecilia Sjöberg:** Activities, Voices and Arenas: Participatory Design in Practice, 1996, ISBN 91-7871-728-0.

No 448 **Patrick Lambrix:** Part-Whole Reasoning in Description Logics, 1996, ISBN 91-7871-820-1.

No 452 **Kjell Orsborn:** On Extensible and Object-Relational Database Technology for Finite Element Analysis Applications, 1996, ISBN 91-7871-827-9.

No 459 **Olof Johansson:** Development Environments for Complex Product Models, 1996, ISBN 91-7871-855-4.

No 461 **Lena Strömbäck:** User-Defined Constructions in Unification-Based Formalisms,1997, ISBN 91-7871-857-0.

No 462 **Lars Degerstedt:** Tabulation-based Logic Programming: A Multi-Level View of Query Answering, 1996, ISBN 91-7871-858-9.

No 475 **Fredrik Nilsson:** Strategi och ekonomisk styrning - En studie av hur ekonomiska styrsystem utformas och används efter företagsförvärv, 1997, ISBN 91-7871-914-3.

No 480 **Mikael Lindvall:** An Empirical Study of Requirements-Driven Impact Analysis in Object-Oriented Software Evolution, 1997, ISBN 91-7871-927-5.

No 485 **Göran Forslund**: Opinion-Based Systems: The Cooperative Perspective on Knowledge-Based Decision Support, 1997, ISBN 91-7871-938-0.

No 494 **Martin Sköld**: Active Database Management Systems for Monitoring and Control, 1997, ISBN 91-7219-002-7.

No 495 **Hans Olsén**: Automatic Verification of Petri Nets in a CLP framework, 1997, ISBN 91-7219-011-6.

No 498 **Thomas Drakengren:** Algorithms and Complexity for Temporal and Spatial Formalisms, 1997, ISBN 91-7219-019-1.

No 502 **Jakob Axelsson:** Analysis and Synthesis of Heterogeneous Real-Time Systems, 1997, ISBN 91-7219-035-3.

No 503 **Johan Ringström:** Compiler Generation for Data-Parallel Programming Langugaes from Two-Level Semantics Specifications, 1997, ISBN 91-7219-045-0.

No 512 **Anna Moberg:** Närhet och distans - Studier av kommunikationsmmönster i satellitkontor och flexibla kontor, 1997, ISBN 91-7219-119-8.

No 520 **Mikael Ronström:** Design and Modelling of a Parallel Data Server for Telecom Applications, 1998, ISBN 91-7219-169-4.

No 522 **Niclas Ohlsson:** Towards Effective Fault Prevention - An Empirical Study in Software Engineering, 1998, ISBN 91-7219-176-7.

No 526 **Joachim Karlsson:** A Systematic Approach for Prioritizing Software Requirements, 1998, ISBN 91-7219-184-8.

No 530 **Henrik Nilsson:** Declarative Debugging for Lazy Functional Languages, 1998, ISBN 91-7219-197-x.

No 555 **Jonas Hallberg:** Timing Issues in High-Level Synthesis,1998, ISBN 91-7219-369-7.

No 561 **Ling Lin:** Management of 1-D Sequence Data - From Discrete to Continuous, 1999, ISBN 91-7219-402-2.

No 563 **Eva L Ragnemalm:** Student Modelling based on Collaborative Dialogue with a Learning Companion, 1999, ISBN 91-7219-412-X.

No 567 **Jörgen Lindström:** Does Distance matter? On geographical dispersion in organisations, 1999, ISBN 91-7219-439-1.

No 582 **Vanja Josifovski:** Design, Implementation and

Evaluation of a Distributed Mediator System for Data Integration, 1999, ISBN 91-7219-482-0.

No 589 **Rita Kovordányi**: Modeling and Simulating Inhibitory Mechanisms in Mental Image Reinterpretation - Towards Cooperative Human-Computer Creativity, 1999, ISBN 91-7219-506-1.

No 592 **Mikael Ericsson:** Supporting the Use of Design Knowledge - An Assessment of Commenting Agents, 1999, ISBN 91-7219-532-0.

No 593 **Lars Karlsson:** Actions, Interactions and Narratives, 1999, ISBN 91-7219-534-7.

No 594 **C. G. Mikael Johansson:** Social and Organizational Aspects of Requirements Engineering Methods - A practice-oriented approach, 1999, ISBN 91-7219-541-X.

No 595 **Jörgen Hansson:** Value-Driven Multi-Class Overload Management in Real-Time Database Systems, 1999, ISBN 91-7219-542-8.

No 596 **Niklas Hallberg:** Incorporating User Values in the Design of Information Systems and Services in the Public Sector: A Methods Approach, 1999, ISBN 91-7219-543-6.

No 597 **Vivian Vimarlund:** An Economic Perspective on the Analysis of Impacts of Information Technology: From Case Studies in Health-Care towards General Models and Theories, 1999, ISBN 91-7219-544-4.

No 598 **Johan Jenvald:** Methods and Tools in Computer-Supported Taskforce Training, 1999, ISBN 91-7219-547-9.

No 607 **Magnus Merkel:** Understanding and enhancing translation by parallel text processing, 1999, ISBN 91-7219-614-9.

No 611 **Silvia Coradeschi:** Anchoring symbols to sensory data, 1999, ISBN 91-7219-623-8.

No 613 **Man Lin:** Analysis and Synthesis of Reactive Systems: A Generic Layered Architecture Perspective, 1999, ISBN 91-7219-630-0.

No 618 **Jimmy Tjäder:** Systemimplementering i praktiken - En studie av logiker i fyra projekt, 1999, ISBN 91-7219-657-2.

No 627 **Vadim Engelson:** Tools for Design, Interactive Simulation, and Visualization of Object-Oriented Models in Scientific Computing, 2000, ISBN 91-7219-709-9.

No 637 **Esa Falkenroth:** Database Technology for Control and Simulation, 2000, ISBN 91-7219-766-8.

No 639 **Per-Arne Persson:** Bringing Power and Knowledge Together: Information Systems Design for Autonomy and Control in Command Work, 2000, ISBN 91-7219-796-X.

No 660 **Erik Larsson:** An Integrated System-Level Design for Testability Methodology, 2000, ISBN 91-7219-890-7.

No 688 **Marcus Bjäreland:** Model-based Execution Monitoring, 2001, ISBN 91-7373-016-5.

No 689 **Joakim Gustafsson:** Extending Temporal Action Logic, 2001, ISBN 91-7373-017-3.

No 720 **Carl-Johan Petri:** Organizational Information Provision - Managing Mandatory and Discretionary Use of Information Technology, 2001, ISBN-91-7373-126-9.

No 724 **Paul Scerri:** Designing Agents for Systems with Adjustable Autonomy, 2001, ISBN 91 7373 207 9.

No 725 **Tim Heyer**: Semantic Inspection of Software Artifacts: From Theory to Practice, 2001, ISBN 91 7373 208 7.

No 726 **Pär Carlshamre:** A Usability Perspective on Requirements Engineering - From Methodology to Product Development, 2001, ISBN 91 7373 212 5.

No 732 **Juha Takkinen:** From Information Management to Task Management in Electronic Mail, 2002, ISBN 91 7373 258 3.

No 745 **Johan Åberg:** Live Help Systems: An Approach to Intelligent Help for Web Information Systems, 2002, ISBN 91-7373-311-3.

No 746 **Rego Granlund:** Monitoring Distributed Teamwork Training, 2002, ISBN 91-7373-312-1.

No 757 **Henrik André-Jönsson:** Indexing Strategies for Time Series Data, 2002, ISBN 917373-346-6.

No 747 **Anneli Hagdahl:** Development of IT-suppor-ted Inter-organisational Collaboration - A Case Study in the Swedish Public Sector, 2002, ISBN 91-7373-314-8.

No 749 **Sofie Pilemalm:** Information Technology for Non-Profit Organisations - Extended Participatory Design of an Information System for Trade Union Shop Stewards, 2002, ISBN 91-7373-318-0.

No 765 **Stefan Holmlid:** Adapting users: Towards a theory of use quality, 2002, ISBN 91-7373-397-0.

No 771 **Magnus Morin:** Multimedia Representations of Distributed Tactical Operations, 2002, ISBN 91-7373-421-7.

No 772 **Pawel Pietrzak:** A Type-Based Framework for Locating Errors in Constraint Logic Programs, 2002, ISBN 91-7373-422-5.

No 758 **Erik Berglund:** Library Communication Among Programmers Worldwide, 2002, ISBN 91-7373-349-0.

No 774 **Choong-ho Yi:** Modelling Object-Oriented Dynamic Systems Using a Logic-Based Framework, 2002, ISBN 91-7373-424-1.

No 779 **Mathias Broxvall:** A Study in the Computational Complexity of Temporal Reasoning, 2002, ISBN 91-7373-440-3.

No 793 **Asmus Pandikow:** A Generic Principle for Enabling Interoperability of Structured and Object-Oriented Analysis and Design Tools, 2002, ISBN 91-7373-479-9.

No 785 **Lars Hult:** Publika Informationstjänster. En studie av den Internetbaserade encyklopedins bruksegenskaper, 2003, ISBN 91-7373-461-6.

No 800 **Lars Taxén:** A Framework for the Coordination of Complex Systems´ Development, 2003, ISBN 91-7373-604-X

No 808 **Klas Gäre:** Tre perspektiv på förväntningar och förändringar i samband med införande av informationsystem, 2003, ISBN 91-7373-618-X.

No 821 **Mikael Kindborg:** Concurrent Comics - programming of social agents by children, 2003, ISBN 91-7373-651-1.

No 823 **Christina Ölvingson:** On Development of Information Systems with GIS Functionality in Public Health Informatics: A Requirements Engineering Approach, 2003, ISBN 91-7373-656-2.

No 828 **Tobias Ritzau:** Memory Efficient Hard Real-Time Garbage Collection, 2003, ISBN 91-7373-666-X.

No 833 **Paul Pop:** Analysis and Synthesis of Communication-Intensive Heterogeneous Real-Time Systems, 2003, ISBN 91-7373-683-X.

No 852 **Johan Moe:** Observing the Dynamic Behaviour of Large Distributed Systems to Improve Development and Testing - An Emperical Study in Software Engineering, 2003, ISBN 91-7373-779-8.

No 867 **Erik Herzog:** An Approach to Systems Engineering Tool Data Representation and Exchange, 2004, ISBN 91-7373-929-4.

No 872 **Aseel Berglund:** Augmenting the Remote Control: Studies in Complex Information Navigation for Digital TV, 2004, ISBN 91-7373-940-5.

No 869 **Jo Skåmedal:** Telecommuting's Implications on Travel and Travel Patterns, 2004, ISBN 91-7373-935-9.

No 870 **Linda Askenäs:** The Roles of IT - Studies of Organising when Implementing and Using Enterprise Systems, 2004, ISBN 91-7373-936-7.

No 874 **Annika Flycht-Eriksson:** Design and Use of Ontologies in Information-Providing Dialogue Systems, 2004, ISBN 91-7373-947-2.

No 873 **Peter Bunus:** Debugging Techniques for Equation-Based Languages, 2004, ISBN 91-7373-941-3.

No 876 **Jonas Mellin:** Resource-Predictable and Efficient Monitoring of Events, 2004, ISBN 91-7373-956-1.

No 883 **Magnus Bång:** Computing at the Speed of Paper: Ubiquitous Computing Environments for Healthcare Professionals, 2004, ISBN 91-7373-971-5

No 882 **Robert Eklund:** Disfluency in Swedish human-human and human-machine travel booking dialogues, 2004. ISBN 91-7373-966-9.

No 887 **Anders Lindström:** English and other Foreign Linquistic Elements in Spoken Swedish. Studies of Productive Processes and their Modelling using Finite-State Tools, 2004, ISBN 91-7373-981-2.

No 889 **Zhiping Wang:** Capacity-Constrained Production-inventory systems - Modellling and Analysis in both a traditional and an e-business context, 2004, ISBN 91-85295-08-6.

No 893 **Pernilla Qvarfordt:** Eyes on Multimodal Interaction, 2004, ISBN 91-85295-30-2.

No 910 **Magnus Kald:** In the Borderland between Strategy and Management Control - Theoretical Framework and Empirical Evidence, 2004, ISBN 91-85295-82-5.

No 918 **Jonas Lundberg:** Shaping Electronic News: Genre Perspectives on Interaction Design, 2004, ISBN 91-85297-14-3.

No 900 **Mattias Arvola:** Shades of use: The dynamics of interaction design for sociable use, 2004, ISBN 91-85295-42-6.

No 920 **Luis Alejandro Cortés:** Verification and Scheduling Techniques for Real-Time Embedded Systems, 2004, ISBN 91-85297-21-6.

No 929 **Diana Szentivanyi:** Performance Studies of Fault-Tolerant Middleware, 2005, ISBN 91-85297-58-5.

No 933 **Mikael Cäker:** Management Accounting as Constructing and Opposing Customer Focus: Three Case Studies on Management Accounting and Customer Relations, 2005, ISBN 91-85297-64-X.

No 937 **Jonas Kvarnström:** TALplanner and Other Extensions to Temporal Action Logic, 2005, ISBN 91-85297-75-5.

No 938 **Bourhane Kadmiry:** Fuzzy Gain-Scheduled Visual Servoing for Unmanned Helicopter, 2005, ISBN 91-85297-76-3.

No 945 **Gert Jervan:** Hybrid Built-In Self-Test and Test Generation Techniques for Digital Systems, 2005, ISBN: 91-85297-97-6.

No 946 **Anders Arpteg:** Intelligent Semi-Structured Information Extraction, 2005, ISBN 91-85297-98-4.

No 947 **Ola Angelsmark:** Constructing Algorithms for Constraint Satisfaction and Related Problems - Methods and Applications, 2005, ISBN 91-85297-99-2.

No 963 **Calin Curescu:** Utility-based Optimisation of Resource Allocation for Wireless Networks, 2005. ISBN 91-85457-07-8.

No 972 **Björn Johansson:** Joint Control in Dynamic Situations, 2005, ISBN 91-85457-31-0.

No 974 **Dan Lawesson:** An Approach to Diagnosability Analysis for Interacting Finite State Systems, 2005, ISBN 91-85457-39-6.

No 979 **Claudiu Duma:** Security and Trust Mechanisms for Groups in Distributed Services, 2005, ISBN 91-85457-54-X.

No 983 **Sorin Manolache:** Analysis and Optimisation of Real-Time Systems with Stochastic Behaviour, 2005, ISBN 91-85457-60-4.

No 986 **Yuxiao Zhao:** Standards-Based Application Integration for Business-to-Business Communications, 2005, ISBN 91-85457-66-3.

No 1004 **Patrik Haslum:** Admissible Heuristics for Automated Planning, 2006, ISBN 91-85497-28-2.

No 1005 **Aleksandra Tešanovic:** Developing Reusable and Reconfigurable Real-Time Software using Aspects and Components, 2006, ISBN 91-85497-29-0.

No 1008 **David Dinka:** Role, Identity and Work: Extending the design and development agenda, 2006, ISBN 91-85497-42-8.

No 1009 **Iakov Nakhimovski:** Contributions to the Modeling and Simulation of Mechanical Systems with Detailed Contact Analysis, 2006, ISBN 91-85497-43-X.

No 1013 **Wilhelm Dahllöf:** Exact Algorithms for Exact Satisfiability Problems, 2006, ISBN 91-85523-97-6.

No 1016 **Levon Saldamli:** PDEModelica - A High-Level Language for Modeling with Partial Differential Equations, 2006, ISBN 91-85523-84-4.

No 1017 **Daniel Karlsson:** Verification of Component-based Embedded System Designs, 2006, ISBN 91-85523-79-8.

No 1018 **Ioan Chisalita:** Communication and Networking Techniques for Traffic Safety Systems, 2006, ISBN 91-85523-77-1.

No 1019 **Tarja Susi:** The Puzzle of Social Activity - The Significance of Tools in Cognition and Cooperation, 2006, ISBN 91-85523-71-2.

No 1021 **Andrzej Bednarski:** Integrated Optimal Code Generation for Digital Signal Processors, 2006, ISBN 91-85523-69-0.

No 1022 **Peter Aronsson:** Automatic Parallelization of Equation-Based Simulation Programs, 2006, ISBN 91-85523-68-2.

No 1030 **Robert Nilsson:** A Mutation-based Framework for Automated Testing of Timeliness, 2006, ISBN 91-85523-35-6.

No 1034 **Jon Edvardsson:** Techniques for Automatic Generation of Tests from Programs and Specifications, 2006, ISBN 91-85523-31-3.

No 1035 **Vaida Jakoniene:** Integration of Biological Data, 2006, ISBN 91-85523-28-3.

No 1045 **Genevieve Gorrell:** Generalized Hebbian Algorithms for Dimensionality Reduction in Natural Language Processing, 2006, ISBN 91-85643-88-2.

No 1051 **Yu-Hsing Huang:** Having a New Pair of Glasses - Applying Systemic Accident Models on Road Safety, 2006, ISBN 91-85643-64-5.

No 1054 **Åsa Hedenskog:** Perceive those things which cannot be seen - A Cognitive Systems Engineering perspective on requirements management, 2006, ISBN 91-85643-57-2.

No 1061 **Cécile Åberg:** An Evaluation Platform for Semantic Web Technology, 2007, ISBN 91-85643-31-9.

No 1073 **Mats Grindal:** Handling Combinatorial Explosion in Software Testing, 2007, ISBN 978-91-85715-74-9.

No 1075 **Almut Herzog:** Usable Security Policies for Runtime Environments, 2007, ISBN 978-91-85715-65-7.

No 1079 **Magnus Wahlström:** Algorithms, measures, and upper bounds for satisfiability and related problems, 2007, ISBN 978-91-85715-55-8.

No 1083 **Jesper Andersson:** Dynamic Software Architectures, 2007, ISBN 978-91-85715-46-6.

No 1086 **Ulf Johansson:** Obtaining Accurate and Comprehensible Data Mining Models - An Evolutionary Approach, 2007, ISBN 978-91-85715-34-3.

No 1089 **Traian Pop:** Analysis and Optimisation of Distributed Embedded Systems with Heterogeneous Scheduling Policies, 2007, ISBN 978-91-85715-27-5.

No 1091 **Gustav Nordh:** Complexity Dichotomies for CSP-related Problems, 2007, ISBN 978-91-85715-20-6.

No 1106 **Per Ola Kristensson:** Discrete and Continuous Shape Writing for Text Entry and Control, 2007, ISBN 978-91-85831-77-7.

No 1110 **He Tan:** Aligning Biomedical Ontologies, 2007, ISBN 978-91-85831-56-2.

No 1112 **Jessica Lindblom:** Minding the body - Interacting socially through embodied action, 2007, ISBN 978-91-85831-48-7.

No 1113 **Pontus Wärnestål:** Dialogue Behavior Management in Conversational Recommender Systems, 2007, ISBN 978-91-85831-47-0.

No 1120 **Thomas Gustafsson:** Management of Real-Time Data Consistency and Transient Overloads in Embedded Systems, 2007, ISBN 978-91-85831-33-3.

No 1127 **Alexandru Andrei:** Energy Efficient and Predictable Design of Real-time Embedded Systems, 2007, ISBN 978-91-85831-06-7.

No 1139 **Per Wikberg:** Eliciting Knowledge from Experts in Modeling of Complex Systems: Managing Variation and Interactions, 2007, ISBN 978-91-85895-66-3.

No 1143 **Mehdi Amirijoo:** QoS Control of Real-Time Data Services under Uncertain Workload, 2007, ISBN 978-91-85895-49-6.

No 1150 **Sanny Syberfeldt:** Optimistic Replication with Forward Conflict Resolution in Distributed Real-Time Databases, 2007, ISBN 978-91-85895-27-4.

No 1155 **Beatrice Alenljung:** Envisioning a Future Decision Support System for Requirements Engineering - A Holistic and Human-centred Perspective, 2008, ISBN 978-91-85895-11-3.

No 1156 **Artur Wilk**: Types for XML with Application to Xcerpt, 2008, ISBN 978-91-85895-08-3.

No 1183 **Adrian Pop:** Integrated Model-Driven Development Environments for Equation-Based Object-Oriented Languages, 2008, ISBN 978-91-7393-895-2.

No 1185 **Jörgen Skågeby:** Gifting Technologies - Ethnographic Studies of End-users and Social Media Sharing, 2008, ISBN 978-91-7393-892-1.

No 1187 **Imad-Eldin Ali Abugessaisa:** Analytical tools and information-sharing methods supporting road safety organizations, 2008, ISBN 978-91-7393-887-7.

No 1204 **H. Joe Steinhauer:** A Representation Scheme for Description and Reconstruction of Object Configurations Based on Qualitative Relations, 2008, ISBN 978-91-7393-823-5.

Linköping Studies in Statistics

No 9 **Davood Shahsavani:** Computer Experiments Designed to Explore and Approximate Complex Deterministic Models, 2008, ISBN 978-91-7393-976-8.

Linköping Studies in Information Science

No 1 **Karin Axelsson:** Metodisk systemstrukturering- att skapa samstämmighet mellan informa-tionssystemarkitektur och verksamhet, 1998. ISBN-9172-19-296-8.

No 2 **Stefan Cronholm:** Metodverktyg och användbarhet - en studie av datorstödd metodbaserad systemutveckling, 1998. ISBN-9172-19-299-2.

No 3 **Anders Avdic:** Användare och utvecklare - om anveckling med kalkylprogram, 1999. ISBN-91-7219-606-8.

No 4 **Owen Eriksson:** Kommunikationskvalitet hos informationssystem och affärsprocesser, 2000. ISBN 91-7219-811-7.

No 5 **Mikael Lind:** Från system till process - kriterier för processbestämning vid verksamhetsanalys, 2001, ISBN 91-7373-067-X

No 6 **Ulf Melin:** Koordination och informationssystem i företag och nätverk, 2002, ISBN 91-7373-278-8.

No 7 **Pär J. Ågerfalk:** Information Systems Actability - Understanding Information Technology as a Tool for Business Action and Communication, 2003, ISBN 91-7373-628-7.

No 8 **Ulf Seigerroth:** Att förstå och förändra systemutvecklingsverksamheter - en taxonomi för metautveckling, 2003, ISBN91-7373-736-4.

No 9 **Karin Hedström:** Spår av datoriseringens värden - Effekter av IT i äldreomsorg, 2004, ISBN 91-7373-963-4.

No 10 **Ewa Braf:** Knowledge Demanded for Action - Studies on Knowledge Mediation in Organisations, 2004, ISBN 91-85295-47-7.

No 11 **Fredrik Karlsson:** Method Configuration - method and computerized tool support, 2005, ISBN 91-85297-48-8.

No 12 **Malin Nordström:** Styrbar systemförvaltning - Att organisera systemförvaltningsverksamhet med hjälp av effektiva förvaltningsobjekt, 2005, ISBN 91-85297-60-7.

No 13 **Stefan Holgersson:** Yrke: POLIS - Yrkeskunskap, motivation, IT-system och andra förutsättningar för polisarbete, 2005, ISBN 91-85299-43-X.

No 14 **Benneth Christiansson, Marie-Therese Christiansson:** Mötet mellan process och komponent - mot ett ramverk för en verksamhetsnära kravspecifikation vid anskaffning av komponentbaserade informationssystem, 2006, ISBN 91-85643-22-X.